



**Институт системного программирования
Российской академии наук**

В.В. Липаев

СЕРТИФИКАЦИЯ ПРОГРАММНЫХ СРЕДСТВ

УЧЕБНИК

**СИНТЕГ®
Москва - 2009**

УДК 004.41(075.8)
ББК 32.973.26-018я73
Л61

Липаев В.В.

Сертификация программных средств. Учебник. – М.: СИНТЕГ, 2010. - 348с.
ISBN 978-5-89638-114-3

В учебнике изложены принципы, методы и средства обеспечения качества в жизненном цикле сложных программных средств (ПС), контроль и подтверждение их соответствие исходным требованиям заказчиков с учетом действующей законодательной базы сертификации и требований национальных и международных стандартов. Качество ответственных программных продуктов должно быть удостоверено и гарантировано компетентными, независимыми организациями путем широких, регламентированных испытаний – *сертификации*.

Учебник состоит из трех частей: методические основы обеспечения качества и сертификации сложных программных средств (лекции 1-2); сертификация процессов производства программных средств (лекции 3-6) и сертификация готовых программных продуктов (лекции 7-11). Представлены концепция, структура и основные требования национальных и международных стандартов в сфере создания программных средств высокого качества. Изложены принципы функционирования систем менеджмента качества на базе международных стандартов серии ИСО 9000; основы стандартизации, сертификации, обеспечения качества и безопасности программных продуктов. Высокое качество программных средств при проектировании и производстве рекомендуется достигать и удостоверять двумя методами: во первых, посредством применения регламентированных высококачественных технологий и систем обеспечения качества ПС, предотвращающих ошибки и дефекты, гарантирующих качество продуктов во время их производства; во вторых, методом использования заключительного контроля и испытаний готовых продуктов и исключения из поставки экземпляров, не соответствующих требуемым показателям качества. Соответственно выделены и подробно изложены два вида сертификационных испытаний: технологий обеспечения жизненного цикла программных средств, поддержанных регламентированными системами качества производства, а также испытаниями готового программного продукта с полным комплектом эксплуатационной документации.

Учебник ориентирован на студентов старших курсов и аспирантов по Программной инженерии, а также на заказчиков, менеджеров, аналитиков и ведущих специалистов, обеспечивающих все этапы жизненного цикла сложных ПС, к которым предъявляются высокие требования к качеству и безопасности функционирования, ограничены доступные ресурсы и сроки разработки.

© **В.В. Липаев**, автор, 2010

ОГЛАВЛЕНИЕ

Введение

Часть 1. МЕТОДИЧЕСКИЕ ОСНОВЫ ОБЕСПЕЧЕНИЯ КАЧЕСТВА И СЕРТИФИКАЦИИ СЛОЖНЫХ ПРОГРАММНЫХ СРЕДСТВ

Лекция 1. Цели и основные принципы сертификации качества производственных предприятий и программных продуктов

Основные понятия, цели и виды сертификации программных средств. Стандартизация и сертификация как основа для обеспечения качества и безопасности программных продуктов. Принципы промышленной сертификации процессов производства и продуктов

Лекция 2. Системные требования, типы и источники дефектов и ошибок в комплексах программ

Формирование назначения, функций и технического задания на проект системы. Системные основы разработки требований к программным продуктам. Типы и источники дефектов и ошибок в комплексах программ.

Часть 2. СЕРТИФИКАЦИЯ ПРОЦЕССОВ ПРОИЗВОДСТВА ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНЫХ СРЕДСТВ

Лекция 3. Базовые стандарты сертификации управления производством программных продуктов

Принципы организации производством программных продуктов. Процессы управления проектами программных средств на основе модели СММІ. Менеджмент – административное управление обеспечением качества систем на основе стандартов **ISO 9000:2000**.

Лекция 4. Стандарты жизненного цикла программных средств для сертификации систем качества предприятий

Подготовка стандартов жизненного цикла программных средств для производства. Базовые стандарты жизненного цикла программных средств. Руководства по применению базовых стандартов жизненного цикла программных средств. Стандарты сопровождения и управления конфигурацией программных средств.

Лекция 5. Подготовка производства программных средств и системы качества предприятия к сертификации ...

Подготовка технологии производства программных продуктов и системы качества предприятия к сертификации. Адаптация базовых стандартов управления производством ISO 12207 и системой качества ISO 90003 программных средств для сертификации. Формирование Руководства по планированию качества программных средств для сертификации на основе стандарта ISO 10005. Формирование Руководства по документированию для сертификации системы менеджмента качества программных средств на основе стандарта ISO 10013.

Лекция 6. Сертификация процессов производства программных продуктов и систем качества предприятий

Требования к квалификации аудиторов по сертификации производства и системам качества программных продуктов. Подготовка и документирование организации процессов сертификации производства и системы качества предприятия. Организация сертификационных испытаний производства и систем качества программных средств.

Часть 3. СЕРТИФИКАЦИЯ ПРОГРАММНЫХ ПРОДУКТОВ

Лекция 7. Формирование требований к характеристикам и качеству программных продуктов

Общие требования к качеству функционирования программных продуктов. Особенности требований заинтересованных лиц к программному продукту. Требования к надежности функционирования программных продуктов. Требования к функциональной безопасности программных продуктов. Требования к производительности и эффективности использования ресурсов ЭВМ программным продуктом в реальном времени. Требования к допустимым рискам динамического применения программных продуктов.

Лекция 8. Организация сертификационных испытаний программных продуктов на соответствие требованиям

Цели, задачи и процессы сертификационных испытаний программных продуктов. Соответствие пространств требований и тестов к функциям и характеристикам комплексов программ. Стратегии и планирование испытаний программных продуктов. Оценки затрат на испытания программных продуктов.

Лекция 9. Подготовка сертификационных испытаний

программных продуктов

Требования к квалификации испытателей сложных программных продуктов. Методы подготовки тестов для испытаний программных продуктов. Выбор тестов для испытаний программных модулей и компонентов. Требования к генерации динамических тестов внешней среды в реальном времени. Компоненты генераторов динамических тестов внешней среды в реальном времени. Средства обработки результатов динамических испытаний программных продуктов в реальном времени. Оценка эффективности динамической генерации тестов в реальном времени.

Лекция 10. Сертификационные испытания программного продукта на соответствие требованиям

Порядок сертификационных испытаний сложного программного продукта. Программа и методики испытаний комплекса программ на соответствие требованиям. Испытания надежности функционирования программного продукта. Испытания функциональной безопасности программного продукта. Испытания производительности и динамического использования ресурсов ЭВМ программным продуктом.

Лекция 11. Удостоверение качества и завершение сертификационных испытаний программных продуктов

Испытания для сокращения и ликвидация опасных рисков при применении программных продуктов. Испытания эксплуатации онной документации на соответствие требованиям к программному продукту. Завершение сертификационных испытаний программных продуктов. Поставка пользователями сертифицированной версии программ продукта для применения. Анализ результатов сертификации и усовершенствование процессов испытаний программного продукта.

Приложение. Международные и государственные стандарты, регламентирующие требования, жизненный цикл, испытания и сертификацию комплексов программ

Литература

ВВЕДЕНИЕ

Современные контракты и планы на создание сложных программных средств (ПС) для информационных систем подготавливаются и оцениваются часто неквалифицированно, на основе неформализованных представлений заказчиков и разработчиков о требуемых функциях и их характеристиках качества. Во многих случаях нужное качество программных средств зависит от интуиции, вкусов и квалификации их производителей, заказчиков и пользователей. Значительные системные ошибки при определении требуемых характеристик качества, оценке трудоемкости, стоимости и длительности создания ПС являются достаточно массовыми и типичными. ***Многие созданные программные продукты не способны полностью выполнять требуемые функциональные задачи с гарантированным качеством*** и их приходится долго и иногда безуспешно дорабатывать для достижения необходимого качества и надежности функционирования, затрачивая дополнительно большие средства и время. В результате программные средства, не соответствующее исходному, декларированному назначению и первоначальным требованиям заказчика к характеристикам качества, не укладываются в согласованные графики и бюджет разработки.

В технических заданиях и реализованных программных продуктах систематически ***умалчиваются и/или недостаточно полно формализуются требования, свойства и метрики качества продукта***, какими характеристиками они должны описываться, как их следует измерять и сравнивать с требованиями, отраженными в контракте, техническом задании или спецификациях. Кроме того, некоторые характеристики часто вообще отсутствуют в требованиях заказчика и потребителей, а также в согласованных с ними документах на программный продукт, что приводит к произвольному их учету или к пропуску при испытаниях. Этому способствует ограниченность ресурсов, особенно времени, необходимых для достижения и оценивания требуемых значений качества, а также недостаточная формализация и документирование всего процесса производства, выбора и анализа комплексов программ.

Сложность анализируемых объектов – комплексов программ и психологическая самоуверенность ряда программистов в собственной **«непогрешимости»** зачастую приводят к тому, что реальные характеристики качества функционирования ПС остаются неизвестными не только для заказчиков и пользователей, но также для самих разработчиков. Проекты оказываются неудачными или даже терпят полный провал из-за недостаточной компетентности привлекаемых разработчиков, их **неадекватного «оптимизма»**, а также вследствие слабого использования современных методов, технологий и стандартов, обеспечивающих требуемое высокое качество продуктов. Отсутствие четкого декларирования в документах понятий и требуемых значений характеристик качества ПС вызывает конфликты между заказчиками-пользователями и разработчиками-поставщиками вследствие неполноты и разной трактовки одних и тех же характеристик.

Возможности и широта применения сертификации при производстве сложных комплексов программ существенно зависят от **методологий и стилей** организации работы специалистов – разработчиков. Эти методологии различаются сферами применения, методами достижения высокого качества комплексов программ, психологическими характеристиками участвующих специалистов и организацией их деятельности в реальном времени. Различие целей и стилей при создании комплексов программ привели к формированию стратегий, позволяющих получать сложные программные **продукты двух классов**, характеризующихся высоким качеством при применении:

- **свободное программное обеспечение**, ориентированное на большое число независимых специалистов, различных по квалификации, дислокации и ответственности за результаты своей добровольной деятельности. Высокие тиражи распространения, применения и гибкая спонтанная модификация программных средств возможны благодаря доступности для многих пользователей исходных текстов программ на стандартном языке программирования, а также унифицированной технологии процессов производства компонентов и сопровождения комплексов программ;

- **закрытые тексты и технологические документы** комплексов программ, создаваемые профессиональными, сплоченными коллективами высокой квалификации под централизованным управлением для конкретных заказчиков определенных систем управления и обработки информации с ограниченным тиражом, производство, рас-

пространение и модификация которых регламентируется и контролируется заказчиком и поставщиком.

Первый класс сложных программных комплексов обычно первоначально создается группами энтузиастов в университетах, корпорациях или сообществах разработчиков и пользователей и распространяется бесплатно. Функции таких комплексов программ ориентированы на решение новых, оригинальных задач массового использования и конкурентоспособных на рынке программных продуктов, куда со временем они поступают. Их разработка, изменение и совершенствование не регламентируется в реальном времени и реализуется по инициативе пользователей без определенных, контролируемых планов и сроков. По мере безвозмездного совершенствования функций и качества программного продукта различными заинтересованными, неуправляемыми пользователями – разработчиками (зачастую сотни и тысячи), расширяется сфера его применения, повышается качество, надежность и безопасность использования, что приводит к активному проникновению в бизнес и систему образования. Вследствие этого слабо документированный, непрерывно, спонтанно изменяющийся программный продукт и его производство трудно сертифицировать, однако целесообразно допускать к его изменениям только специалистов, работающих с использованием сертифицированных технологий и систем качества производства.

Второй класс программных продуктов обычно имеет конкретного заказчика, относительно узкую сферу применения, предназначен для конкретных систем или пользователей, жестко регламентирован технологией производства, модификаций и документирования, что сближает их создание с обычным промышленным производством сложных изделий. Оно управляется планами и ограниченными сроками поставки готовых испытанных продуктов заказчику и пользователям, которые допускается эпизодически изменять только с санкции заказчика. Современные комплексы программ для систем управления и обработки информации в реальном времени активно применяются в сложных критических и ответственных системах динамического управления объектами в высокоточном технологическом производстве, в авиации, космическими аппаратами, атомными электростанциями и оборонной техникой. Такие изделия являются одними из наиболее сложных интеллектуальных систем высокого качества, создаваемых человеком, для которых доступна и необходима сертификация

не только производственных процессов, но и их результатов – программных продуктов. Далее в учебнике основное внимание сосредоточено на этом классе программ.

В соответствии со стандартами, **обеспечение качества** – это «совокупность планируемых и систематически проводимых мероприятий, необходимых для уверенности в том, что продукция или процессы удовлетворяют определенным требованиям потребителей к качеству». Для реализации этого положения предназначены технологические **системы обеспечения качества**, каждая из которых включает: «совокупность организационной структуры, процедур, процессов и ресурсов, обеспечивающую ответственность руководства за качество процессов производства и/или продукции».

Разнообразие областей применения компьютеров становится все шире, и их корректная работа часто является определяющей для качественного управления объектами, успеха предприятий или безопасности человека. Поэтому **тщательное специфицирование и оценивание требуемых характеристик качества программного продукта – ключевой фактор обеспечения их адекватного применения**. Это может быть достигнуто на основе выделения и определения подходящих характеристик с учетом целей использования и функциональных задач программного продукта. Применительно к программным средствам **система обеспечения качества** – это совокупность методов и средств организации управляющих и исполнительных подразделений предприятия, участвующих в проектировании, производстве и сопровождении комплексов программ с целью придания им свойств и качества, обеспечивающих удовлетворение потребностей заказчиков и потребителей при минимальном или допустимом расходовании ресурсов.

Для сложных ПС с особенно высокими требованиями к качеству проектирование, производство и применение таких систем должно сопровождать весь жизненный цикл основной продукции – комплексы программ. Для этого необходимы экономические и моральные стимулы, а также **воля** руководителей, организация специалистов – исполнителей, методы и технология для управления качеством создаваемых программ. **Радикальное повышение качества отечественных сложных программных продуктов и обеспечение их конкурентоспособности** на мировом рынке возможно только на базе внедрения современных стандартизированных технологий и систем ка-

чества, поддерживающих и контролирующих весь жизненный цикл (ЖЦ) программного продукта.

Характеристики качества продукта или процесса зависят от того, для какой *цели*, для какого *потребителя* и для каких *условий* внешней среды делается их оценка. Один и тот же объект может иметь несколько различных представлений и оценок качества, произведенных для различных целей и при разных условиях применения. Различия фактических и требуемых показателей качества объектов или процессов квалифицируются как дефекты или ошибки и являются первичными стимулами для реализации решений по изменению определяемых значений качества.

Объективное повышение сложности функций, реализуемых программами, непосредственно приводит к увеличению их размеров и трудоемкости создания. Соответственно росту сложности программ *возрастает количество выявляемых и остающихся в них дефектов и ошибок*, что отражается на качестве функционирования. По мере увеличения сложности задач, решаемых программами, ошибки могут угрожать катастрофами в системах, выполняющих критические функции управления крупными, дорогими и особенно важными объектами или процессами. Разработка и сопровождение сложных ПС на базе современных технологий позволяет предупреждать и устранять наиболее опасные системные и алгоритмические ошибки на ранних стадиях проектирования, а также использовать неоднократно проверенные в других проектах программные и информационные компоненты высокого качества.

Потребителей – заказчиков интересует, прежде всего, *качество готового конечного продукта* и обычно не очень беспокоит, как оно достигнуто. Однако это качество должно быть ответственно *удостоверено и гарантировано* компетентными, желательно независимыми организациями путем достаточно широких, регламентированных испытаний. Гарантирование качества продукции возможно посредством сертификационных испытаний *процессов производства* комплексов программ и/или испытаний их результатов – *программных продуктов*. *Сертификация* – процедура подтверждения соответствия требованиям, посредством которой независимое от изготовителя и потребителя предприятие *юридически удостоверяет* в письменной форме, что состояние продукции и/или производства и системы менеджмента качества способно обеспечить стабильность характери-

стик изготавливаемой продукции и соответствует установленным заказчиком требованиям и стандартам (см. Приложение). Качество при проектировании и производстве программных продуктов можно обеспечивать *двумя методами* (рис. 1):

- посредством применения регламентированных технологий и систем обеспечения качества процессов проектирования и производства, предотвращающих дефекты и гарантирующих высокое качество продуктов в процессе их создания;

- путем использования заключительного контроля и испытаний готовых продуктов и исключения из поставки или направлением на доработку изделий, не соответствующих требуемым показателям качества.

Первый метод обеспечивает высокое качество выполнения всего процесса проектирования и производства, и тем самым минимум экономических потерь от брака, что рентабельно при создании сложных систем. Многие руководители осознали, что для создания современных прикладных высококачественных информационных систем необходимы не менее *качественные технологии* их производства. При этом качество тех и других должно удостоверяться регламентированными поэтапными и заключительными испытаниями. Качество технологии трудно измерять, контролировать и гарантировать ее соблюдение специалистами, однако оно отражается на качестве продукции, а, следовательно, при этом может оставаться не полностью определенным достигаемое качество программных продуктов. Результаты испытаний *процессов производства* трудно измерять количественными критериями, и обычно характеризуют рядом требований к качественному выполнению стандартизированных производственных процессов. Они оцениваются набором свойств, непосредственно отражающихся на характеристиках качества конечного программного продукта, однако при этом нет гарантии адекватного и однозначного на них влияния. Этот метод может приводить при производстве к неопределенности качества компонентов и комплексов программ в целом и к значительным экономическим потерям за счет затрат на создание не пригодного к использованию продукта (брака), что может быть очень дорого для сложных систем.

Второй метод акцентирован на анализ и контроль качества готового программного продукта, которое удостоверяется при сертификационных испытаниях.

**Обеспечение и удостоверение высокого качества сложных программных продуктов –
испытания технологий и конечного программного продукта**

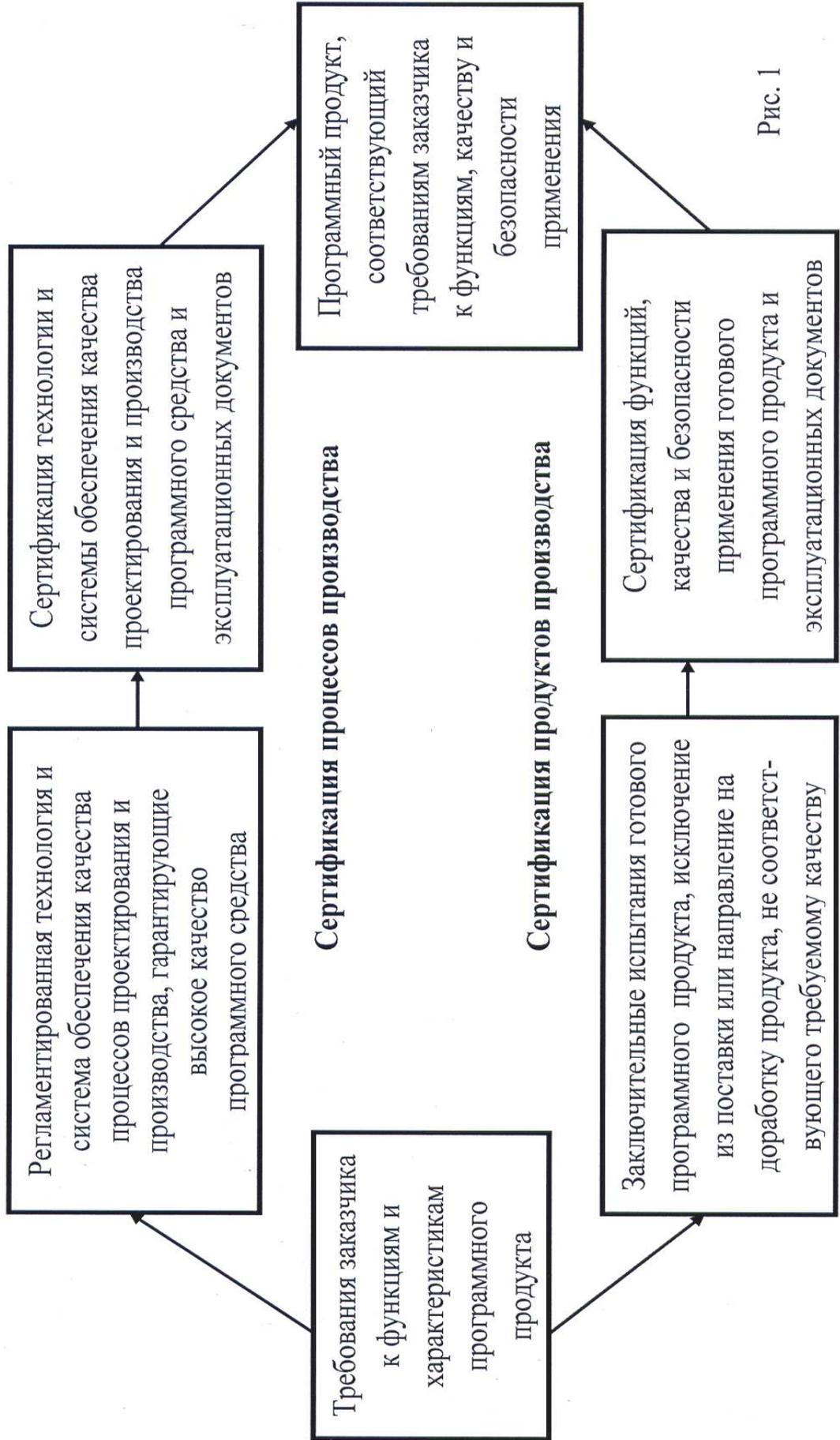


Рис. 1

Достижение при производстве необходимого качества продукции за счет только выходного контроля, при отсутствии или недостатках системы обеспечения качества в технологическом процессе разработки, может приводить к длительному итерационному процессу массовых доработок и повторных испытаний. При сертификационных испытаниях качества готового **программного продукта** могут использоваться стандартизированные количественные и качественные критерии и характеристики, которые непосредственно отражают функции и свойства продукции, интересующие заказчика и потребителей, их можно измерить и достоверно установить.

Таким образом, **задача обеспечения и удостоверения высокого качества сложных программных продуктов сводится к испытаниям** технологий процессов проектирования и производства программных средств, поддержанных системой качества, и конечного продукта, созданного на базе таких технологий. Соответственно можно выделить **два вида сертификационных испытаний: технологий** обеспечения жизненного цикла программных средств, поддержанных регламентированными системами качества и/или готового программного **продукта** с полным комплектом эксплуатационной документации (см. рис. 1).

Взаимосвязь качества разработанных комплексов программ с качеством технологии их создания и с затратами на производство становится особенно существенной при необходимости получения **критического конечного продукта** с особенно высокими значениями характеристик качества. Затраты на производство обычно резко возрастают, когда требуемые показатели качества приближаются к пределу, достижимому при данной технологии и уровне автоматизации процессов проектирования и производства. В этих случаях для обеспечения высокого качества необходима совместная **сертификация технологий и системы обеспечения качества** их проектирования, производства и сопровождения, а **также сертификация готового программного продукта**. Этот вид комплексной сертификации обеспечивает контроль реализации требований алгоритмической и функциональной корректности программного продукта, что особенно важно в программных комплексах для **обеспечения безопасности критических систем**, а также сокращения случайных дефектов и ошибок программ. Сертификация технологических процессов относительно слабо связана с конкретной функциональностью продукта и должна обеспечивать, в основном, его конструктивную безопасность

– минимум рисков, дефектов и технических ошибок. Зачастую это обстоятельство не учитывается в важных критических системах, в которых может оставаться дефект или ошибка в каждой тысяче строк программного кода, которые способны *резко снизить безопасность* сложных программных продуктов.

Наиболее полно в России стандартизирована сертификация производства продукции различных видов. Она поддержана стандартами: Временный порядок сертификации производств с учетом требований ГОСТ Р ИСО 9001:2001; ГОСТ Р 40.003: 2005 и ГОСТ Р ИСО 19011: 2003 (см. Приложение). Их концепции акцентированы на *Системе менеджмента качества производства (СМКП)* (Система менеджмента для руководства и управления производством применительно к качеству продукции) и представлены в Лекции 6. При этом *сертификация производства* определена как *процедура подтверждения соответствия*, посредством которой независимая от изготовителя (продавца, исполнителя) и потребителя (покупателя) организация удостоверяет в письменной форме, что состояние производства (системы менеджмента качества производства) способно обеспечить стабильность характеристик изготавливаемой продукции и соответствует требованиям ГОСТ Р ИСО 9001. К работе по сертификации привлекаются эксперты (аудиторы) по сертификации производств, зарегистрированные в Регистре системы сертификации персонала – *сертифицированные специалисты*. Область сертификации производства определяет заказчик по согласованию с председателем комиссии органа по сертификации продукции.

На практике акцент, распределение ресурсов и усилий на эти два вида сертификации зависят от особенностей характеристик комплекса программ, квалификации коллектива специалистов – разработчиков, требований заказчика – потребителей и наличия сертификационной лаборатории соответствующей тематической квалификации. Для этого организация и процессы сертификации должны специализироваться на определенные виды сертификации и на определенные классы программных продуктов, предусматривать соответствующие технологические работы и документы, обеспечивающие создание продукта требуемого качества. В общем случае *специализация сертифицирующих коллективов* для испытаний технологий и систем качества может быть более широкой и универсальной, чем для

сертификации конкретных программных продуктов, которые описываются более определенными и точными критериями качества.

Кроме сертификации технологических процессов и готовых программных продуктов для эффективного их производства и применения, важное значение имеет **сертификация квалификации специалистов**, реализующих эти процессы. Для этого необходимо их обучение и аттестация на допуск к участию в таких работах, требующих определенных уровней профессиональной квалификации. Они должны освоить и знать методологию программной инженерии, методы тестирования и документирования программных средств, а также основы сертификации производственных процессов и продуктов. Методы и процессы программной инженерии должны включаться в Программы комплексного обучения для обеспечения необходимой профессиональной квалификации руководителей и специалистов, освоения и применения дисциплин регламентированной деятельности крупных коллективов.

Сертифицированные специалисты должны знать и уметь активно применять стандарты как органическую часть производства, развития и контроля качества систем, знать основы экономики и стандартов качества программных продуктов при их применении. Им следует владеть предсказуемыми и управляемыми производственными процессами, зависящими от ресурсов, выделяемых на их достижение, освоить современную культуру промышленного проектирования, производства и жизненного цикла сложных комплексов программ высокого качества. Они должны уметь подготавливать, организовывать, контролировать и учитывать процессы и результаты сертификации технологических процессов и программных продуктов, которые содержатся в **трех группах документов, разрабатываемых и используемых при сертификационных испытаниях**:

- законы, стандарты и общие нормативные документы сертифицируемых технологий производства и продуктов;
- технологическую документацию на изготовление конкретной продукции и связанную с контролем функционирования производства и системы обеспечения качества программного продукта;
- регистрационную документацию, оформляемую по результатам сертификационных испытаний процессов производства продукции и системы обеспечения ее качества.

Основная цель учебника – представить отечественным специалистам современный комплекс задач, методов и стандартов создания и развития сложных программных продуктов *требуемого высокого качества с применением сертификационных испытаний производства и продуктов*. Внимание акцентировано на комплексе индустриальных методов и стандартов, которые непосредственно обеспечивают эффективный жизненный цикл сложных высококачественных программных средств. При этом предполагается, что процессы и технология создания программ и документов опираются на совокупность современных, автоматизированных методов и инструментальных средств поддержки длительного жизненного цикла программных продуктов.

Для решения важнейших проблем развития и применения современных систем требуется *подготовка и воспитание квалифицированных специалистов в области индустрии сложных программных продуктов высокого качества* [1 – 7]. Необходимо их обучение методам и современной программистской культуре промышленного создания высококачественных продуктов. Они должны уметь формализовать требования и достигать конкретные значения характеристик качества функционирования и применения сложных комплексов программ с учетом ограниченных ресурсов, которые доступны для получения и совершенствования этого качества. В жизненном цикле сложных программных средств для обеспечения их высокого качества целесообразно *выделять сертифицированных специалистов*, ответственных за соблюдение регламентированной, промышленной технологии, за измерение и контроль качества комплексов программ в целом и их компонентов. Для систематической, координированной борьбы с угрозами качеству необходимо научить специалистов анализу и оцениванию конкретных факторов, влияющих на качество функционирования программных продуктов со стороны реально существующих и потенциально возможных дефектов и ошибок в программах.

Квалифицированным специалистам следует учитывать *юридическую ответственность* за качество производства и продуктов, за результаты сертификации и *достоверность документации* при применении программных продуктов пользователями. Соответствие продукции требованиям и документам они должны уметь оценивать на основе данных об испытаниях продукции в процессе и/или при за-

вершении производства. Если в соответствии с действующим в стране законодательством к определенной продукции предъявляют обязательные специальные требования качества, установленные национальными стандартами или другими нормативными документами, то при сертификации производства следует уметь проверять способность и гарантии специалистов предприятия обеспечивать соблюдение этих требований.

При применении Учебника предполагается, что обучающиеся знают основы современной программной инженерии и освоили цикл учебных курсов по индустриальному, коллективному созданию сложных программных продуктов. На этой базе необходимо обучать и воспитывать специалистов, способных знать, понимать и применять методы и документы *сертификационных испытаний комплексов программ, гарантирующих их соответствие требованиям заказчиков и пользователей*. Сертификат специалиста программных продуктов должен демонстрировать и создавать уверенность у руководителей и заказчиков, что он способен квалифицированно выполнять порученные работы и у него есть потенциал для совершенствования квалификации и карьерного профессионального роста.

Для удостоверения качества программных продуктов в *первых двух вводных лекциях* специалистам рекомендуется освоить концепцию стандартизации и общие методы сертификационных испытаний, документирования и контроля качества всего жизненного цикла комплексов программ. В учебнике сертификационные испытания разделены по *двум частям* лекций, которые содержат сертификацию *процессов производства* и сертификацию *продуктов производства*, и не обязательно используются одновременно и полностью. Это позволяет на практике выбирать и/или акцентировать внимание заказчиков и сертифицированных на тот класс испытаний, который наиболее соответствует условиям производства, особенностям и характеристикам конкретного комплекса программ.

Иллюстрации в тексте учебника ориентированы на возможность их использования в виде слайдов при чтении лекций в аудитории. Учебник рассчитан на 32 часа лекций, которые сгруппированы в 11 тем, по 2 – 3 академических часа.

Часть 1

МЕТОДИЧЕСКИЕ ОСНОВЫ ОБЕСПЕЧЕНИЯ КАЧЕСТВА В СЕРТИФИКАЦИИ СЛОЖНЫХ ПРОГРАММНЫХ СРЕДСТВ

Лекция 1

ЦЕЛИ И ОСНОВНЫЕ ПРИНЦИПЫ СЕРТИФИКАЦИИ КАЧЕСТВА ПРОИЗВОДСТВЕННЫХ ПРЕДПРИЯТИЙ И ПРОГРАММНЫХ ПРОДУКТОВ

Основные понятия, цели и виды сертификации программных средств

В стандарте **ISO/IEC 00002** – Общие термины и определения в области стандартизации и смежных видов деятельности – *сертификация соответствия* определена как действие третьей стороны, доказывающее, что обеспечивается необходимая уверенность в том, что должным образом идентифицированная продукция, процесс или услуга соответствует конкретному стандарту или другому нормативному документу. В понятие нормативные документы включены документы, содержащие правила, общие принципы или характеристики, касающиеся различных видов деятельности или их результатов, стандарты, технические требования, инструкции и регламенты по применению – рис. 1.1.

Результатом положительных испытаний является *сертификат соответствия* – документ, изданный в соответствии с правилами системы сертификации, удостоверяющий, что обеспечивается необходимая уверенность в том, что должным образом идентифицированная продукция, процесс или услуга соответствует конкретным стандартам или другим нормативным документам.

**Цели и основные принципы сертификации качества
производственных предприятий и программных продуктов:**

- основные понятия, цели и виды сертификации программных средств:
 - стандартное определение понятий и компонентов процессов сертификации;
 - распределение потребностей в сертификации продукции среди производителей и заказчиков программных средств;
 - задачи и эффективность обязательной и добровольной сертификации программных средств;
- стандартизация и сертификация как основа для обеспечения качества и безопасности программных продуктов:
 - исходные данные для сертификации программных средств;
 - потенциальные угрозы качеству в процессе производства и применения программных средств;
 - понятия и проблемы обеспечения безопасности применения программных продуктов;
- принципы промышленной сертификации и стандартизации процессов производства и продуктов:
 - приоритетные цели сертификации процессов производства и программных продуктов;
 - оценка функциональной и экономической целесообразности внедрения сертификации комплексов программ в процессы создания, приемки в эксплуатацию и сопровождение;
 - оценка риска при приобретении и эксплуатации не сертифицированных программных средств с возможным потенциальным ущербом от снижения качества функционирования программного продукта;
 - рациональное использование нормативных документов процессов сертификации, с учетом достигнутого научно-технического и технологического уровня производства программных средств и методов их испытаний;
 - формирование и применение профилей стандартов при сертификации производства и программных продуктов;
 - обоснование и совершенствование технологий производства программных продуктов на основе квалифицированной экспертизы и сертификационных испытаний технологий и продуктов.

Рис. 1.1

При этом в качестве первой стороны в процессе сертификации выступают разработчики или поставщики программных продуктов и их компонентов, а второй стороной являются заказчики, потребители или пользователи.

Качество **должно быть удостоверено и гарантировано** компетентными, желательно независимыми организациями путем достаточно широких, регламентированных испытаний – **сертификации**. Взаимосвязь качества разработанных программ с качеством технологии их создания и с затратами на разработку становится особенно существенной при необходимости получения конечного продукта с высокими значениями показателей качества. Затраты на разработку резко возрастают, когда показатель качества приближается к пределу, достижимому при данной технологии и уровне автоматизации процесса разработки. В этих случаях для обеспечения высокого качества необходима **сертификация технологии и системы обеспечения качества** их проектирования, разработки и сопровождения. Для этого предприятия и процессы сертификации должны предусматривать соответствующие технологические работы и документы, обеспечивающие создание продукта требуемого качества.

Основной целью сертификации технологий проектирования и производства систем и программных средств является защита интересов пользователей, государственных и ведомственных интересов на основе контроля качества продукции, обеспечения их высоких потребительских свойств, повышения эффективности затрат в сфере их производства, эксплуатации и сопровождения, повышения объективности оценок характеристик и обеспечения конкурентоспособности конечного продукта. Проведение сертификации систем качества предприятия обычно **планируется для достижения одной или нескольких целей:**

- определения соответствия или несоответствия элементов системы качества установленным требованиям производства;
- определения эффективности внедренной системы качества предприятия с точки зрения соответствия поставленным целям для обеспечения качества продукции;
- обеспечения возможности предприятию улучшить свою систему качества;
- определения соответствия системы качества производства регламентирующим требованиям.

При анализе и организации процессов сертификационных испытаний технологий и/или объектов системы и комплекса программ следует учитывать *ряд базовых компонентов методологии сертификации*, подлежащих рассмотрению, применению и утверждению для конкретного проекта [2, 4, 11]:

- цели сертификации – правовые, экономические, формальные;
- проблемы, которые необходимо решать для обеспечения высокой эффективности и достоверности результатов сертификационных испытаний;
- исходные данные и документы, необходимые для проведения сертификации: стандарты и нормативные документы, их структура и содержание;
- характеристики и классификацию продуктов и/или процессов сертификации и их показатели качества;
- ресурсы обеспечения испытаний – финансовые, кадры специалистов, их аппаратурная оснащенность, нормативные и инструментальные средства.

В зависимости от области применения систем, от назначения и класса программных продуктов их сертификация может быть обязательной или добровольной. Первоначальные затраты на их проведение могут нести инициаторы испытаний либо заказчик и конкретные потребители систем, либо ее разработчики и поставщики. Соответственно изменяются экономические и юридические механизмы: взаимодействия, распределения прибыли и ответственности за дефекты, за качество сертифицированной продукции или технологии.

Обязательная сертификация необходима для программных продуктов и их производства, выполняющих особо ответственные функции, в которых недостаточное качество, ошибки или отказы могут нанести большой ущерб или опасны для жизни и здоровья людей. Этот ущерб может определяться степенью безопасности применения комплексов программ в авиации, для управления в космосе, в атомной энергетике, в военных системах; или большими экономическими потерями в результате низкого качества функционирования ПС в системах государственного управления, в финансовых, банковских, транспортных системах. В подобных системах обязательная сертификация программных продуктов способствует значительному снижению риска заказчика и повышению безопасности функционирования программного продукта у потребителя до необходимого уровня. В

этих случаях разработчики и поставщики программного продукта обязаны подвергать свои изделия сертификации на соответствие требованиям качества и безопасности для получения разрешения компетентных органов на их реальную эксплуатацию и применение по прямому назначению.

Экономическую рентабельность обязательной сертификации количественно определить чаще всего невозможно. Ее эффект сосредоточен в повышении таких показателей качества систем, как адекватность функционирования, надежность, своевременность представления информации, полнота, достоверность, конфиденциальность, безопасность применения программного продукта, которые зачастую трудно представить и оценить конкретными экономическими категориями. Необходимость проведения обязательной сертификации, как правило, определяет заказчик или потребитель программного продукта для получения формальных *гарантий* достижения производителем заданных значений показателей качества и безопасности продукта. Он же может выступать в качестве заявителя при обращении к сертификационной лаборатории на выполнение испытаний, а также участвовать в формулировании требуемых показателей и в контроле их измерений при испытаниях. Соответственно заявитель финансирует испытания и получает документы, регистрирующие их результаты, в том числе *сертификат соответствия* при положительных результатах. В конечном итоге эти затраты отражаются на цене продукта, однако они первоначально вкладываются заказчиком или потребителем как дополнительная часть стоимости создания заказанного продукта. Роль разработчика состоит в устранении выявленных недостатков для достижения заданного требованиями уровня качества и безопасности. Если результаты испытаний не позволяют сертифицирующей лаборатории и органу дать положительное заключение, то проверенная продукция возвращается разработчикам для доработки и предъявления на повторные испытания.

Добровольная сертификация применяется с целью повышения конкурентоспособности продукции, расширения сферы ее использования и получения дополнительных экономических преимуществ. Экономическими целями сертификации могут быть большие тиражи изделий при производстве, большая длительность жизненного цикла с множеством версий, снижение налогов за высокое качество, увеличение прибыли разработчиков и поставщиков программного продук-

та, сокращение рекламаций пользователей. Результаты сертификации должны оправдывать затраты на ее проведение вследствие получения пользователями продукции более высокого и *гарантированного качества* при некотором повышении ее стоимости. Таким сертификационным испытаниям подвергаются компоненты операционных систем и пакеты прикладных программ широкого применения, повышение гарантии качества, которое выгодно как для поставщиков, так и для пользователей программного продукта. В этих случаях разработчики и поставщики добровольно предоставляют ПС для сертификации с учетом экономических оценок выгоды ее проведения для их продуктов.

Необходимость добровольной сертификации обычно определяет разработчик или поставщик, по инициативе которых формируется совокупность характеристик качества и безопасности продукции, и ее назначение. Кто-то из них выступает в качестве заявителя в сертифицирующую лабораторию для проведения испытаний. При положительных результатах заявитель *получает сертификат* соответствия, который использует для рекламы продукции при взаимодействии с потенциальными пользователями или потребителями. Последние специалисты непосредственно не взаимодействуют с сертифицирующей лабораторией. В случае выявления недостатков в сертифицированном изделии они обращаются непосредственно к поставщику, который обязан обеспечить доработку и повторные сертификационные испытания. При этом возможен прогноз потенциальной экономической рентабельности сертификационных испытаний путем оценки предполагаемого роста прибыли от продажи сертифицированных изделий более высокого качества. Потребитель и пользователь в данном случае может быть заранее неизвестен, и средства на проведение сертификации вкладывают разработчики или поставщики, непосредственно заинтересованные в гарантиях качества собственной продукции. В свою очередь при выявлении скрытых дефектов в сертифицированном изделии может быть подана апелляция в центральный орган с претензиями к качеству проведенных испытаний.

Стандартизация и сертификация как основа для обеспечения качества и безопасности программных продуктов

Решение о выдаче сертификата на технологию, систему обеспечения качества и/или программный продукт должно основываться на оценке соответствия действующим и/или специально разработанным документам (см. Приложение):

- международным и государственным стандартам на технологии создания ПС, их системы обеспечения качества и конкретную продукцию;
- стандартам на сопровождающую программный продукт документацию с учетом необходимости и достаточности номенклатуры документов, семантической полноты и однозначности понимания содержания документов;
- нормативным и эксплуатационным документам на конкретный программный продукт – техническим условиям, техническим описаниям, спецификациям требований и другим регламентирующим документам по согласованному выбору заказчика, разработчика и испытателя;
- действующим международным и национальным стандартам на тестирование, испытания, аттестацию программ, требования которых не ниже требований, регламентируемых отечественными документами.

В исходных нормативных документах должны быть сосредоточены *все функциональные и эксплуатационные характеристики*, обеспечивающие заказчику и пользователям возможность корректного применения сертифицированного продукта и/или технологического процесса во всем многообразии его функций и показателей качества. Выбор и ранжирование показателей качества должны производиться с учетом классов комплексов программ или технологий, их функционального назначения, режимов эксплуатации, степени ответственности и жесткости требований к результатам функционирования и проявлениям возможных ошибок и дефектов.

Сертификационные испытания являются наиболее формализованным и регламентированным этапом тестирования, как продуктов, так и процессов их создания, поддерживаемым значительным числом документов (см. рис. 1.1). При сертификации обычно *руководству-*

ются следующими основными исходными требованиями заказчика:

- утвержденным заказчиком и согласованным с разработчиком техническим заданием и/или спецификацией требований к продукту, а также утвержденным комплектом эксплуатационной документации на комплекс программ и его компоненты, а также на систему обеспечения их качества;
- действующими международными, государственными и ведомственными стандартами на проектирование и испытания программ, а также на техническую документацию производства и продукции;
- Программой испытаний по всем требованиям технического задания и положениям эксплуатационной документации;
- методиками испытаний по каждому разделу требований технического задания и документации.

Программа испытаний, методики их проведения и оценки результатов, разработанные совместно разработчиком и заказчиком при участии специалистов по сертификации, должны быть согласованы и утверждены. Они должны содержать уточнения требований технического задания и документации для проверяемых продуктов и/или процессов, должны гарантировать корректную проверку заданных характеристик.

При незначительной модификации версий комплекса программ или технологии некоторые требования повторно могут не проверяться либо вследствие ограниченных сроков, либо по другим техническим причинам. Поэтому допускается вносить в модифицированные версии отдельные небольшие изменения без полных повторных сертификационных испытаний. Тем не менее, при любых изменениях необходимо подтверждение имеющегося сертификата и проведение некоторого минимума испытаний, удостоверяющих корректность выполненных изменений. Для этого используется система официальных уведомлений пользователей о проведенных изменениях и о подтверждении сертификата. Таким образом, обычный процесс сопровождения ПС должен дополняться соответствующей системой последовательных официальных уведомлений об изменениях и дополнительных контрольных испытаниях, подтверждающих их корректность.

Ресурсы для сертификации программных средств и систем качества должны выделяться в зависимости от характеристик объек-

та сертификации или процесса. В результате сложность комплексов программ, а также доступные ресурсы становятся косвенными критериями или факторами, влияющими на выбор методов испытаний и на достигаемое их качество. При этом следует учитывать, что каждый вид ресурсов в реальных условиях может варьироваться в некотором диапазоне и в той или иной степени ограничен. Наиболее общим видом ресурсов, используемых при испытаниях, являются допустимые временные и финансовые затраты или сметная стоимость сертификации. При анализе этот показатель может применяться или как вид ресурсных ограничений, или как оптимизируемый критерий. Определяющим ресурсом сертификации является совокупная численность и структура коллектива специалистов, а также его квалификация и подготовленность к коллективной проверке конкретного типа ПС и компонентов или системы качества. Аппаратурная оснащенность испытателей конкретного ПС определяется, прежде всего, ресурсами и другими характеристиками реализующей и технологической ЭВМ, доступных для использования коллективу специалистов при сертификации.

В процессе производства программных продуктов потенциальными угрозами качеству являются [3, 4, 11]:

- низкое технологическое качество производства компонентов и комплекса программ, не гарантирует качества конечной продукции;
- недостаточно эффективные средства защиты информационных и программных ресурсов;
- несоответствие реальных и декларируемых функциональных характеристик разрабатываемых компонентов и комплексов программ;
- несоответствие требованиям стандартов, влекущее за собой невозможность взаимодействия, совершенствования и развития систем;
- реализованные алгоритмы обработки информации, неспособны обеспечить в течение жизненного цикла ПС надежное и своевременное представление полной, безошибочной, актуальной и конфиденциальной информации для функционального использования.

В процессе эксплуатации программного продукта потенциальные угрозы качеству составляют:

- сбои и отказы технических средств и программного продукта, длительное время восстановления функционирования систем;

- ухудшения реальных вероятностно-временных характеристик функционирования систем и средств;
- ошибки и неадекватные действия обслуживающего персонала и пользователей программного продукта при подготовке и использовании информации, выполнении технологических операций;
- несанкционированный доступ пользователей к системе, ее информационным и программным ресурсам;
- проникновения и активизации компьютерной вирусной инфекции;
- уничтожения, разрушения или хищения средств обработки информации, оригиналов и дубликатов носителей информации, программных или аппаратных ключей и средств защиты информации;
- перехват информации, навязывание заведомо недостоверной информации, умышленные перегрузки каналов связи и вычислительных ресурсов.

При эксплуатации угрозы качеству функционирования и безопасности ПС могут оказаться реализованными в процессе сбора, подготовки и хранения информации у источника, при ее передаче, приеме, обработке, хранении и выдаче. Наибольший ущерб системе может быть нанесен с автоматизированного рабочего места диспетчера вычислительного процесса, администратора базы данных или ответственного за безопасность информации в программном продукте.

Следствием угроз качеству функционирования и безопасности программного продукта могут быть:

- отказ от адекватного выполнения функции согласно штатному режиму функционирования комплексом программ;
- выполнение программным продуктом непредусмотренных действий;
- блокировка доступа к информационным и программным ресурсам;
- недопустимое ухудшение вероятностно-временных характеристик функционирования ПС;
- разрушение технических средств, нарушение целостности и сохранности программных ресурсов;
- уничтожение, искажение, подмена, ухудшение уровня полноты, достоверности и конфиденциальности информационных ресурсов и программного продукта.

Полное устранение перечисленных угроз принципиально невозможно. Проблема состоит в выявлении *при сертификации* факторов, от которых они зависят, в создании методов и средств уменьшения их влияния на программный продукт, а также в рациональном распределении ресурсов для обеспечения системы, равнопрочной в плане качества и безопасности ПС в целом, как одного из его важнейших свойств.

В общем случае под *качеством функционирования программного продукта* понимается совокупность свойств, обуславливающих его пригодность в течении жизненного цикла обеспечивать надежное и своевременное представление полной, достоверной и конфиденциальной информации для ее последующего целевого использования. При этом качество функционирования продукта подразумевает адекватное заданным требованиям представление выходной информации для использования, выполнение функциональных технологических операций и наличие технических возможностей ПС к взаимодействию, совершенствованию и развитию. Обеспечение качества функционирования ПС неотделимо от решения проблемы *гарантии его функциональной безопасности*. Широкое внедрение ЭВМ в органы государственного управления, в финансовые, банковские, энергетические системы и военную технику, в другие критические сферы деятельности человека привело к появлению ряда общих проблем качества программных продуктов:

- необходимо обеспечивать требуемую *непрерывность и корректность функционирования* программных продуктов в реальном времени, в том числе выполнение требований физической безопасности для людей, экологической безопасности и материальной сохранности имущества при подготовке и использовании программного продукта;
- необходимо обеспечить конфиденциальность информации, *защиту имущественных прав* граждан, предприятий и государства на информацию в соответствии с требованиями гражданского, административного и хозяйственного права, включая защиту секретов и защиту интеллектуальной собственности.

Непрерывно возрастающая *уязвимость программных средств* от случайных и преднамеренных отрицательных воздействий выдвинула эти проблемы в разряд важнейших, определяющих принципиальную возможность и эффективность применения программных

продуктов. Широкому их применению сопутствует проблема обеспечения защиты на уровне, гарантирующем национальную безопасность России [3, 11]. В результате техническая **проблема защиты систем**, комплексов программ и данных вырастает до уровня национальных, государственных проблем, для решения которых необходимы законодательные, организационные, технологические и стандартизационные мероприятия **при сертификации**.

Особенностью этих проблем является возможность нанесения **огромного ущерба** обществу и потребителям информации в процессе функционирования программных продуктов критических приложений **при очень малых затратах на искажение**, уничтожение, кражу информации или программ. Потенциальная возможность и реальные проявления информационных диверсий, а также непредумышленных негативных воздействий, заставляют активизировать исследования, разработки и совершенствование методов и средств обеспечения безопасности, в том числе **путем сертификации**.

Непрерывное повышение уровня автоматизации для подготовки и принятия ответственных решений в системах государственного и военного управления, в финансовых, банковских, энергетических системах перекладывает все большие функции на программные продукты с соответствующими базами данных. В результате **проблема обеспечения безопасности их функционирования сдвигается от лиц высокого ранга и квалификации**, принимающих важные решения, к лицам, непосредственно разрабатывающим методы и средства автоматизации подготовки и реализации таких ответственных решений. Результаты их деятельности воплощаются в прикладные программы и информацию баз данных, а также в алгоритмы, инструкции пользователям и обслуживающему персоналу таких управляющих систем. Одновременно понижается уровень ответственности и системной квалификации лиц, от которых зачастую зависят стратегически важные решения, в итоге некоторые **простейшие ошибки этих лиц могут приводить к тяжелым последствиям, что повышает важность сертификации систем**. Непредусмотренные при проектировании ситуации и ошибки функционирования программ и данных становятся потенциальными источниками угроз отказов и катастроф систем при применении таких программных продуктов.

Возрастание важности и ответственности задач, возлагаемых на программные продукты, сопровождается увеличением их уязвимости

от преднамеренных внешних воздействий с целью использования или разрушения информации и программ, которые по своему содержанию предназначены для применения ограниченным кругом лиц. Следует учитывать также возможность преднамеренной негативной модификации программ и данных отдельными специалистами, участвующими в создании или сопровождении ПС. Для решения этой проблемы созданы и активно развиваются методы, средства и стандарты защиты программ и данных от преднамеренных негативных внешних воздействий. Таким образом, возникла и настоятельно требует системного разрешения проблема обеспечения **сертификации функциональной безопасности** ПС как одного из важнейших потребительских свойств продукта.

Поскольку функциональная и информационная безопасность – одно из важнейших потребительских свойств комплексов программ, проявляемых в процессе их функционирования, а, в свою очередь, множество всех потребительских свойств характеризует качество ПС в целом, то их безопасность является составной частью понятия качества функционирования программных продуктов. Следовательно, понятие **функциональной и информационной безопасности ПС** можно рассматривать как свойство предотвращения реализации потенциальных угроз, направленных на нарушение штатного режима и снижение качества функционирования систем, и нейтрализации последствий их негативного воздействия.

Заключение по результатам сертификационных испытаний разрабатывается сертифицированными и содержит обобщенные сведения о результатах испытаний и обоснование целесообразности выдачи сертификата. В случае получения отрицательных результатов сертификационных испытаний принимается решение об отказе в выдаче сертификата соответствия. После доработки сертифицируемой продукции испытания могут быть повторены. Результаты анализа состояния технологии или качества продукции **оформляются актом**, в котором даются оценки по всем позициям Программы испытаний и содержатся выводы, включающие общую оценку состояния производства и продукции, необходимость корректирующих мероприятий. По результатам сертификационных испытаний и экспертизы документации принимается решение о выдаче сертификата. В случае получения отрицательных результатов сертификационных испытаний принимается решение об **отказе в выдаче сертификата** соответст-

вия. Кроме того, предприятию-заявителю могут быть направлены предложения по устранению предполагаемых причин отрицательных результатов испытаний, и после доработки сертифицируемой продукции испытания могут быть повторены.

Принципы промышленной сертификации и стандартизации процессов производства и продуктов

Сертификация – это стандартизированный, апробированный механизм целенаправленных регламентированных испытаний производства и продуктов, должна быть в максимальной степени ориентирована на противодействие различным угрозам и нейтрализацию негативных последствий их реализации. При этом обеспечение высокого качества функционирования программного продукта в условиях потенциальной возможности реализации угроз должно являться гарантией безопасности применения всей системы. Техническая политика заказчиков, пользователей программного продукта, органов и испытательных лабораторий в области сертификации должны быть гармонично ориентированы на достижение сформированных приоритетных целей и реализацию основных направлений сертификации. В качестве основных направлений полезно учитывать *следующие принципы* [2, 3, 4]:

- применение системных мер, направленных на обеспечение конечного качества функционирования программного продукта, начиная с этапов научно-исследовательских работ и технического задания на проектирование ПС и заканчивая внедрением мер контроля и управления качеством в процессе его эксплуатации;
- анализ функциональной и экономической целесообразности внедрения сертификации комплексов программ в процессы создания, приемки в эксплуатацию и сопровождения;
- соизмеримость осознанного риска при приобретении и эксплуатации не сертифицированных технических и программных средств с возможным потенциальным ущербом от недостаточного качества функционирования программного продукта;
- рационального использования нормативных документов, существующих де-юре и де-факто, с учетом достигнутого научно-технического и технологического уровня производства комплексов программ и методов их испытаний;

- приоритетного инвестирования и государственного заказа разработчикам, производящим продукцию высокого качества, подтверждаемого сертификатами соответствия производства и продукции.

При введении сертификации целесообразна количественная оценка ожидаемого уровня повышения качества функционирования комплекса программ. Эти оценки могут базироваться на результатах моделирования процессов производства и функционирования системы в условиях реализации различных угроз. Эффект от сертификации в каждом случае состоит в **повышении гарантий качества** на основе увеличения уровня приоритетных испытаний (см. рис. 1.1).

Приоритетными целями сертификации для заказчиков и разработчиков являются:

- установление достигнутого уровня качества функционирования программного продукта, снижение риска заказчика при задании, разработке и принятии продукта в эксплуатацию, повышение его функциональной безопасности;

- обоснование рациональных технологических решений по производству, совершенствованию и развитию комплекса программ на основе квалифицированной экспертизы и испытаний технологии и продуктов;

- удовлетворение потребностей рынка в качественной продукции и расширение экспортных возможностей отечественных программных продуктов.

Первоочередными объектами сертификации выступают **системы качества на предприятиях-разработчиках**, разрабатываемые и созданные комплексы программ и непосредственно системы. Именно комплексная сертификация систем позволяет гарантировать качество их функционирования в целом. Ведение информационного банка данных о **сертифицированных программных продуктах**, типовых технических решениях и результатах сертификации, а также хранение копий сертифицируемых ПС позволят избежать дублирования, повторения ошибок проектирования и унифицировать процессы производства. За счет этого достигается сокращение сроков и затрат на разработку систем. Их реализация позволит занять научно-технологическую «**нишу**», связанную с разработкой и внедрением методов **сертификационных испытаний ПС**, позволяющих повысить их качество функционирования и безопасность. Развивая в стра-

тегическом плане решения технических вопросов поддержания жизненного цикла ПС целесообразно базироваться на стандартах ISO, а также на других зарубежных стандартах (см. Приложение).

При создании и сопровождении сложных, распределенных, тиражируемых программных продуктов требуется гибкое формирование и применение *гармонизированных совокупностей базовых стандартов* и нормативных документов разного уровня, выделение в них требований и рекомендаций, необходимых для эффективной реализации конкретных функций систем. Для унификации и регламентирования реализации этих функций, совокупности базовых стандартов должны адаптироваться и конкретизироваться применительно к определенным классам проектов, их функций, процессов и компонентов. В связи с этим выдвинулось и сформировалось понятие *«профиля стандартов»* как основного инструмента функциональной стандартизации и сертификации.

Профиль стандартов – это совокупность нескольких (или подмножество одного) базовых стандартов (и других нормативных документов) с четко определенными и гармонизированными подмножествами обязательных и факультативных возможностей, предназначенная для реализации заданной функции или группы функций. Функциональная характеристика (заданный набор функций) объекта стандартизации является исходной для формирования и применения профиля этого объекта или процесса. В профиле выделяются и устанавливаются допустимые факультативные возможности и значения параметров каждого базового стандарта и/или нормативного документа, входящего в профиль. Он должен использовать факультативные возможности и значения параметров в пределах допустимых, выбранные из альтернативных вариантов. На базе одной и той же совокупности базовых стандартов могут формироваться и утверждаться различные профили стандартов для разных проектов и сфер применения. Ограничения базовых документов профиля и их гармонизация, проведенная разработчиками профиля, должны обеспечивать качество, совместимость и корректное взаимодействие компонентов системы, соответствующих профилю в заданной области его применения.

Основными целями применения профилей стандартов при создании, применении и сертификации комплексов программ являются:

- снижение трудоемкости, длительности, стоимости и улучшение других технико-экономических показателей проектов систем и комплексов программ;
- повышение качества разрабатываемых или применяемых покупных компонентов и комплексов в целом при их разработке, приобретении, эксплуатации и сопровождении;
- обеспечение расширяемости комплексов программ по набору прикладных функций и масштабируемость в зависимости от размерности решаемых задач;
- обеспечение переносимости программ и данных между разными аппаратно-программными платформами.

Применение профилей, относящихся к программным комплексам (функциональным частям систем), облегчает повторное использование в проектируемой системе уже разработанных и проверенных программных компонентов. Профили ПС унифицируют и регламентируют только часть требований и характеристик продуктов и процессов, выделенных и формализованных на базе стандартов и нормативных документов. Другая часть функциональных и технических характеристик систем определяется заказчиками и разработчиками творчески, без учета положений нормативных документов. Чем сложнее объекты или процессы, подлежащие стандартизации и сертификации, тем больше необходимо использовать и формулировать предварительных условий, учитываемых в требованиях и рекомендациях стандартов, которые следует адаптировать и конкретизировать для корректного их применения в определенном проекте.

Профиль стандартов жизненного цикла комплекса программ (функциональных частей системы) должен **определять архитектуру программного комплекса** (модели функций, логические модели данных, внешние интерфейсы) и их структуру (разбиение системы на подсистемы и систем на модули, определение унифицированных интерфейсов взаимодействия между комплексами программ и их компонентами). Жизненный цикл программных средств отражается в профиле стандартов набором: процессов, этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующим ведение разработки, сопровождение и эксплуатацию, от анализа и подготовки требований **до завершения сертификационных испытаний** ряда версий программного продукта и прекращения их использования. **Жизненный цикл** включает описания исход-

ной информации, способов и методов выполнения операций и работ, устанавливает требования к результатам и правилам их контроля, а также определяет содержание технологических и эксплуатационных документов. Он определяет организационную структуру коллектива специалистов, регламентирует распределение и планирование работ, а также контроль за ходом разработки. Каждый из выделенных профилей должен для последующего длительного использования пройти стадию формирования, адаптации и параметризации применительно к характеристикам стандартизируемых продуктов или процессов.

Для корректного применения **описания профилей стандартов должны содержать:**

- определение целей, которые предполагается достичь применением данного профиля стандартов;
- перечисление функций программного продукта или процесса производства, определяемого данным профилем;
- формализованные сценарии применения базовых стандартов и спецификаций, включенных в данный профиль;
- ссылки на конкретный набор стандартов и других нормативных документов, составляющих профиль, с точным указанием используемых положений, редакций и ограничений, способных оказать влияние на достижение корректного взаимодействия объектов стандартизации при использовании данного профиля;
- информационные ссылки на спецификации тестов проверки соответствия профилю.

Особенности организационных структур, различия в размерах и сложности проектов, требованиях к системам и применяемым методам их разработки, необходимость преемственности с системами, находящимися в эксплуатации, влияют на организацию разработки, приобретения, применения и сопровождения аппаратных и программных средств. Для **эффективного применения конкретного профиля стандартов необходимо:**

- выделить объединенные единой логической связью проблемно-ориентированные области функционирования систем, где могут использоваться стандарты, общие для одной организации или группы предприятий;
- идентифицировать стандарты и нормативные документы, варианты их применения и параметры, которые необходимо включить в профиль стандартов;

- документально зафиксировать участки конкретного профиля, где требуется создание новых стандартов или нормативных документов, и идентифицировать характеристики, которые могут оказаться важными для разработки недостающих стандартов и нормативных документов этого профиля;
- формализовать профиль в соответствии с его категорией, включая стандарты, различные варианты нормативных документов и дополнительные параметры, которые непосредственно связаны с профилем.

Разделение труда специалистов в крупных проектных коллективах приводит разработчиков к необходимости их дифференциации по квалификации и областям деятельности. При этом выделяются разработчики программных компонентов и высоко квалифицированные системотехники – интеграторы сложных проблемно – ориентированных ПС. Эти две категории специалистов работают над проектом в значительной степени разными методами, на разных языках проектирования, используют различные средства автоматизации и имеют на выходе различные результаты. Вследствие этого необходим их профессиональный отбор, специфически ориентированные подготовка и методы обучения. Специалисты по системному проектированию сложных ПС и комплексированию компонентов должны иметь, прежде всего, хорошую подготовку по системному анализу алгоритмов и пакетов прикладных программ, по методам оценки эффективности проектов, организации и планированию крупномасштабных разработок программ и баз данных. Им необходима высокая квалификация по архитектурному построению, комплексной отладке и испытаниям ПС определенных классов и умение организовать коллектив для решения общей целевой задачи системы.

Наиболее часто применяются **две схемы организации коллектива специалистов для реализации проектов сложных программных продуктов:**

- формирование для каждого проекта жесткой организационной структуры коллектива с полным составом необходимых специалистов под единым, централизованным руководством;
- выделение руководителя (главного конструктора) и небольшой группы интеграторов, по заданиям которых выполняются частные работы узкими специалистами, не входящими организационно в единый коллектив реализации конкретного проекта.

Первая схема предпочтительна, когда предприятие реализует небольшое число особенно крупных проектов – заказов и имеет возможность для каждого из них скомплектовать полноценную, организационно замкнутую бригаду. Однако при этом возможны простои отдельных специалистов из-за ожидания заданий или результатов последовательных этапов проекта.

Вторая схема для предприятия может иметь преимущества при большом числе относительно небольших проектов, близких по содержанию и функциональному назначению. В этом случае большинство специалистов одновременно участвуют в нескольких проектах по локальным заданиям интеграторов различных проектов, и может использоваться более полно. Однако задачи интеграторов при этом усложняются и требуют более высокой квалификации.

Успех и качество при проектировании сложных программных комплексов все больше зависят от слаженной работы и профессионализма коллектива специалистов на всех этапах и уровнях создания таких проектов. При проектировании **необходима оценка требований к тематической и технологической квалификации возможного коллектива специалистов, его способности создавать и реализовать разработанный проект с заданным качеством.** Комплексное, скоординированное применение профилей стандартов и средств в процессе создания, развития и применения комплексов программ позволяет исключать многие виды дефектов или значительно ослаблять их влияние. Тем самым уровень достигаемого качества программного продукта **становится предсказуемым и управляемым**, непосредственно зависящим от ресурсов, выделяемых на его достижение, а главное, от системы качества и эффективности технологии, используемых на всех этапах жизненного цикла ПС.

Лекция 2

СИСТЕМНЫЕ ТРЕБОВАНИЯ, ТИПЫ И ИСТОЧНИКИ ДЕФЕКТОВ И ОШИБОК В КОМПЛЕКСАХ ПРОГРАММ

Формирование назначения, функций и технического задания на проект системы

Основы процессов, составляющих жизненный цикл сложных вычислительных систем, содержит стандарт **ISO 15288:2002** (см. Приложение). Стандарт отражает потребности в единой стратегии и структуре для совершенствования связей и *кооперации между специалистами*, создающими, использующими и управляющими современными системами, для того, чтобы они вместе могли согласованно действовать. Документ охватывает концепции и идеи, имеющие отношение к системам, он включает процессы заказа и поставки систем, содержит возможность для оценки и совершенствования процессов их жизненного цикла. Процессы и компоненты в данном стандарте образуют полное множество, из которого предприятия могут конструировать модели систем, соответствующие их продукции. Стандарт может быть использован для выбора, структуризации и применения установленной среды в интересах производства продукции, при ее оценке на соответствие заявленной и сформированной внешней среде. Он рекомендуется заказчикам и разработчикам *для согласования требований*, касающихся процессов и результатов их деятельности – рис. 2.1.

Требования к системе должны определять совокупность работ, которая позволяет в рамках предприятия и/или проекта оптимизировать прибыль и уменьшать риски, возникающие вследствие принятия проектных решений и осуществления соответствующих действий.

Эти работы обеспечивают условия для того, чтобы продукция была нужными и полезными, экономически выгодными, функциональными, надежными, пригодными к обслуживанию, производству и использованию, и обладали другими качествами, необходимыми

для того, чтобы удовлетворить требования, как заказчика, так и пользователя.

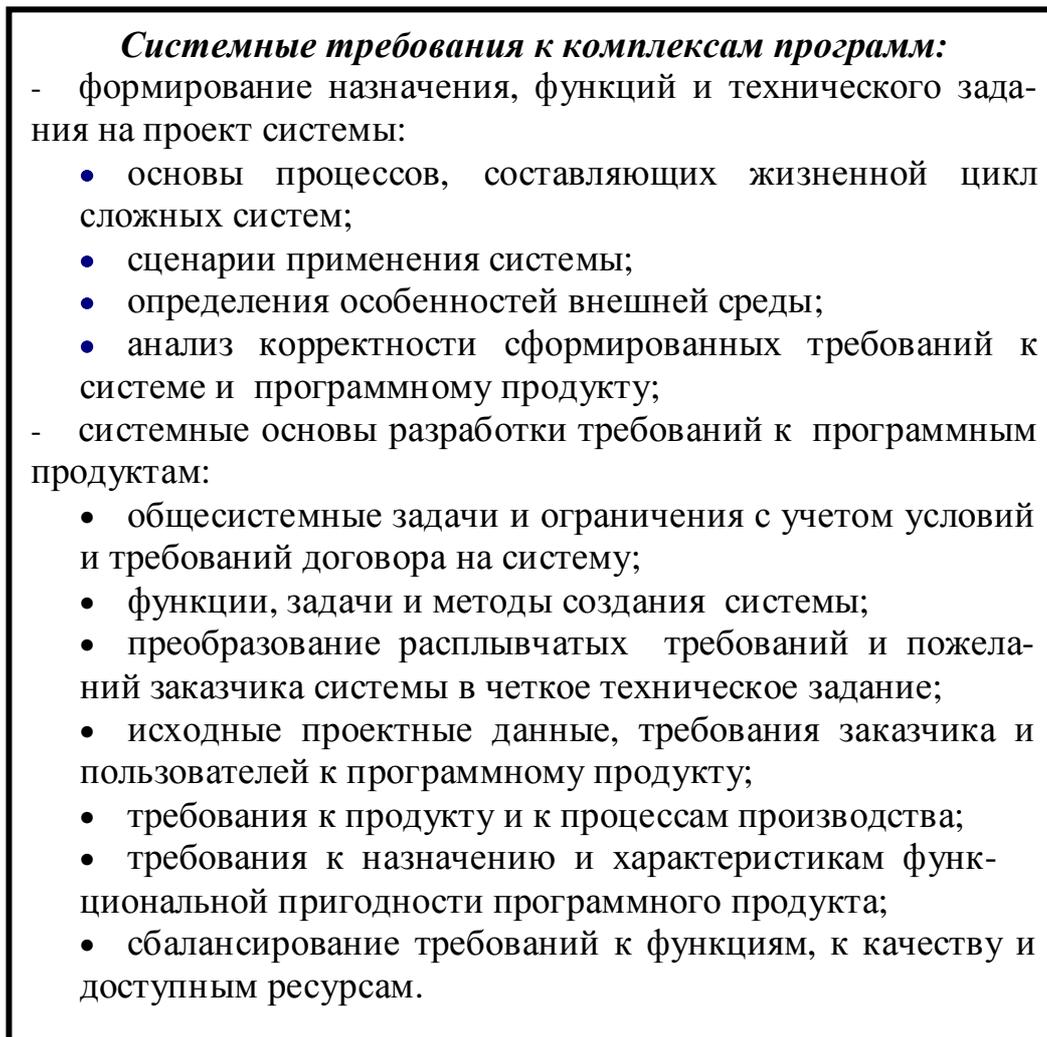


Рис. 2.1

Они также обеспечивают условия для того, чтобы продукция соответствовала ожиданиям или законным требованиям общества, включая общие факторы обеспечения здоровья, безопасности, надежности и экологии.

Технологические процессы в стандарте включают:

- определение требований заказчика;
- анализ полноты и корректности требований;
- проектирование архитектуры системы;
- реализацию компонентов проекта;
- комплексирование компонентов системы;
- верификацию реализации требований к системе;

- валидацию (аттестацию) системы;
- сопровождение системы;
- передачу системы заказчику;
- списание (утилизацию) системы.

Цель процесса определения и формирования требований заказчика состоит в формулировании требований к системе, выполнение которых должно обеспечить **функциональные возможности**, необходимые пользователям системы и иным заинтересованным лицам, в заданной эксплуатационной среде. Должны быть определены цели создания и назначения системы, а также функции и область ее применения. Следует выявить и зафиксировать заинтересованных лиц или их группы, которые будут связаны с системой на протяжении всего жизненного цикла, а также их потребности и пожелания (см. рис. 2.1). Эти данные анализируются и преобразуются в общий набор требований заказчика, они описывают необходимое поведение системы в процессе взаимодействия с эксплуатационной средой и совокупность образцовых показателей, проверка на соответствие которым является целью процесса аттестации и позволяет подтвердить, что система отвечает заявленным требованиям. В соответствии с принятой политикой предприятия и/или проекта для **определения требований к системе** должны осуществляться **следующие базовые действия** [1, 4, 9]:

- идентификация заинтересованных лиц или групп специалистов, имеющих законный интерес к системе в течение ее жизненного цикла;
- определение ограничений системных решений, которые являются неизбежным следствием существующих соглашений, управленческих или технических решений;
- установление масштаба и требуемых ресурсов для реализации системы;
- определение представительного набора последовательных действий для идентификации всех требуемых функциональных возможностей, которые отвечают предполагаемым сценариям и внешним средам применения, функционирования и сопровождения системы;
- определение взаимодействия между пользователями и системой;

- анализ полноты и корректности множества, установленных требований к системе;
- специфицирование экологических, медицинских, безопасности и других требований заказчика, имеющих отношение к критическим показателям применения системы;
- разрешение проблем и конфликтов, возникающих в связи с определением требований к системе;
- доведение до сведения соответствующих заинтересованных лиц результатов анализа требований для подтверждения того, что их потребности и ожидания были поняты и корректно отражены в требованиях;
- документирование требований заказчика в форме, приемлемой для управления требованиями в течение жизненного цикла системы.

В состав заинтересованных лиц могут входить: заказчики; пользователи; предприятия, занимающиеся сопровождением; разработчики; производители, поставщики и покупатели. Требования заказчика могут выражаться в форме потребностей, пожеланий и ограничений. Они выражаются в виде моделей, ориентированных на цели и назначение системы, и описывающих систему в контексте внешней среды и условий функционирования. Для осуществления этих действий может быть полезной модель характеристик качества программной продукции, которая отражена в стандартах **ISO 9126** и **ISO 25000** (см. Приложение).

Требования заказчика должны учитывать нужды и потребности общества и ограничения, налагаемые приобретающей организацией, а также возможностями обслуживающего персонала. При выборе решений следует **исключать необоснованные ограничения**. Для основных потребностей заказчика необходимо устанавливать показатели эффективности, чтобы эксплуатационные характеристики могли быть измерены и оценены. Ограничения могут возникать в результате существования ситуаций или областей решения, определенных заказчиком; решений по реализации, принятых на более высоких уровнях системной иерархии; требований по использованию определенных обеспечивающих систем, ресурсов или персонала.

Сценарии применения системы должны использоваться для анализа ее функционирования в заданной среде с целью **выявления требований**, которые формально могли быть не заданы заказчиком,

например, юридические или социальные обязательства. Также следует анализировать социальное воздействие организации на пользователей, которые могут повлиять на использование системы или сдерживать процесс ее проектирования. Стандарты и правила должны использоваться для **определения особенностей внешней среды**:

- архитектуры и ресурсов аппаратуры вычислительной системы;
- рабочих мест, внешней среды и инструментов визуализации, в том числе используемого вспомогательного оборудования;
- нормальных, необычных и чрезвычайных ситуаций функционирования системы;
- набора, обучения и развития операторов и пользователей;
- физических, умственных, квалификационных способностей и ограничений пользователей.

Следует идентифицировать угрозы безопасности и, если они имеются, то устанавливать **требования и функции по обеспечению безопасности применения системы**. Сюда относятся риски, связанные с процессами функционирования и сопровождения, здоровьем и безопасностью, угрозами собственности и с внешними воздействиями (стандарт **IEC 61508**). Необходимо устанавливать требуемые функции повышения безопасности, включая смягчение и сокращение рисков, ссылаясь на соответствующие стандарты и утвержденные профессиональные правила, в случае их применимости.

Анализ корректности сформированных требований к системе и программному продукту включает выявление и идентификацию противоречивых, пропущенных, неполных, неоднозначных, нелогичных или непроверяемых требований; расстановку приоритетов и разрешение проблем, возникающие в связи с определением требований. Сюда же относятся требования, которые принципиально не могут быть реализованы или которые нецелесообразно реализовывать. Необходимо достигать соглашения с заинтересованными лицами, по решениям, касающимся противоречивых, нецелесообразных и неосуществимых требований, устанавливать, чтобы их требования были корректно изменены.

Цель процесса анализа требований состоит в преобразовании потребностей заказчика, выраженных в виде пользовательского представления о системе, в **формализованные функциональные возможности**. В ходе этого процесса должно создаваться четкое представ-

ление о будущей системе, которая будет удовлетворять требованиям заказчика и не потребует специальных мероприятий в связи с ее практическим применением. В результате определяется комплекс оцениваемых требований, которые задают, с точки зрения разработчика, какими функциями и характеристиками должна обладать система и какими должны быть их значения, чтобы удовлетворить требованиям заказчика. В результате **успешного осуществления анализа требований** должны формализоваться:

- требуемые характеристики, свойства, функциональные и эксплуатационные требования к системе;
- ограничения, влияющие на архитектурное проектирование системы, а также на средства ее реализации;
- способы, с помощью которых обеспечиваются целостность выполнения системных требований, потребностей заказчика и взаимное соответствие между ними;
- основа для верификации комплекса системных требований.

Цель процесса проектирования архитектуры системы состоит в синтезе структурных решений, которые бы удовлетворяли системным требованиям. Они определяются на основе требований к набору системных компонентов, из которых образуется система. Конкретные требования, являются основой для верификации реализации системы и для разработки стратегий комплексирования и верификации. В результате проектирования архитектуры системы должны:

- устанавливаться порядок, в соответствии с которым выполняется проектирование архитектуры;
- задаваться реализуемый набор функциональных, системных компонентов, которые удовлетворяют требованиям, предъявляемым к системе;
- определяться требования к интерфейсам компонентов и с внешней средой при проектировании архитектуры;
- устанавливаться связь между проектированием архитектуры и системными требованиями;
- определяться основа для верификации функциональных, системных компонентов;
- устанавливаться основа комплексирования системных компонентов.

При проектировании архитектуры следует **определять и задавать производные требования** для описания функциональных и эксплуатационных требований, функциональных возможностей и свойств, требований к потокам данных в соответствии с логической архитектурой. Проектные критерии включают физические, эксплуатационные, поведенческие характеристики, характеристики надежности и устойчивости. Определения требований заказчика, анализа требований и проектирования архитектуры рекурсивно применяются для последовательной детализации системной архитектуры до тех пор, пока компоненты не могут быть созданы, приобретены, повторно использованы **для программного продукта**. Эти требования выполняются в контексте известных факторов и предположений.

Необходимо устанавливать стоимостные, технические и временные **риски**, связанные с решениями о разработке, модификации или закупке компонентов. Следует оценивать альтернативные проектные решения, моделируя их с той степенью детализации, которая позволяет сравнивать спецификации по таким выраженным в требованиях заказчика критериям, как характеристики функционирования, стоимость, затраты времени и риски. Определения должны производиться с той степенью детализации и контроля, которая соответствует созданию, использованию и **обеспечению целостности системы**. Интерфейсы типа «человек-система» и «система-система» также должны определяться и контролироваться.

Спецификации требований являются результатом системных решений и источником для соглашений о приобретении функциональных компонентов и критериев их приемки. Они также являются основой для принятия решений по производству, повторному использованию или приобретению компонентов, для их верификации и для установления стратегии комплексирования этих компонентов в крупную систему. Порядок проектирования архитектуры должен позволять **проводить анализ изменений требований** в течение жизненного цикла системы, а также является источником информации для последующего повторного использования архитектуры и компонентов. Также это источник информации, при помощи которой **определяются необходимые тесты** в ходе комплексирования системы.

Системные требования должны храниться в соответствующем **архиве данных**, что позволит отслеживать связь между изменениями требований заказчика и проектированием архитектуры системы. Сле-

дует **документировать требования заказчика** в форме детального **технического задания**, приемлемого для управления реализацией требований в течение жизненного цикла системы. В техническом задании должны также содержаться общие сведения о проекте системы. Они являются базой для поддержания взаимного соответствия между существующими системными требованиями и потребностями источника сведений, которым в дальнейшем пользуются для определения последующих системных требований.

Системные основы разработки требований к программным продуктам

Стоимость аппаратных средств постепенно снижается, тогда как **стоимость программных продуктов стремительно возрастает**. Возникла необходимость в новых технологиях и методах управления комплексными проектами разработки больших программных продуктов. Такие методы составили **программную инженерию**. Возрастает как размер целостного программного продукта, так и его сложность. Это также является одной из причин возникновения проблем при разработке крупных программных продуктов, как и то, что многие компании, занимающиеся их производством, не уделяют должного внимания эффективному применению современных методов и стандартов, разработанных в программной инженерии. Инженеры должны понимать, что они работают в организационных и финансовых рамках заключенных контрактов и ищут решение поставленной перед ними задачи **с учетом условий и требований договора на систему**. Программная инженерия не рассматривает технические аспекты детального создания компонентов – в ее ведение входят такие задачи, как управление проектами и разработка методов, теорий и средств, необходимых для обеспечения жизненного цикла сложных комплексов программ. Программирование компонентов – это дело, главным образом, **индивидуальное**, а программная инженерия систем – всегда **коллективная работа**.

По мере увеличения в системах роли готовых программных компонентов и продуктов **методы программной инженерии** все шире используются в процессе создания разнообразных систем. Специалисты по созданию программного продукта **должны понимать функции, задачи и методы создания системы**, поскольку возникающие проблемы часто являются результатом решений, принятых

руководителями и разработчиками проекта всей системы. Технологии программной инженерии часто являются критическим фактором при разработке сложных вычислительных систем. Высокие темпы роста основных ресурсов аппаратных средств (приблизительно на порядок каждые пять лет) и сохраняющаяся потребность в увеличении их использования со стороны различных пользователей и сфер применения приводят к необходимости *адекватного совершенствования технологий создания крупных комплексов программ*. Методологической базой такого развития может служить Свод знаний о программной инженерии – стандарт **ISO 19759:2005 – SWEBOOK**, который сосредоточил в двенадцати разделах основные концепции в этой сфере деятельности [1, 10, 17].

Системные, функциональные требования – должны отражать высокоуровневые требования к задачам и функциям программного комплекса, содержащего ряд взаимосвязанных подсистем. При этом система может быть как целиком программной, так и состоять из программной и аппаратной частей. В общем случае, частью системы может быть персонал, выполняющий определенные функции системы. Программные средства все больше встраиваются в различные системы. Работа с такими проектами требует от программного инженера широкого взгляда на общие *задачи проектирования систем*. Программному инженеру необходимо участвовать в выработке требований для всей системы, а также освоить прикладную область, требованиям которой должен будет отвечать программный продукт (см. рис. 2.1).

Программный менеджер и/или системный инженер должен быть знаком с основными способами проектирования сложных систем, знать, *как перевести расплывчатые требования и пожелания заказчика системы в четкое техническое задание*, и уметь разговаривать с заказчиком и потенциальными пользователями системы на языке предметной области, а не на профессиональном программистском жаргоне. Программный инженер – это *член команды*, поэтому должен обладать навыками общения и межличностных отношений, а также уметь планировать не только свою работу, но и координировать ее с работой других. Иногда люди покидают проект, и это влияет не только на работу, выполняемую непосредственно ими, но и на работу тех, кто от них зависит. Замена разработчика в проекте может требовать обучения и серьезнейшей подготовки нового специалиста

для освоения им технических условий проекта и текущего состояния системы.

В *требованиях к продукту и к процессу* должно проводиться разграничение соответствующих требований как свойств продукта, который необходимо получить, и процесса, с помощью которого продукт будет создаваться; при этом ряд требований может быть заложен неявно и программные требования могут порождать требования к процессу. Они устанавливают основные соглашения между пользователями (заказчиками) и разработчиками в отношении того, **что должна делать система**, и чего от нее не стоит ожидать. Документ должен включать процедуры проверки получаемого программного продукта на соответствие предъявляемым ему требованиям, характеристики, определяющие качество и методы его оценки, вопросы безопасности и другие свойства. В то же время, существуют полуформальные и формальные методы, используемые для спецификации программных требований. В любом случае, задача состоит в том, чтобы программные требования были ясны, связи между ними прозрачны, а содержание спецификаций не допускало разночтений и интерпретаций, способных привести к созданию программного продукта, не отвечающего потребностям заинтересованных лиц [7].

Исходные проектные данные и требования к программному продукту, включая установленные законодательные и регламентирующие нормативные требования, должны быть оформлены документально, а их выбор проанализирован поставщиком на адекватность. Спецификацию требований должен представить потребитель-заказчик. Однако по взаимному согласию ее может подготовить поставщик-разработчик в тесном сотрудничестве с потребителем для предупреждений разногласий путем, например, уточнения определенных терминов, объяснения предпосылок и обоснования требований. **Трассировка требований** обеспечивает связь между требованиями и отслеживание потребностей источников требований. Трассировка является фундаментальной основой проведения анализа влияния при изменении требований, помогая предсказывать эффект от внесения таких изменений. Неполные, двусмысленные или противоречивые требования должны быть предметом урегулирования с лицами, ответственными за их предъявление.

Для конкретного комплекса программ доминирующие требования выделяются и определяются его **функциональным назначением**.

Программы для ЭВМ как объекты проектирования, разработки, испытаний и оценки качества характеризуются в жизненном цикле следующими **обобщенными характеристиками**:

- проблемно – ориентированной областью применения, техническим и социальным назначением программного комплекса;
- конкретным классом и назначением решаемых функциональных задач с достаточно определенной областью применения соответствующими пользователями;
- масштабом и сложностью совокупности программ и базы данных, решающей единую целевую задачу системы;
- архитектурой комплекса программ и базы данных;
- необходимыми составом и требуемыми значениями характеристик качества функционирования программ и величиной допустимого ущерба – риска из-за недостаточного их качества;
- составом потребителей характеристик комплекса программ, для которых важны соответствующие атрибуты качества;
- комплектом стандартов и их содержания, которые целесообразно использовать при выборе характеристик комплекса программ;
- реальными ограничениями всех видов ресурсов проекта;
- степенью связи решаемых задач с реальным масштабом времени или допустимой длительностью ожидания результатов решения задач;
- прогнозируемыми значениями длительности эксплуатации и перспективой создания, множества версий программного продукта;
- предполагаемым тиражом производства и применения комплекса программ;
- степенью необходимой документированности программного продукта.

Исходные требования к комплексу программ могут быть представлены и согласованы в **составе спецификации всей системы**. Если программный продукт нуждается во взаимодействии с другими программными или аппаратными продуктами, то в спецификации требований должны быть оговорены непосредственно или при помощи ссылок интерфейсы между разрабатываемыми и другими применяемыми продуктами. В этом случае должны быть разработаны процедуры, обеспечивающие четкое распределение системных требований между аппаратными и программными компонентами, а также соответствующие спецификации интерфейсов с внешней средой. При

заключении контракта спецификация требований может быть определена не полностью, она может быть доработана в ходе реализации проекта, и уточнены:

- требования к входной информации;
- источники информации и их идентификаторы;
- перечень и описание входных сообщений (идентификаторы, формы представления, регламент, сроки и частота поступления);
- перечень и описание структурных единиц информации входных сообщений или ссылка на документы, содержащие эти данные;
- описание и оценка преимуществ и недостатков разработанных альтернативных вариантов функций в концепции создания проекта;
- обоснование выбора оптимального варианта требований к содержанию и приоритетам функций компонентов;
- общие требования к структуре, составу компонентов и интерфейсам с внешней средой.

Требования к назначению и характеристикам функциональной пригодности должны выполняться для любых классов программных средств. Однако номенклатура учитываемых требований к конструктивным характеристикам качества существенно зависит от назначения и функций комплексов программ. Так, например, при проектировании:

- систем управления объектами в реальном времени наибольшее значение имеет время реакции (отклика на задание), защищенность, корректность и надежность функционирования стабильного комплекса программ и менее важно может быть качество обеспечения сопровождения и конфигурационного управления, способность к взаимодействию и практичность;
- для административных систем, кроме корректности, важно обеспечивать практичность применения, комфортное взаимодействие с пользователями и внешней средой, и может не иметь особого значение эффективность использования вычислительных ресурсов и обеспечение мобильности комплекса программ;
- для пакетов прикладных программ вычислений или моделирования процессов можно не всегда учитывать их мобильность, защищенность и временную эффективность, но особенно важна может быть корректность и точность расчетов.

При планировании, разработке и реализации **требований к характеристикам программного средства** необходимо, в первую очередь, учитывать следующие основные факторы [9, 12]:

- функциональную пригодность (функциональность) конкретного проекта программного продукта;
- возможные конструктивные характеристики качества комплекса программ, необходимые для улучшения функциональной пригодности;
- доступные ресурсы для создания и обеспечения всего жизненного цикла программного средства с требуемым качеством.

Эти факторы целесообразно применять последовательно, итерационно на каждом из основных этапов жизненного цикла комплекса программ. При первоначальном определении требований к функциональной пригодности и к конструктивным характеристикам качества, заданные заказчиком ограничения ресурсов не всегда могут учитывать ряд особенностей проекта, что обусловит недопустимое снижение (или завышение) требований к некоторым характеристикам качества. Кроме того, возможно, что некоторые характеристики противоречивы или принципиально нереализуемы в данном проекте. В результате **не сбалансированные требования** к качеству и доступные ресурсы проявятся как риски – ущерб в виде потерь в качестве или в перерасходовании ресурсов. Для устранения или снижения рисков до допустимых пределов потребуется изменение требований к функциональной пригодности и/или к конструктивным характеристикам. Таким образом, целесообразно анализ и разработку требований к качеству ПС проводить в два этапа: предварительно **максимизируя функциональную пригодность** и конструктивные характеристики качества, а затем **минимизируя риски** снижения требуемого качества или используемых ресурсов.

Типы и источники дефектов и ошибок в комплексах программ

Понятие дефекта или ошибки в программе, в общем случае, подразумевает неправильность, погрешность или неумышленное искажение объекта или процесса, что может быть **причиной ущерба – риска** при функционировании и применении программного продукта. При этом должно быть **известно или задано правильное, эталонное состояние объекта или процесса**, по отношению к которому может

быть определено наличие отклонения – ошибка или дефект. Исходным эталоном обычно является спецификация требований заказчика или потенциального пользователя, предъявляемая к программному продукту. Подобный документ устанавливает состав, содержание и значения результатов применения программы, которые должен получать пользователь при определенных условиях и исходных данных (рис. 2.2).

Типы и источники дефектов и ошибок в комплексах программ:

- характеристики дефектов и ошибок в комплексах программ:
 - понятие дефекта и ошибки в программе;
 - отсутствие полностью определенной программы-эталона;
 - устранения первичных ошибок, на основе их вторичных проявлений;
 - организационные дефекты при определении и использовании требований к программному продукту
 - системные ошибки и недостатки требований к программному продукту;
 - системные ошибки корректности выполнения требований к программному продукту;
 - ошибки проектирования и производства алгоритмов программных компонентов и модулей;
 - отчеты специалистов о выявленных дефектах и предложениях по корректировке комплекса программ;
- уровни ошибок сложных поставляемых программных продуктов, как ориентиры для прогнозирования длительности производства новых продуктов высокого качества.

Рис. 2.2

Любое отклонение результатов функционирования программы от предъявляемых к ней требований и сформированных по ним эталонов-тестов, следует квалифицировать как **ошибку – дефект в программе**, наносящий некоторый ущерб. Различия между ожидаемыми и полученными результатами функционирования комплекса программ могут быть следствием ошибок не только в созданных программах, но и ошибок в первичных требованиях спецификаций, явившихся базой при создании эталонов. Тем самым проявляется объективная реальность, заключающаяся в невозможности абсолют-

ной корректности и полноты исходных требований и эталонов для проектов крупных программных продуктов.

Изучение характеристик ошибок непосредственно связано с достигаемой корректностью, безопасностью и надежностью функционирования комплексов программ и **помогает**:

- оценивать реальное состояние проекта и планировать необходимую трудоемкость и длительность для его завершения и устранения ошибок;
- выбирать методы и средства автоматизации тестирования и отладки комплекса программ, адекватные текущему состоянию разработки и сопровождения, наиболее эффективные для устранения определенных видов дефектов;
- рассчитывать необходимую эффективность контрмер и дополнительных средств оперативной защиты от потенциальных дефектов и не выявленных ошибок;
- оценивать требуемые ресурсы ЭВМ по расширению памяти и производительности с учетом затрат на реализацию контрмер при модификации программ для устранения дефектов и ошибок.

На практике исходные требования – эталоны поэтапно уточняются, модифицируются, расширяются и детализируются по согласованию между заказчиком и разработчиком. Базой таких уточнений являются **неформализованные представления и знания** специалистов-заказчиков и разработчиков, а также результаты промежуточных этапов проектирования и тестирования. Однако установить некорректность таких эталонов еще труднее, чем обнаружить дефекты в программах, так как принципиально отсутствуют точные, формализованные данные, которые можно использовать как исходные.

В процессе жизненного цикла комплекса программ его требования подвергаются декомпозиции на спецификации программных и информационных компонентов. Эти спецификации рассматриваются как частные эталоны для составных частей комплекса, однако они редко бывают абсолютно полными и корректными. В процессе декомпозиции и верификации исходной спецификации требований, возможно появление ошибок в спецификациях на группы программ и на отдельные компоненты. Это способствует расширению спектра возможных дефектов и вызывает необходимость создания **гаммы методов и средств тестирования** для выявления некорректностей в

спецификациях и компонентах разных уровней детализации комплексов программ.

Важной особенностью процесса выявления ошибок в программах является *отсутствие полностью определенной программы-эталона*, которой должны соответствовать результаты функционирования разрабатываемой программы. Поэтому установить наличие и локализовать дефект в программе непосредственным ее сравнением с эталонной программой абсолютно без ошибок невозможно. При отладке и тестировании обычно сначала обнаруживаются *вторичные* ошибки, т.е. последствия и результаты проявления некоторых внутренних дефектов или некорректностей требований или функционирования программ (рис. 2.3).

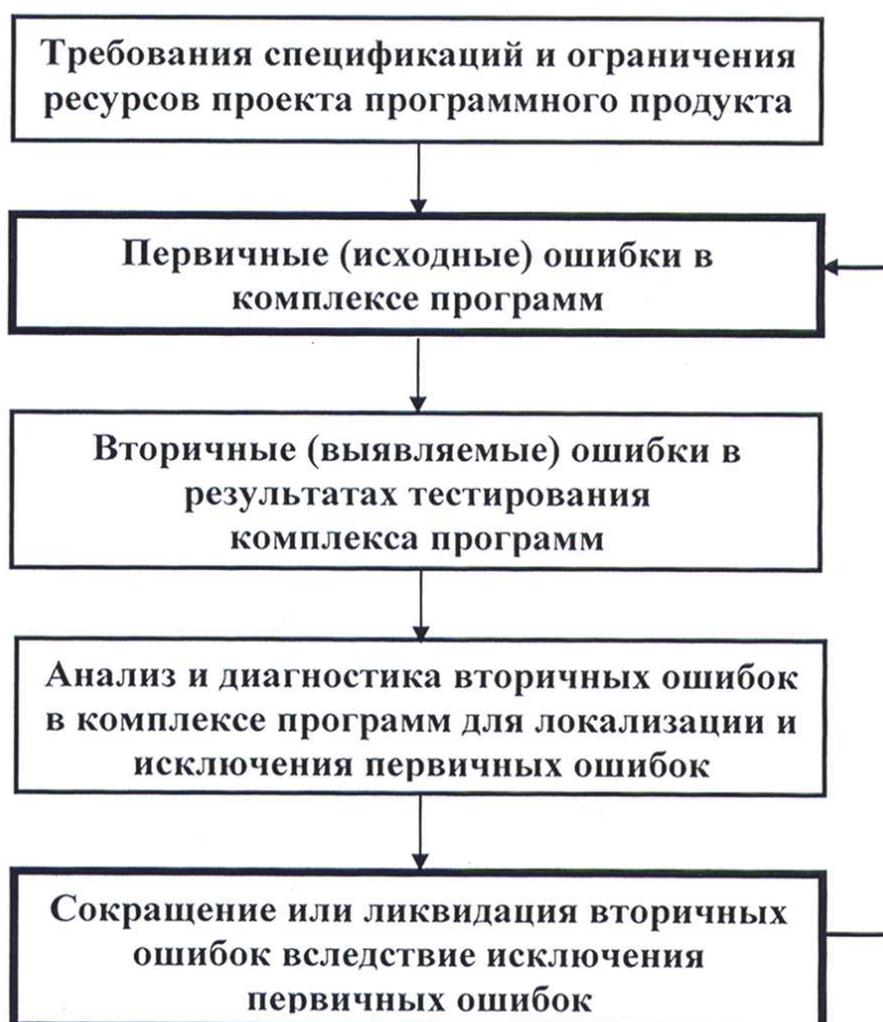


Рис. 2.3

Эти внутренние **дефекты** следует квалифицировать как источники – **первичные** ошибки или причины обнаруживаемых дефектов результатов. Последующая локализация и корректировка таких первичных ошибок должна приводить к устранению ошибок, первоначально выявляемых в результатах функционирования программ.

Потери эффективности и качества программ за счет неполной корректности в первом приближении можно считать прямо пропорциональными (с коэффициентом) вторичным ошибкам. Типичным является случай, когда одинаковые по величине и содержанию вторичные ошибки существенно различаются по своему воздействию на общую эффективность и последствия применения комплекса программ. Оценка последствий, отражающихся на вторичные ошибки при функционировании программ, может, **в принципе**, производиться по значениям ущерба – риска вследствие не устраненных их причин – первичных ошибок в программе. Вторичные ошибки являются определяющими для эффективности функционирования программ, однако не каждая первичная ошибка вносит заметный вклад в выходные результаты. Вследствие этого ряд первичных ошибок может оставаться не обнаруженным и, по существу, не влияет на функциональные характеристики комплексов программ.

Ошибкам в программах, естественно, предшествует их обнаружение и устранение на **основе вторичных проявлений**. Наибольшее число первичных ошибок вносится на этапах системного анализа, разработки или модификаций программ. При этом на долю системного анализа приходится наиболее сложные для обнаружения и устранения дефекты. Общие тенденции состоят в быстром росте затрат на выполнение каждого устранения ошибки на последовательных этапах процессов разработки комплекса программ. При системном анализе интенсивность обнаружения ошибок относительно не велика, и ее трудно выделять из процесса проектирования и разработки комплекса программ. Интенсивность проявления и обнаружения вторичных ошибок наиболее велика на этапе активного тестирования и автономной отладки программных компонентов. Затем она снижается приблизительно экспоненциально. Различия интенсивностей **устранения первичных ошибок на основе их вторичных проявлений** и внесения первичных ошибок при корректировках программ определяют скорость достижения заданного качества версий программного продукта.

Отчеты об ошибках – экспертные оценки, инспекции, быстрые просмотры – мощные инструменты для раннего обнаружения проблем в продуктах и документах любых видов, включая ошибки в программах. Полезный отчет об ошибке должен включать краткое описание, которое показывает суть и значение проблемы специалистам по управлению изменениями или комиссии по разбору ошибок. Краткое описание неопределимо при расстановке приоритетов ошибок для их устранения. Если пропускать этот этап как слишком затратный, чтобы быстрее ввести отчет об ошибке в систему управления дефектами, следует учитывать задержки и возможные срывы планов, которые могут произойти, когда разработчик потратит много времени на воспроизведение ситуации для устранения проявления ошибки.

Подготовка **полезного отчета об ошибке** должна начинаться с надежного, хорошо организованного тестирования. Воспроизведение проблемы (дефекта) обостряет ее понимание и позволяет фиксировать последовательность шагов, которые заново приведут к проявлению ошибки. Иногда появляются ошибки, которые не просто воспроизводятся повторным тестированием и с большим трудом поддаются исправлению. На поведение программы могут влиять многие факторы, а некоторые из них воздействуют на симптомы ошибки. Точная локализация ошибки способствует росту доверия к тестирующему и дает преимущества при отладке. Попытка найти общее правило возникновения ошибок углубляет представление о системе и позволяет убедить разработчиков и менеджеров в том, что некоторая проблема не является единичным случаем. Отчеты об ошибках после того, как их занесли в систему управления дефектами, обычно остаются там навсегда и могут быть прочитаны людьми, не участвующими в проекте.

Источниками ошибок в комплексах программ являются специалисты – конкретные люди с их индивидуальными особенностями, квалификацией, талантом и опытом (таблица 2.1).

При этом можно выделить предсказуемые дефекты требований, модификаций, расширения и совершенствования ПС, и необходимые изменения, обусловленные выявлением случайных, непредсказуемых дефектов и ошибок. Вследствие этого, плотность потоков и размеры необходимых корректировок в требованиях к комплексу и компонентам программ могут различаться в десяток раз.

Таблица 2.1

Специалисты – источники и типы первичных дефектов и ошибок

Специалисты – источники дефектов и ошибок	Типы первичных дефектов и ошибок программного средства и документации
Заказчик проекта программного продукта	Дефекты организации проекта и исходных требований заказчика на программный продукт
Менеджер проекта программного продукта	Дефекты, обусловленные реальной сложностью программного продукта
Менеджер-архитектор комплекса программ	Ошибки организации проектирования и архитектуры комплекса программ
Проблемно-ориентированные аналитики и системные архитекторы	Системные и алгоритмические дефекты и ошибки комплекса программ
Спецификаторы компонентов комплекса программ	Алгоритмические ошибки компонентов и документов комплекса программ
Системные интеграторы комплекса программ	Системные ошибки и дефекты реализации версий программного продукта и документации
Тестирующие требования и реализации компонентов и комплекса программ	Пропущенные системные, программные и алгоритмические ошибки компонентов, комплекса программ и документации
Управляющие сопровождением и конфигурацией версий программного продукта	Ошибки проектирования и реализации версий программного продукта
Документаторы программного продукта	Дефекты и ошибки технологических и эксплуатационных документов программного продукта

Однако в крупных комплексах программ статистика и распределение типов ошибок и выполняемых изменений для коллективов разных специалистов нивелируются и проявляются достаточно общие закономерности, которые могут использоваться как ориентиры при их выявлении. Каждому типу необходимых корректировок соответствует более или менее определенная категория специалистов, являющихся источником дефектов данного типа (см. таблицу 2.1). Такую корреляцию целесообразно учитывать как **общую качественную тенденцию** при анализе и поиске их причин. Этому могут помогать оценки типовых дефектов и корректировок путем их накопления и обобщения по опыту разработки определенных классов комплексов программ в конкретных предприятиях.

Одной из основных причин ошибок в комплексах программ являются **организационные дефекты при определении и использовании требований к программному продукту**, которые отличаются от остальных типов и условно могут быть выделены как самостоятельные (см. рис. 2.2). Ошибки и дефекты данного типа появляются из-за недостаточного понимания коллективом специалистов целей и функций комплекса программ, а также вследствие отсутствия четкой его организации и поэтапного контроля требований качества продуктов. Это порождается пренебрежением руководителей к организации всего технологического процесса формализации требований сложных программных продуктов и приводит к серьезной недооценке их дефектов, а также трудоемкости и сложности их выявления. При отсутствии планомерной и методичной разработки и тестирования требований может оставаться не выявленным значительное количество ошибок и, прежде всего, дефекты требований к взаимодействию отдельных функциональных компонентов между собой и с внешней средой. Для сокращения этого типа массовых ошибок активную роль должны играть лидеры – менеджеры и аналитики - системотехники, способные вести контроль и конфигурационное управление требованиями, изменениями и развитием версий и компонентов комплексов программ.

Ошибки оценки характеристик системы и внешней среды, принятых в процессе разработки комплекса программ за исходные, могут быть результатом аналитических расчетов, моделирования или исследования аналогичных систем. В ряде случаев может отсутствовать полная адекватность предполагаемых и реальных характеристик,

что является **причиной сложных и трудно обнаруживаемых системных ошибок и дефектов развития проекта**. Ситуация с такими ошибками дополнительно усложняется тем, что эксперименты по проверке взаимодействия программного продукта с реальной внешней средой во всей области изменения характеристик зачастую сложны и дороги, а в отдельных случаях, при создании опасных ситуаций, недопустимы. В этих случаях приходится использовать моделирование и имитацию внешней среды с заведомым упрощением ее отдельных элементов и характеристик, хотя степень упрощения не всегда можно оценить с необходимой точностью. Однако полной адекватности моделей внешней среды и реальной системы добиться трудно, а во многих случаях и невозможно, что может являться причиной значительного числа дефектов.

Системные ошибки и недостатки требований к программному продукту определяются, прежде всего, неполной информацией о реальных процессах, происходящих в источниках и потребителях информации [5, 18]. Кроме того, эти процессы зачастую зависят от самих алгоритмов и поэтому не могут быть достаточно определены и описаны заранее без исследования функционирования комплекса программ во взаимодействии с внешней средой. На начальных этапах разработки не всегда удается точно и полно сформулировать целевую задачу всей системы, а также целевые задачи основных функциональных групп программ, и эти задачи уточняются в процессе проектирования. В соответствии с этим уточняются и конкретизируются спецификации на функциональные компоненты и выявляются отклонения от уточненного задания требований, которые могут квалифицироваться как системные ошибки.

Во многих случаях отсутствует полная адекватность условий получения предполагаемых и реальных характеристик внешней среды, что может являться причиной сложных и трудно обнаруживаемых ошибок. Это усугубляется тем, что очень часто невозможно заранее предусмотреть все разнообразие возможных внешних условий и реальных вариантов сценариев функционирования и применения версий программных продуктов. При автономной и в начале комплексной отладки версий компонентов, относительная доля системных ошибок может быть невелика (около 10%), но она существенно возрастает (до 35 – 40%) на завершающих этапах комплексной отладки новых версий программного продукта. В процессе сопровождения

системные ошибки обычно являются преобладающими (около 60 – 80% от всех ошибок).

Системные ошибки корректности выполнения требований к программному продукту считаются наиболее критичными для общего успеха его версий и системы в целом. Ошибки требований являются наиболее трудными для обнаружения и наиболее сложными для исправления. Вот почему исправление ошибок выполнения требований может быть в 15 – 70 раз дороже, чем ошибок программирования. Требование может быть пропущено в спецификации к системе и комплексу программ. Это ведет к неудовлетворенности пользователя, и комплекс программ считается заказчиком и пользователем дефектным. Пропуск части требований – наиболее обычная проблема среди ошибок требований. Ошибки требований могут также представлять собой конфликтующие требования в спецификации. Может проявляться неопределенность требований – такой способ формулирования требования, что даже если и не конфликтует с другим требованием, оно выражено недостаточно ясно, чтобы привести к единственному, конструктивному решению при разработке комплекса программ. Конечный пользователь часто называет это ошибкой, хотя на самом деле это выбор конструктивного решения на основе неполного или неопределенного требования. Многочисленные исследования показали, что *ошибки требований труднее всего обнаруживать и дороже всего исправлять*.

Ошибки проектирования и разработки архитектуры программного комплекса определяются процессами перевода неопределенных и общих положений, сделанных на этапе первичных спецификаций требований, в более точные технические описания сценариев того, как программный продукт и система должны работать. Ошибки структуры инспекциями легче обнаружить, чем ошибки требований, но они в конечном итоге могут оказаться при корректировках столь же дорогостоящими. Главная причина того, что ошибки структуры дорого исправлять, состоит в том, что они могут влиять на систему в целом. Ошибки архитектуры можно разделить на **три категории**: пропуски, конфликты и ошибки перевода. Пропуски означают неспособность включить изменения одного или более требований в окончательную структуру комплекса программ. Когда пропуск новой функции или компонента попадает в окончательную структуру, он отражается ошибкой в конечном программном продук-

те. Ошибки, которые основаны на конфликтах на этом уровне, часто невозможно исправить без полного исправления версии программного продукта. Ошибки перевода проявляются, когда требования интерпретируются неправильно, по крайней мере, с точки зрения конечного пользователя. Если разработчик структуры либо неверно прочитает требования, либо не увидит содержание требования так же, как заказчик или конечный пользователь, то появится ошибка разработки структуры данного компонента или комплекса программ.

Первичные **ошибки в компонентах и модулях комплексов программ** можно анализировать с разной степенью детализации и в зависимости от различных факторов. Практический опыт показал, что **наиболее существенными факторами, влияющими на характеристики обнаружения ошибок являются:**

- методология, технология и уровень автоматизации системного и структурного проектирования комплексов программ, а также непосредственного программирования компонентов;
- длительность с начала процесса отладки и текущий этап разработки и модификации комплекса программ;
- класс комплекса программ, масштаб (размер) и типы компонентов, в которых обнаруживаются ошибки;
- методы, виды и уровень автоматизации тестирования, их адекватность требованиям и характеристикам компонентов и потенциально имеющимся в программах ошибкам;
- виды и достоверность эталонов – требований, которые используются для обнаружения ошибок.

Для организации эффективного тестирования конкретного комплекса программ целесообразно унифицировать регистрацию ошибок и дефектов на основе шаблона типовых **отчетов специалистов о выявленных дефектах и предложениях** по корректировке комплекса программ, включающих:

- идентификатор специалиста, представившего отчет о дефектах и/или предложениях;
- дату фиксирования дефекта или предложения на изменение комплекса программ;
- номер и параметры адаптации пользовательской версии программного продукта, на которой обнаружен дефект;
- идентификацию и регистрацию: аномального функционирования программного продукта, несогласованности процессов и ре-

зультатов с требованиями, планами и стандартами разработки, недостатки документации;

- описание ошибки и/или дефекта, достаточное для его понимания и устранения, описание возможных корректирующих действий, предназначенных для устранения зарегистрированного дефекта;
- подробное описание сценария и исходных данных, при которых выявлен дефект и документы результатов его регистрации;
- предположение о причине, вызвавшей проявление ошибки и/или дефекта;
- описание возможных корректирующих действий, предназначенных для устранения зарегистрированного дефекта;
- тесты, исходные данные и сценарий, при которых повторно проявляется выявленный дефект;
- результаты анализа и рекомендации о возможных способах устранения дефекта или о реализации предложения по совершенствованию комплекса программ;
- оценки сложности, трудоемкости, эффективности и срочности корректировки комплекса программ для устранения дефекта;
- оценки влияния предлагаемых изменений на возможность эксплуатации версий программного продукта, имеющихся у пользователей.

Исследования ошибок 20 крупных завершённых и поставляемых программных продуктов, созданных в 13 различных организациях, показали, что коллективы специалистов добились среднего уровня 0,06 дефекта на тысячу строк нового и изменённого программного кода. В пяти наибольших из них по размеру достигнуто 0,04 – 0,075 ошибок на тысячу строк. Таким образом, **уровень ошибок около 0,05 на тысячу строк кода** в разных публикациях считается близким к предельному уровню **для высококачественных программных продуктов**. Уровень ошибок критического проекта особенно высокого качества может служить программный продукт бортовых систем Шатла [19], созданный NASA. В нем содержится менее одной ошибки на 10000 строк кода. Однако стоимость программного продукта достигала 1000 \$ за строку кода, что в среднем в сто раз больше, чем для административных систем, и в десять раз больше, чем для ряда критических систем реального времени.

Приведенные характеристики типов дефектов и количественные данные могут служить **ориентирами при прогнозировании возмож-**

ного уровня не выявленных ошибок при тестировании различных сложных программных продуктов высокого качества. Следующим логическим шагом процесса их оценивания может быть усреднение для большого числа проектов фактических данных о количестве ошибок **на конкретном предприятии**, приходящихся на тысячу строк кода, которые обнаружены в различных комплексах программ. Тогда в следующих проектах будет возможность использовать эти данные, в качестве меры количества ошибок, обнаружение которых следует ожидать при выполнении проекта с таким же уровнем качества, или с целью повышения производительности или для оценки **момента прекращения дальнейшего тестирования**. Развитие технологии обеспечения жизненного цикла сложных комплексов программ, и средств автоматизации проектирования приводит к изменению **уровня и интенсивности устранения ошибок**. Подобные оценки гарантируют **от избыточного оптимизма** при определении сроков необходимого тестирования при разработке графиков реализации комплексов программ с заданным высоким качеством. Непредсказуемость конкретных ошибок в программах приводит к целесообразности последовательного тестирования, методичного фиксирования и анализа возможности проявления любого типа дефектов и необходимости их исключения на наиболее ранних этапах разработки при минимальных затратах.

Любые **испытания ограничены допустимым количеством и объемом тестирования**, а также длительностью работы комиссии испытателей, поэтому **не могут гарантировать абсолютную проверку** программного продукта соответствию требованиям функций и характеристик. Для повышения достоверности определения и улучшения оценивания характеристик после внутренних или квалификационных испытаний некоторые типы комплексов программ целесообразно передавать пользователям на **опытную эксплуатацию в типовых условиях**. Это позволяет более глубоко оценить эксплуатационные характеристики созданного комплекса и устранить некоторые дефекты и ошибки. Опытную эксплуатацию целесообразно проводить разработчиками с участием испытателей-заказчиков и некоторых пользователей, назначаемых заказчиком. Результаты и характеристики качества опытной эксплуатации после внутренних испытаний могут учитываться при проведении заказчиком квалификационных испытаний для их сокращения.

Для заказчика и пользователей доминирующее значение могут иметь номенклатура и особенности реализации некоторых отдельных функций комплекса программ, которые, как правило, требуют наибольших затрат на тестирование и определяют основной эффект от применения программного продукта, а также потенциальный уровень спроса на рынке. Если затраты на разработку комплекса программ можно оценивать и прогнозировать с некоторой достоверностью, то эффективность применения и особенно будущий спрос на конкретную версию программного продукта со стороны пользователей априори оценить трудно. Такие оценки могут проводиться на основе специальных маркетинговых исследований и опыта эксплуатации аналогичных комплексов программ или достаточно близких их прототипов.

Часть 2

СЕРТИФИКАЦИЯ ПРОЦЕССОВ ПРОИЗВОДСТВА ПРОГРАММНЫХ СРЕДСТВ

Лекция 3

БАЗОВЫЕ СТАНДАРТЫ СЕРТИФИКАЦИИ УПРАВЛЕНИЯ ПРОИЗВОДСТВОМ ПРОГРАММНЫХ ПРОДУКТОВ

Принципы организации производства программных продуктов

Сертификация процессов производства программных продуктов предполагает проверку организации и управления на предприятии регламентированных и стандартизированных технологических процессов жизненного цикла комплексов программ. Поэтому необходимо освоить и понимать основные принципы этих производств.

*Методической основой технологии ЖЦ ПС, регламентирующей деятельность специалистов, является **типовой технологический процесс***. Он отражается набором этапов и операций в последовательности их выполнения и взаимосвязи, обеспечивающих упорядоченное ведение работ на всех стадиях от инициирования проекта и подготовки технического задания до завершения испытаний или применения версии программного продукта. Индустриализация технологий создания ПС базируется на стандартизации процессов разработки программ, их структурного построения и интерфейсов с операционной и внешней средой. Технические и управленческие проверки, анализ качества результатов промежуточных работ и компонентов, а также корректности их взаимосвязей должны обеспечивать руководителям и всем разработчикам *уверенность достижения требуемого конечного результата проекта* – рис.3.1.

**Принципы организации производства программных продуктов:
для подготовки к сертификации процессов производства
программных продуктов руководители и специалисты
должны освоить и понимать:**

- методические основы жизненного цикла и технологии производства программных средств, регламентирующих деятельность специалистов и типовой технологический процесс;
- концептуальные и организационные основы административного управления жизненным циклом и качеством ПС, на базе восьми принципов, которые декларированы в стандартах **ISO 9000:2000**;
- цель системного проектирования – подготовить, обосновать и согласовать замыслы и решения заказчика (потребителя) и разработчика (поставщика) создания программного средства и его качества;
- цель управления проектом – рациональное использование и предупреждение потери ресурсов путем сбалансированного распределения их по частным работам на протяжении всего жизненного цикла ПС;
- стратегическое планирование жизненного цикла проекта должно содержать долгосрочные цели развития и модификации комплекса программ определенного функционального назначения;
- методы и средства технико-экономического обоснования трудоемкости и длительности производства программного средства, а также числа необходимых специалистов;
- подготовку контракта (договора) на детальное проектирование или на весь жизненный цикл программного средства.

Рис. 3.1

В современных автоматизированных технологиях создания и совершенствования сложных ПС с позиции обеспечения их качества можно выделить базовые методы и средства, позволяющие:

- **создавать программные модули и функциональные *компоненты высокого, гарантированного качества***;
- ***предотвращать дефекты проектирования*** за счет систем обеспечения качества, эффективных технологий и инструментальных средств автоматизации всего жизненного цикла комплексов программ и баз данных;
- ***обнаруживать и устранять различные дефекты и ошибки*** проектирования, разработки и сопровождения программ путем вери-

фикации и систематического тестирования на всех этапах жизненного цикла ПС;

- **удостоверять достигнутые значения качества функционирования** программного продукта в процессе их испытаний и сертификации перед передачей в регулярную эксплуатацию пользователям.

Комплексное, скоординированное применение этих методов, стандартов и средств в процессе создания, развития и применения ПС позволяет исключать многие виды дефектов или значительно ослаблять их влияние. Тем самым уровень достигаемого качества ПС **становится предсказуемым и управляемым**, непосредственно зависящим от ресурсов, выделяемых на его достижение, а главное, от системы качества и эффективности технологии, используемых на всех этапах жизненного цикла ПС. Основными **целями** упорядочивания, регламентирования процессов и применения **стандартов в жизненном цикле программных средств** являются:

- снижение трудоемкости, длительности, стоимости и улучшение других технико-экономических показателей проектов программных продуктов;
- повышение качества разрабатываемых и/или применяемых компонентов и программных средств в целом при их проектировании, производстве, сопровождении и эксплуатации;
- обеспечение возможности расширять ПС по набору прикладных функций и масштабировать комплекс программ в зависимости от изменения размерности решаемых задач;
- обеспечение переносимости прикладных программ и данных между разными аппаратными и операционными платформами и повторного использования программных компонентов.

Методология обеспечения жизненного цикла ПС поддержана рядом методических документов и инструментальных средств программной инженерии, а также **формализована группой международных стандартов** (см. Приложение). Концептуальные и организационные основы административного управления жизненным циклом и качеством ПС определены в **восьми базовых принципах**, которые декларированы в стандартах **ISO 9000:2000** и составляют методическую основу технологических процессов в этих стандартах.

Принцип 1 – Ориентация предприятия-разработчика на потребителя-заказчика. – Предприятия зависят от своих потреби-

лей и, таким образом, должны понимать текущие и будущие потребности потребителей-заказчиков, удовлетворять их требования и стремиться превзойти их ожидания.

Принцип 2 – Лидерство-руководство. – Лидеры обеспечивают единство назначения и направления деятельности предприятия. Они должны создавать и поддерживать внутреннюю окружающую среду, в которой специалисты могут в полной мере участвовать в достижении стратегических целей предприятия.

Принцип 3 – Вовлечение персонала. – Люди составляют сущность предприятия на всех уровнях, и их полноценное участие в деятельности способствует применению их способностей на благо целей предприятия.

Принцип 4 – Процессный подход. – Желаемый результат достигается более эффективно, когда требуемые ресурсы и деятельность специалистов предприятия управляются как единый связанный процесс.

Принцип 5 – Системный подход к административному управлению. – Выявление и понимание задач и административное управление системой взаимосвязанных процессов для заданной стратегической цели, повышает эффективность и результативность предприятия.

Принцип 6 – Постоянное усовершенствование. – Непрерывное усовершенствование процессов и повышение качества продукции должно быть постоянной стратегической целью предприятия и его специалистов.

Принцип 7 – Подход к принятию решений, основанный на фактах. – Эффективные решения должны базироваться на анализе только реальных данных и достоверной информации.

Принцип 8 – Взаимовыгодные отношения с поставщиками. – Предприятие-пользователь и его поставщики-разработчики взаимозависимы, и взаимовыгодные отношения между ними повышают способность обоих производить качественную продукцию.

Выполнение этих принципов способствует повышению управленческой культуры, применению системы административного управления качеством во всех видах деятельности предприятий и, как следствие, обеспечению высокого качества и конкурентоспособности создаваемой продукции, проектов и систем.

Каждый из этих принципов рекомендуется применять при:

- формулировке политики и стратегии обеспечения всего ЖЦ ПС;
- выборе целей проекта и плановых характеристик качества ПС, непосредственно связанных с потребностями и ожиданиями заказчиков и потребителей;
- управлении операциями в процессе реализации проекта для удовлетворения требований заказчика и потребителей;
- управлении ресурсами и специалистами предприятия для обеспечения ЖЦ ПС и его качества.

В процессе эксплуатации ПС у каждого пользователя могут появляться некоторые претензии к функционированию, которые квалифицируются им как **ошибки или дефекты** базовой или собственной, адаптированной версии. От пользователей или заказчиков могут поступать также предложения по внесению изменений в базовую версию для улучшения эксплуатационных характеристик и расширения функциональных возможностей ПС. Аналогичные предложения могут поступать от разработчиков комплекса программ. Для решения таких задач разработаны и активно применяются в жизненном цикле стандартизированные методы, методики и средства автоматизации регламентированного **сопровождения и управления конфигурацией**. Они позволяют представить отдельным специалистам и руководителям, состояние проекта и его компонентов в любой момент времени и не допускать хаоса при коллективной модификации программ и данных. Дисциплина сопровождения и конфигурационного управления, а также ее соблюдение, в значительной степени, определяют технико-экономические показатели жизненного цикла сложного проекта, его качество, длительность применения и конкурентоспособность программного продукта. При этом основу этих процессов управления могут составлять стандарты менеджмента качества жизненного цикла систем – **СММІ**, административного управления системой качества – **ISO 9000**. **Жизненный цикл ПС** целесообразно определять на основе подмножества процессов, работ и задач стандарта **ISO 12207** (см. лекцию 4), выбирая их с учетом характеристик конкретной системы.

Для управления проектом системы и/или программного средства, прежде всего, должны быть адекватно описаны **цели и объект проектирования**. Для сложных систем формализация и детализация характеристик жизненного цикла объекта происходит одновременно

с процессом его проектирования. Последовательно уточняются архитектура объекта, основные функции и их характеристики, требующиеся показатели качества функционирования и методы решения задач. Все эти данные отражаются в концепции, техническом задании, спецификации требований и предварительном описании проекта, которые детализируются и конкретизируются по мере его развития. Это определяет принципиальную особенность планирования проектов сложных систем, состоящую в наличии влияния на план изменяющихся значений и достоверности знаний разработчиков о требуемых характеристиках объекта разработки. С этим связана необходимость итерационного уточнения планов и концепции на всех этапах проектирования, разработки и совершенствования систем.

Цель управления проектом – рациональное использование и предупреждение потери ресурсов путем сбалансированного распределения их по частным работам на протяжении *всего жизненного цикла* объекта с заданным качеством. Управление проектом – это особый вид деятельности, включающий постановку задач, подготовку решений, планирование, организацию и стимулирование специалистов, контроль хода работ и использования ресурсов при создании сложных систем. Задачи целевого управления такими работами – сводить воедино усилия исполнителей – специалистов разной квалификации, подрядчиков и субподрядчиков, добиваясь, чтобы они **выступали как команда**, а не как разрозненная группа функциональных специалистов при создании компонентов систем. В результате должны обеспечиваться концептуальная целостность системы и высокое качество решения главных задач при сбалансированном использовании ресурсов на все функциональные задачи.

Методологической базой целевого планирования и управления жизненным циклом проекта является системный анализ, который предполагает:

- обследование объектов и среды проектирования для предварительной формализации целей, назначения и задач проекта;
- исследование и сопоставление альтернативных действий, которые должны приводить к достижению поставленных целей проектирования;
- сравнение альтернатив по величине достигаемого эффекта проекта в зависимости от затрат на его достижение (желательно, по показателю «эффективность/стоимость»);

- учет и анализ влияния неопределенностей характеристик альтернатив, определяющих их выбор, на эффект реализации проекта.

Чтобы найти и проанализировать все разумные альтернативы, обычно недостаточно одного специалиста, и необходимо участие в системном анализе специалистов разной квалификации. Не во всех системах и задачах оказывается доступным точный количественный подход. Во многих, чаще всего особенно сложных случаях, приходится ограничиваться качественным анализом свойств объекта, факторов и их влияния на конечный результат и возможный эффект проекта. Поэтому оптимизация решений и выбора альтернатив может ограничиваться оценкой логических суждений экспертов.

План проекта должен отражать **рациональное сочетание целей, стратегий действий, конкретных процедур, доступных ресурсов и других компонентов**, необходимых для достижения поставленной **основной цели создания продукта с заданным качеством**. Планирование жизненного цикла проектов должно обеспечивать компромисс между требующимися характеристиками создаваемой системы и ограниченными ресурсами, необходимыми на ее разработку и применение. По мере уточнения исходных данных об объекте разработки, внешней среде применения и ресурсах, в процессе системного анализа и проектирования возрастает достоверность планирования. При этом определяющими являются организация, стимулирование и контроль развития проекта. Контроль является органической функцией управления и должен иметь средства регулирования поведения отдельных личностей и коллектива проектировщиков в целом.

Для интеграции усилий специалистов и эффективного использования ресурсов проекта **должен выделяться руководитель – лидер, управляющий проектом**. Все ресурсы и исходные данные, необходимые для эффективного выполнения проекта, лидер получает от заказчика и функциональных подразделений и специалистов. Задача управляющего проектом наряду с прямыми воздействиями на подчиненных и координацией их работ – стимулировать и контролировать активность прямых связей между исполнителями частных работ. Для того чтобы процесс достижения целей был рациональным, лицо, принимающее решение (управляющий), должно иметь выбор среди альтернативных действий, ведущих к цели [9, 12].

Для получения, накопления и применения достоверных данных об объектах управления и альтернативах **необходима информацион-**

ная система обеспечения жизненного цикла проекта. Такая информационная система представляет собой комплекс формальных и неформальных каналов обмена информацией между участниками проекта, ее накопления и обработки. Степень формализации может варьироваться от утверждаемых руководителями планов и подробных технических заданий до личных бесед между специалистами. Таким образом, целевое управление проектами позволяет планировать, контролировать и анализировать информацию о состоянии и тенденциях изменения объекта разработки, его качестве и затраченных ресурсах. При этом должны **сохраняться основные стратегические цели проекта и главные процессы их достижения.** Это позволяет рассматривать альтернативы технических решений и предотвращает от сосредоточения внимания на частных задачах или вариантах решений, которые кажутся полезными и интересными, но мало отражаются на достижении главной цели.

Основная цель системного проектирования – подготовить, обосновать и согласовать замыслы и решения заказчика (потребителя) и разработчика (поставщика) о необходимости, направлениях и концепции создания или модернизации существующего ПС и изменениях его качества на **базе современных стандартов.** Методы и средства системного проектирования должны подготавливать эффективную технологическую базу для обеспечения всего жизненного цикла ПС требуемого качества. Характеристики качества комплексов программ должны анализироваться и формулироваться в начале их жизненного цикла и определять эффективность всех последующих процессов. Результатом этих работ должны быть системный проект, техническое задание и контракт на продолжение проектирования или решение о его нецелесообразности и/или прекращении. В системном проекте должны быть обобщены и отражены следующие основные **результаты выполненных исследований и разработок:**

- обобщенный анализ проведенного обследования объекта информатизации, функций существующей системы, качества ее основных программных компонентов и базы данных;
- совокупность предварительных исходных требований к функциям и характеристикам качества создаваемого комплекса программ;
- оценки имеющихся и потенциально доступных ресурсов (финансовых, вычислительных средств, специалистов) для обеспе-

чения всего жизненного цикла и требуемого качества проекта комплекса программ;

- результаты предварительного анализа архитектуры комплекса программ на основе моделей и прототипов аналогичных систем, позволяющие наметить планы разработки и всего жизненного цикла проекта ПС;

- цели, задачи и функции предполагаемой новой или модернизированной системы, обобщенные в *концепции* создания соответствующего программного средства;

- проекты планов жизненного цикла, гарантирования требуемого качества ПС, защиты и обеспечения безопасности его функционирования;

- результаты технико-экономического обоснования целесообразности и основных направлений продолжения проектирования и всего ЖЦ ПС;

- предварительный план организации работ, требования к составу и квалификации специалистов для выполнения проекта и всего жизненного цикла ПС;

- проект формализованного *технического задания и спецификации требований к ПС*, а также предложения по его финансированию и обеспечению ресурсами;

- *системный проект*, обобщающий проведенные исследования и разработки, позволяющий заключить контракт между разработчиком и заказчиком на финансирование и продолжение проектирования и/или на весь жизненный цикл ПС.

Разработка исходных требований для технического задания на проект программного средства начинается с анализа результатов обследования объекта информатизации и оценки доступных ресурсов для реализации проекта. Эта деятельность требует специальной организации специалистов наивысшей квалификации и тесной совместной работы представителей заказчика и разработчика. Наличие обычно ряда неформализованных, неструктурированных и противоречивых содержательных требований заказчика и разработчика требует их совместной обработки, согласования и корректировки.

Предварительный анализ и моделирование процессов обработки данных при проектировании должны проходить этапы от простого установления базовых отношений между понятиями, че-

рез определение интерфейсов доступа и атрибутов, к проекту модели состояний и взаимодействий между реальными объектами и процессами ПС. Итеративный характер построения формализованного описания проекта системы предопределен изначально не только потому, что не удастся сразу получить непротиворечивое и полное описание из-за неясностей в исходном описании, но и потому, что сложную систему можно описывать только, начиная с основной части ее предметной области, которая затем постепенно расширяется и детализируется. Схемы потоков данных, потоков управления, сущность-связь и другие составляют комплекс удобных и гибких графических методов и средств описания систем, *облегчающих взаимопонимание* между разработчиками и заказчиками на разных уровнях детализации функций, качества и архитектуры ЖЦ ПС.

Одним из наиболее эффективных направлений сокращения затрат и повышения качества комплексов программ является активное использование методического, технологического, алгоритмического и программного задела из предшествующих проектов, которое может быть названо *прототипированием* в широком смысле слова [12]. Математические модели и прототипы различных компонентов и функций систем и ПС обеспечивают возможность применять готовые апробированные решения, а также выделять и исследовать принципиально новые методы и процессы для реализации их в ПС. Прототипирование позволяет наглядно представить заказчику и пользователю функции будущей системы, виды и динамику применения экранов, меню, отчетов и форм запросов, а также откорректировать их для развития ПС на всех этапах жизненного цикла. Для этого следует анализировать и выбирать прототипы комплексов программ, характеристики которых наиболее близки к создаваемой версии ПС и которые позволили бы получить в результате объекты с необходимыми характеристиками качества.

Стратегическое планирование жизненного цикла системы и ПС должно содержать долгосрочные цели развития комплекса программ определенного функционального назначения. Планы должны отражать предварительные проекты всего *будущего жизненного цикла ПС*, обеспечения их качества, защиты и безопасности функционирования, верификации и тестирования, управления конфигурацией и сопровождения. По этим данным руководителем разработки и заказчиком принимается решение о целесообразности

продолжения проектирования и осуществляется стратегическое планирование ЖЦ проекта, которое формализуется в проекте и в техническом задании на ПС. В процессе проектирования последовательно уточняются характеристики объекта и среды разработки, вследствие чего появляется возможность более полно и точно **спланировать и обосновать весь последующий жизненный цикл ПС**.

Такое постепенное повышение достоверности прогнозов жизненного цикла приводит к целесообразности оценки достигаемых значений качества по этапам и к возможности разработки укрупненного, поэтапного плана выполнения всего комплекса работ. Если необходимые требования к функциям и качеству ПС не могут быть удовлетворены при доступных ресурсах, технологиях и специалистах, то возможны решения по прекращению дальнейшей разработки проекта. Таким образом, **производится технико-экономическое обоснование проекта**, определяются приближенные значения трудоемкости и длительности всей разработки ПС, а также число необходимых специалистов, что позволяет оценить предварительный укрупненный план создания и всего ЖЦ ПС в заданных условиях, ресурсах и сроках.

Вследствие творческого характера большинства работ на этом этапе невозможно составить жесткий план их выполнения. Помочь может типовой перечень частных работ, представленный в **стандартах ЖЦ ПС** и ориентировочный график, иллюстрирующий их взаимосвязь. При этом следует подготовить **требования к документации и обеспечить их реализацию**, которая должна быть однозначной – написана в стандартизированных терминах, которые допускают только единственную интерпретацию, уточняемую, если необходимо, соответствующими комментариями [15]. Если заказчик удовлетворен результатами проектирования, то возможно оформление **акта завершения работ и утверждение системного проекта** комплекса программ с требуемыми характеристиками качества для новой или модернизированной системы, а также **контракта (договора) на детальное проектирование или на весь жизненный цикл ПС**.

Процессы управления проектами программных средств на основе модели – СММІ

Достижение высоких значений качества программ существенно зависит от качества – *зрелости* технологии и инструментальных средств, используемых разработчиками для обеспечения жизненного цикла ПС [16]. Уровень зрелости автоматизации, качество технологии и средств, используемых для поддержки всего жизненного цикла ПС, обычно сильно коррелирован с достигаемым качеством комплексов программ, а также с качеством средств автоматизации для их оценивания. Оценивание зрелости технологической базы ЖЦ позволяет прогнозировать возможное качество ПС и ориентировать заказчика и пользователей при выборе разработчика и поставщика для определенного проекта с требуемыми характеристиками. Поэтому определение уровня *зрелости* технологической поддержки процессов жизненного цикла, организационного и инструментального обеспечения ПС, непосредственно *связано с оцениванием реальных или возможных характеристик качества* конкретного комплекса программ. Значительные достижения в организации, планировании, развитии и применении современных методов и технологии обеспечения *проектов программных средств* сосредоточены в методологии и стандарте де-факто *Capability Maturity Model Integration (CMMI) for Systems Engineering/Software Engineering/Integrated Product and Process Development* – Интегрированная модель оценивания зрелости для инженерии систем/инженерии программных средств/интегрированных продуктов и процессов разработки.

Эту комплексную модель СММІ, уточняющую и совершенствующую предшествовавшие модели СММ, а также учитывающую основные *требования существующих международных стандартов* в области менеджмента программных средств американский институт программной инженерии (SEI) опубликовал в 2003 году. Внедрение этой модели акцентировано на улучшении процессов управления проектами ПС, обеспечении их высокого качества и конкурентоспособности, с основной целью – сделать процессы проектов более управляемыми, а результаты – предсказуемыми.

Назначение методологии СММІ – состоит в предоставлении необходимых общих рекомендаций и инструкций предприятиям и специалистам, производящим программные продукты, по выбору

стратегии совершенствования качества, процессов и продуктов путем анализа степени их производственной зрелости и оценивания факторов, в наибольшей степени влияющих на качество ЖЦ ПС, а также посредством выделения процессов, требующих модернизации. Для достижения устойчивых результатов в процессе развития технологии и организации управления жизненным циклом ПС рекомендуется использовать эволюционный путь, который позволяет совершенствовать и постепенно повышать качество процессов и продуктов, вскрывать преимущества и недостатки предприятия. **Уровни зрелости характеризуются** степенью формализации, адекватностью измерения и документирования процессов и продуктов ЖЦ ПС, широтой применения стандартов и инструментальных средств автоматизации работ, наличием и полнотой реализации функций системой обеспечения качества технологических процессов и их результатов.

Два варианта модели **СММІ** созданы для возможности **непрерывного** оценивания процессов в определенной области создания ПС и/или для **поэтапного** оценивания и совершенствования зрелости предприятия, а также организации его производства комплексов программ в целом. Модели **СММІ** представляют помощь проектировщикам при организации и совершенствовании их продуктов, по обслуживанию процессов разработки и сопровождения ПС. Компоненты непрерывной и поэтапной моделей в значительной степени подобны и могут выбираться и применяться в разном составе и последовательности использования в зависимости от свойств и характеристик проектов. Варианты моделей построены по единой схеме, которая содержит **основные разделы**:

4-й раздел – содержание уровней и главные компоненты каждого варианта модели (разработка целей и процедур);

5-й раздел – структура процедур – четыре категории **СММІ** процессов:

- менеджмент процессов;
- управление проектом;
- инженерия – технология;
- поддержка – сопровождение.

6-й раздел – использование **СММІ** моделей.

Последний, **седьмой раздел** самый большой в каждом варианте стандарта и занимает около 500 страниц из полного объема документа до 700 страниц. В этом разделе представлены **подробные рекомен-**

дации для реализации каждого из перечисленных выше четырех процессов, которые учитывают особенности конкретных моделей.

Первый вариант непрерывной модели отражает документ: **CMMI-SE/SW/IPPD, V1.1, Continuous** – Интегрированная модель оценивания зрелости для инженерии продуктов и процессов разработки – *непрерывное представление*. В этой модели *седьмой раздел содержит детализацию* процессов 5-го раздела – рис. 3.2.

Второй вариант представляет документ: **CMMI-SE/SW/ IPPD, V1.1, Staged** – Интегрированная модель оценивания зрелости для инженерии продуктов и процессов разработки – *поэтапное представление*. Модель базируется на концепции пяти уровней зрелости – рис.3.3.

Уровень 1 – Начальный. Массовые разработки проектов ПС характеризуются относительно небольшими объемами программ в несколько тысяч строк, создаваемых несколькими специалистами. Они применяют простейшие не формализованные технологии с использованием типовых инструментальных компонентов операционных систем. Основные процессы ЖЦ ПС на этом уровне не регламентированы, выполняются не совсем упорядоченно и зависят от не координированных индивидуальных усилий и свойств отдельных специалистов. Успех проекта, как правило, зависит от энергичности, таланта и опыта нескольких руководителей и исполнителей. Процессы на первом уровне характеризуются своей непредсказуемостью по трудоемкости и срокам в связи с тем, что их состав, назначение и последовательность выполнения могут меняться случайным образом в зависимости от текущей ситуации.

Уровень 2 – Управляемый – базовое управление. Для сложных проектов ПС объемом в десятки и сотни тысяч строк, в которых участвуют десятки специалистов разной квалификации, необходимы организация, регламентирование технологии и унификация процессов деятельности каждого из них. Процессы на этом уровне заранее планируются, их выполнение контролируется, чем достигается предсказуемость результатов и времени выполнения этапов, компонентов и проекта в целом. Основной особенностью второго уровня является наличие формализованных и документированных процессов управления проектами, которые пригодны для модернизации, а их результаты, поддаются количественной оценке.

Первый вариант непрерывной модели СММІ***Менеджмент процессов:***

- содержание (усовершенствование) организационных процессов;
- определение организационных процессов;
- обучение организации;
- организация процессов изменений;
- организация инноваций и расширений.

Управление проектом:

- планирование проекта;
- мониторинг и контроль процессов проекта;
- управление поставщиками;
- интегрированное управление процессами и продуктами проекта;
- управление рисками;
- интеграция команды разработчиков;
- количественное управление проектом.

Инженерия – технология:

- управление требованиями;
- разработка требований;
- технические решения;
- интеграция продукта;
- верификация;
- валидация – аттестация.

Поддержка – сопровождение:

- управление конфигурацией;
- обеспечение качества процессов и продуктов;
- измерение и анализ;
- анализ и принятие решений на изменения;
- организация окружения для интеграции;
- анализ причин и разрешение проблем – устранение дефектов.

Рис. 3.2

Второй вариант –
позапанная модель СММІ

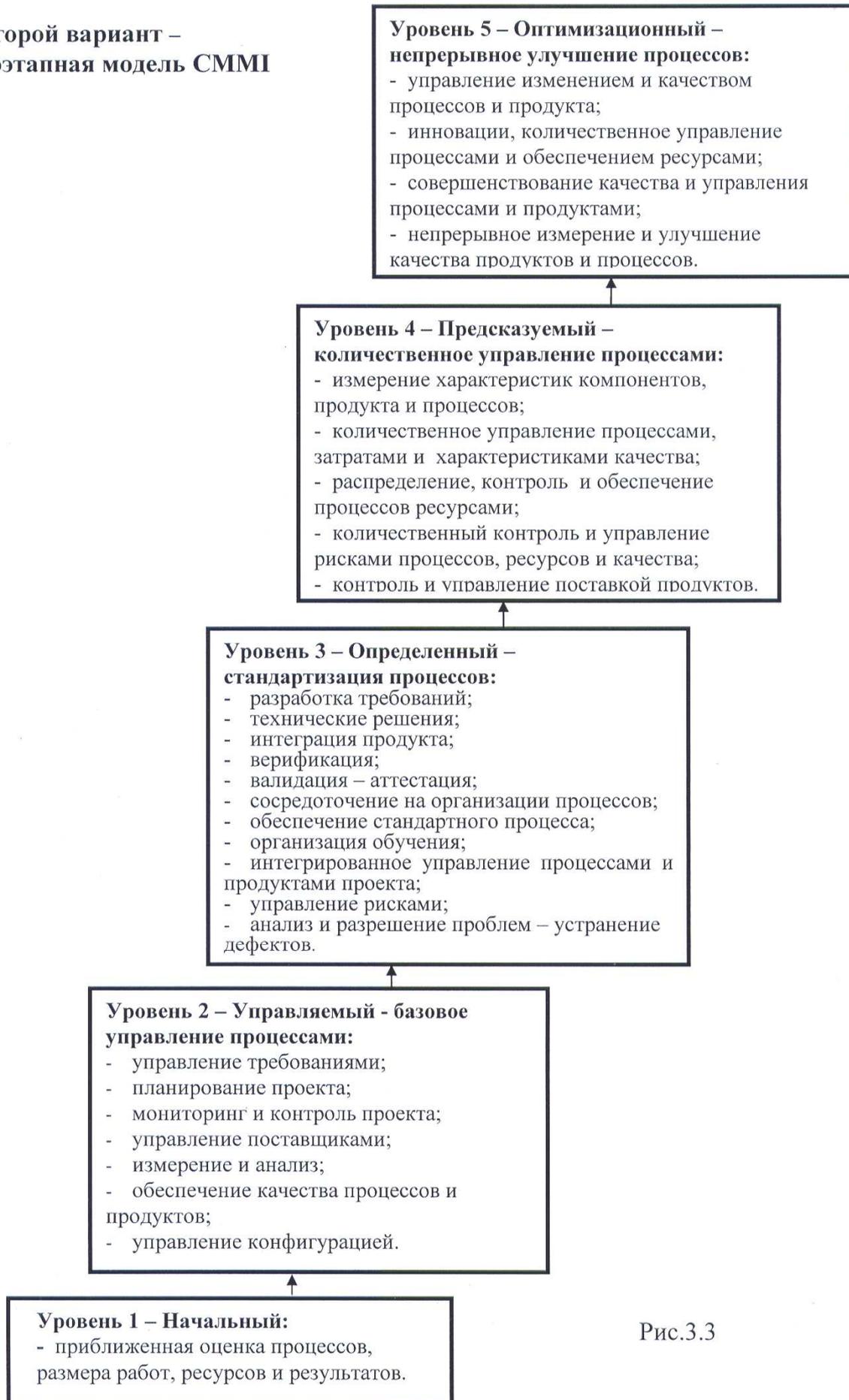


Рис.3.3

На этом уровне акценты управления сосредотачиваются на предварительном упорядочении и регламентировании процессов создания, сопровождения и оценивания качества программного средства, однако для крупномасштабных проектов ПС с гарантированным качеством, риск провала может оставаться еще достаточно большим.

Уровень 3 – Определенный – стандартизация процессов. При высоких требованиях заказчика и пользователей к конкретным характеристикам качества сложного ПС и к выполнению ограничений по использованию ресурсов, необходимо дальнейшее совершенствование и повышение уровня зрелости процессов ЖЦ ПС. Процессы ЖЦ ПС на этом уровне должны быть стандартизированы, и представлять собой единую технологическую систему, обязательную для всех подразделений. На основе единой технологии поддержки и обеспечения качества ЖЦ ПС, для каждого проекта могут разрабатываться дополнительные процессы последовательного оценивания качества продуктов с учетом их особенностей. Описание каждого процесса должно включать условия его выполнения, входные данные, рекомендации стандартов и процедуры выполнения, механизмы проверки качества результатов, выходные данные, условия и документы завершения процессов. В описания процессов включаются сведения об инструментальных средствах, необходимых для их выполнения, роль, ответственность и квалификация специалистов.

Уровень 4 – Предсказуемый – количественное управление. Для реализации проектов крупномасштабных, особенно сложных ПС в жестко ограниченные сроки и с высоким гарантированным качеством, необходимы активные меры для предотвращения и выявления дефектов и ошибок на всех этапах ЖЦ ПС. Управление должно обеспечивать выполнение процессов в соответствии с текущими требованиями к характеристикам качества компонентов и ПС в целом. На этом уровне, должна применяться система детального поэтапного оценивания характеристик качества, как технологических процессов ЖЦ, так и самого создаваемого программного продукта и его компонентов. Должны разрабатываться и применяться методики количественной оценки реализации процессов и их качества. Одновременно с повышением сложности и требований к качеству ПС следует совершенствовать управление проектами за счет сокращения текущих корректировок и исправлений дефектов при выполнении процессов. Результаты процессов становятся предсказуемыми по срокам и качеству

в связи с тем, что они измеряются в ходе их выполнения и реализуются в рамках заданных ресурсных ограничений.

Уровень 5 – Оптимизационный – непрерывное совершенствование и улучшение. Дальнейшее последовательное совершенствование и модернизация технологических процессов ЖЦ ПС для повышения качества их выполнения и расширение глубины контроля за их реализацией. Одна из основных целей этого уровня – сокращение проявлений и потерь от случайных дефектов и ошибок путем выявления сильных и слабых сторон используемых процессов. При этом приоритетным является анализ рисков, дефектов и отклонений от заданных требований заказчика. Эти данные также используются для снижения себестоимости ЖЦ особо сложных ПС в результате внедрения новых технологий и инструментария, а также для планирования и осуществления модернизации всех видов процессов. Технологические нововведения, которые могут принести наибольшую выгоду, должны стандартизироваться и адаптироваться в комплексную технологию обеспечения и оценивания системы качества предприятия и его продукции.

Первый и пятый уровни отличаются значительной нестабильностью и неопределенностью процессов в различных проектах, поэтому при уточнении и детализации содержания процессов рекомендуется ограничиваться **тремя основными уровнями** из четырех старших, представленных в поэтапном варианте СММІ (см. рис.3.3).

Рекомендуется на каждом более высоком уровне зрелости применять **все процессы** предыдущих нижних уровней. Упорядочение и оценка используемых процессов в соответствии с уровнями, позволяет устанавливать производственный потенциал предприятий – разработчиков программных продуктов по прогнозируемому качеству результатов их деятельности и возможности сертификации. Это уменьшает зависимость заказчиков и пользователей от возможных недостатков исполнителей проектов и позволяет их выбирать с учетом прогнозируемого качества продуктов.

Практически все перечисленные процессы и требования, конкретизированные на трех выделенных выше уровнях **модели СММІ соответствуют**, регламентированным и детализированным в стандартах **ISO 9001:2000, ISO 12207** основных компонентах **профиля стандартов жизненного цикла сложных ПС**. Стандарты СММІ и **ISO 9001:2000** во многом подобны по структуре и содержанию де-

тальных требований к организации и планированию ЖЦ ПС (рис. 3.4).

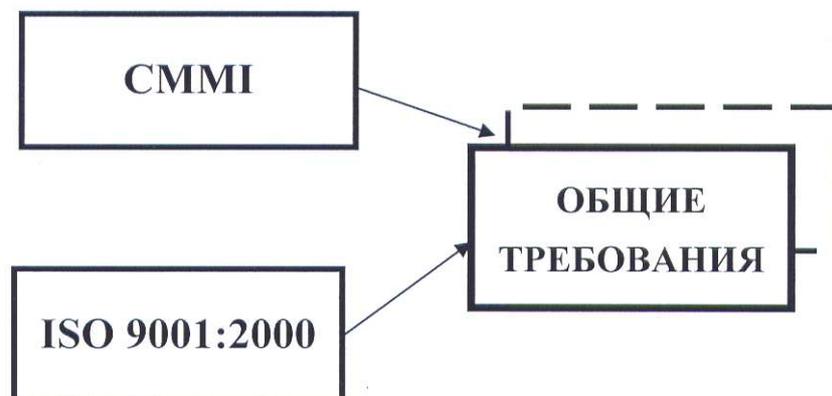


Рис. 3.4

Требованиям в функциональных разделах 4 – 8 стандарта **ISO 9001** могут быть сопоставлены подобные по содержанию разделы в *поэтапной модели CMMI*. Общность процессов и требований **CMMI** и **ISO 9001** состоит в подобии терминологии, структуры, рекомендуемых процессов управления, планирования, учета доступных ресурсов, оценивания организации специалистов. Некоторые требования в **ISO 9001** и тем более положения и рекомендации почти *всего профиля международных стандартов* не покрываются содержанием требований в моделях **CMMI**. При практической реализации и обеспечении *всего жизненного цикла сложных ПС* разработчикам и поставщикам целесообразно использовать полный *профиль стандартов* (см. Приложение), а для оценивания заказчиками уровня менеджмента, организационного и технологического обеспечения проектов ПС применять конкретные рекомендации **CMMI** или **ISO 9001:2000**.

Менеджмент – административное управление обеспечением качества систем на основе стандартов ISO 9000:2000

Серия стандартов ISO 9000:2000 разработана, чтобы помочь предприятиям всех типов и размеров внедрить и использовать эффективные *системы менеджмента (административного управления) качества*. Совместно они образуют комплект согласованных стандартов управления системами качества производства:

- **ISO 9000:2000** – представляет введение в системы управления качеством продукции и услуг и словарь качества;
- **ISO 9001:2000** – устанавливает детальные требования для систем управления качеством, достаточные в случае необходимости продемонстрировать способность предприятия обеспечить соответствие качества продукции и услуг требованиям заказчика;
- **ISO 9004:2000** – содержит руководство по внедрению и применению широко развитой системы управления качеством, чтобы достичь постоянного улучшения деловой деятельности и результатов предприятия.

Стандарты ISO 9000:2000 применяют процессный подход в административном управлении системами качества предприятий, а также рассматривают способы быстрого выявления и реализации возможностей для их улучшения. Процессная модель подчеркивает тот факт, что заказчики и другие заинтересованные стороны играют значительную роль в процессе установления исходных требований.

В **стандарте ISO 9004** детализированы руководящие указания и рекомендации по применению системы менеджмента качества, которые изложены в том же порядке, как требования в **ISO 9001**. Оба стандарта ссылаются на **ISO 9000**, который объясняет используемую терминологию и определения. Структура основных требований и рекомендаций в этих стандартах сведена к четырем объединенным крупным процессам:

- **обязанности и ответственность администрации управления качеством**: ориентация на потребителя, требования заказчика к качеству; политика обеспечения качества; цели и планирование качества; система административного управления качеством; анализ системы качества со стороны руководства (раздел 5);
- **административное управление ресурсами**: отбор и подготовка персонала по квалификации и компетентности; инфраструктура, производственная среда и инструментальные средства для обеспечения качества продукции (раздел 6);
- **процессы жизненного цикла продукции и управления ее качеством**: планирование процессов жизненного цикла; организация взаимодействия с заказчиком; проектирование и разработка; закупка; операции производства и обслуживания; контроль и измерение качества; обслуживание продукции после поставки (раздел 7);

- **измерения, анализ и совершенствование:** мониторинг корректировок; учет изменений требований заказчика; анализ и измерение характеристик объекта и процессов; процессы улучшения и корректирующие действия (раздел 8).

Стандарты серии **ISO 9000:2000** рекомендуется применять в деятельности предприятия, начиная от идентификации требований заказчика, и охватывать все процессы системы управления качеством, вплоть до достижения соответствия требованиям. Применение сокращенного, адаптированного варианта требований не освобождает предприятие от ответственности предоставлять продукцию, которая удовлетворяет всем требованиям заказчика. Система качества должна быть внедрена, поддерживаться в рабочем состоянии и подвергаться улучшениям со стороны специалистов предприятия. Масштаб и глубина процедур должна определяться такими факторами как размер и тип предприятия, сложность и взаимосвязь процессов, применяемые методы, а также квалификация и степень подготовки персонала, участвующего в выполнении работ. Они **должны включать:**

- общесистемные процедуры, которые описывают деятельность, необходимую для внедрения и применения системы качества;
- процедуры, описывающие последовательность и внутреннее содержание процессов, необходимых для обеспечения уверенности в соответствии продукции установленным требованиям;
- инструкции, описывающие операционную деятельность и управление процессами.

Стандарт ISO 9001:2000 – Система менеджмента (административного управления) качества. Требования – является базой для Руководства по их реализации в стандарте **ISO 9004:2000** – рис. 3.5. **ISO 9001** – «устанавливает требования к системе менеджмента качества, которые могут использоваться для внутреннего применения организациями в **целях сертификации или заключения контрактов**». **ISO 9004** – «содержит методические указания и рекомендации, но **не предназначен** для сертификации или использования в контрактах и регламентах».

Общие требования к системе менеджмента качества – установить и управлять процессами, необходимыми для обеспечения уверенности в том, что продукция и/или услуга соответствуют требованиям заказчика. В качестве способа внедрения и демонстрации, установленных процессов, организация должна создать систему ме-

неджмента качества, основываясь на требованиях настоящего международного стандарта.

Структура и основные разделы стандарта ISO 9001:2000

Основы системы менеджмента качества:

- общие требования;
- требования к документации.

Ответственность руководства:

- обязательства руководства;
- ориентация на потребителя;
- политика в области качества;
- планирование;
- ответственность, полномочия и обмен информацией;
- анализ со стороны руководства.

Менеджмент ресурсов:

- обеспечение ресурсами;
- человеческие ресурсы;
- инфраструктура;
- производственная среда.

Процессы жизненного цикла продукции:

- планирование процессов жизненного цикла продукции;
- процессы, связанные с потребителями;
- проектирование и разработка;
- закупки;
- производство и обслуживание;
- управление контрольными и измерительными приборами.

Измерение, анализ и улучшение:

- общие положения;
- измерения и мониторинг;
- управление несоответствующей продукцией;
- анализы данных в целях улучшения;
- улучшение.

Рис. 3.5

Система менеджмента качества должна быть внедрена, поддерживаться в рабочем состоянии и подвергаться улучшениям со стороны организации. Организация должна подготовить процедуры системы менеджмента качества, которые описывают процессы, необходимые для внедрения системы менеджмента качества. Масштаб и глубина процедур должна определяться такими факторами как размер и тип организации, сложность и взаимосвязь процессов, применяемые методы, а также квалификация и степень подготовки персонала, участвующего в выполнении работ.

Высшее руководство предприятия должно продемонстрировать свои обязательства относительно:

- создания и поддержания важности удовлетворения требований заказчика;
- разработки политики качества и целей в области качества, а также планирования качества;
- создания системы менеджмента качества;
- проведения анализа деятельности со стороны руководства;
- обеспечения уверенности в наличии ресурсов.

Требования заказчика – высшее руководство должно обеспечить, что:

- потребности и ожидания заказчика установлены и переведены в соответствующие требования, имеющие целью обеспечить доверие со стороны заказчика;
- требования заказчика полностью поняты и могут быть удовлетворены.

Высшее руководство должно разработать **политику в области качества** и обеспечить уверенность в том, что она:

- соответствует потребностям заказчиков;
- включает обязательства по удовлетворению требований и постоянному улучшению;
- обеспечивает основу для разработки и анализа целей в области качества;
- распространена, понята и внедрена во всей организации;
- анализируется с целью постоянного поддержания ее пригодности.

Планирование – предприятие должно установить цели в области качества для каждой соответствующей функции и для каждого уровня внутри организации. Цели в области качества должны соот-

ветствовать политике в области качества и обязательствам относительно непрерывного улучшения. Организация должна установить и планировать деятельность и ресурсы, необходимые для достижения целей в области качества. Такое планирование должно отвечать другим требованиям к системе менеджмента качества, а его результаты должны быть документированы. Планирование должно охватывать:

- процессы, необходимые в рамках системы менеджмента качества;
- процессы создания и необходимые ресурсы, а также установленные характеристики качества на различных стадиях с целью достижения планируемых результатов;
- деятельность по проверке, критерии приемлемости и необходимые отчеты по качеству.

Система управления качеством – предприятие должно создать систему менеджмента качества как средство реализации ее политики в области качества, достижения своих целей в области качества и обеспечения уверенности в том, что продукция и/или услуга отвечает требованиям заказчика. Роли сотрудников и их взаимосвязи, а также ответственность и полномочия персонала должны быть установлены для того, чтобы способствовать эффективному управлению качеством, и должны быть доведены до соответствующих уровней организации. Высшее руководство должно уполномочить одного (или нескольких) лиц для:

- обеспечения уверенности в том, что система менеджмента качества внедрена и поддерживается в рабочем состоянии в соответствии с требованиями настоящего международного стандарта;
- доклада высшему руководству о функционировании системы менеджмента качества, включая вопросы, связанные с необходимостью ее улучшения;
- обеспечения уверенности в осознании во всей организации требований заказчика.

Следует разработать **Руководство по качеству**, которое должно включать:

- описание элементов системы менеджмента качества и их взаимосвязей;
- общесистемные процедуры или соответствующие ссылки на них.

Предприятие должно установить общесистемные **процедуры для управления документами**, необходимыми для функционирования системы менеджмента качества, обеспечивающие уверенность в том, что:

- документы проверены на адекватность до их применения;
- документы анализируются, при необходимости уточняются и утверждаются;
- соответствующие выпуски документов находятся в тех местах, где осуществляется деятельность, имеющая существенное значение для эффективности функционирования системы менеджмента качества;
- устаревшие документы изъяты из всех мест их рассылки и применения или предприняты другие методы управления, предотвращающие их непреднамеренное использование;
- любые устаревшие документы, оставленные для юридических целей или в целях сохранения знаний, должным образом идентифицированы.

Должен быть составлен специальный перечень или применяться другая эквивалентная процедура управления, идентифицирующая статус текущей версии документов, которая была бы доступна в целях предотвращения использования недействительных и/или устаревших документов. Для демонстрации соответствия требованиям и эффективности функционирования системы менеджмента качества должны вестись подходящие для организации **отчеты о качестве**. Следует создать и поддерживать в рабочем состоянии процедуры общесистемного уровня по идентификации, хранению, восстановлению, обеспечению сохранности, установлению времени и места хранения отчетов о качестве. Анализ со стороны руководства должен через установленные периоды времени проводиться для обеспечения уверенности в сохранении ее пригодности, адекватности и эффективности.

Управление ресурсами необходимо для создания и поддержания в рабочем состоянии системы менеджмента качества (см. рис. 3.5). Предприятие должно проводить назначение персонала с целью обеспечения уверенности в том, что те, кто имеет обязанности, определенные системой менеджмента качества, являются компетентными для осуществления своей деятельности на основе соответствующего

образования, подготовки, мастерства и опыта, создать и поддерживать в рабочем состоянии общесистемные процедуры по:

- определению потребностей в компетентном персонале и в подготовке персонала;
- обеспечению подготовки персонала в соответствии с выявленными потребностями;
- оценке через установленный период времени эффективности подготовки кадров;
- ведению соответствующих отчетов об образовании кадров, их подготовке, уровне мастерства и опыта.

Предприятие должно создать и поддерживать в рабочем состоянии процедуры, обеспечивающие осознание ее работниками в соответствующих службах и на соответствующих уровнях – таблица 3.1:

- важности соответствия политике в области качества и требованиям к системе менеджмента качества;
- влияния их деятельности на качество – фактическое или потенциальное;
- выгоды от улучшения работы персонала;
- своей роли и ответственности в достижении соответствия политике в области качества и процедурам, а также требованиям к системе менеджмента качества;
- потенциальных последствий отклонений от установленных процедур.

Предприятие должно установить **перечень информации**, которая необходима для управления процессами, а также для обеспечения уверенности в соответствии продукции и/или услуги. Общесистемные процедуры по управлению информацией должны обеспечить уверенность в доступности и сохранности информации.

Следует определить, создать и поддерживать в рабочем состоянии **инфраструктуру**, необходимую для достижения соответствия продукции и/или услуги:

- рабочие места и соответствующие помещения;
- оборудование, вспомогательные средства и программное обеспечение;
- пригодные способы поддержания работоспособности;
- вспомогательные услуги.

Таблица 3.1

Типы изменения требований, право на их выполнение и утверждение

Объекты изменения требований	Типы изменений требований объектов	Право на выполнение изменения требований имеют:	Право на утверждение изменения требований имеют:
Требования к версиям модулей и компонентов программ и документов	Устранение дефектов в требованиях к программным модулям и компонентам	Программисты – разработчики требований к модулям и компонентам	Руководители разработки требований к функциональным группам программ
Требования к версиям функциональных групп программ и документов	Корректировка требований к функциям и взаимодействию программных компонентов	Руководители разработки требований к функциональным компонентам	Менеджер-архитектор требований к программному продукту
Требования к версии программного продукта и комплексу документации	Требования к модификации и улучшению функций и качества версии программного продукта	Менеджер-архитектор требований к программному продукту	Менеджер и заказчик проекта программного продукта
Версии требований к программному продукту пользователей	Адаптация требований к характеристикам внешней среды пользователей	Заказчик или сопровождающий версию программного продукта пользователей	Менеджер, сопровождающий версию программного продукта пользователей

Процессы, необходимые для **выпуска требуемой продукции**, их последовательность и взаимосвязи должны быть определены, спланированы и внедрены. При определении таких процессов предприятие должно учесть результаты планирования качества. Должна быть уверенность в том, что эти процессы осуществляются в управляемых условиях и их результаты **соответствуют требованиям заказчика**:

- установить (в необходимой степени) для этих процессов соответствующие методы и практику выполнения, необходимые для обеспечения постоянной работоспособности процесса;

- определить и внедрить (в необходимой степени) критерии и методы управления процессами для обеспечения соответствия продукции и/или услуги требованиям заказчика;
- удостовериться, что процессы могут функционировать в той мере, которая позволяет обеспечить соответствие продукции и/или услуги требованиям заказчика;
- определить и реализовать меры, обеспечивающие измерение, оценку и осуществление необходимых последующих воздействий, обеспечивающих уверенность в том, что процессы продолжают оставаться управляемыми с точки зрения достижения запланированных результатов;
- обеспечить уверенность в наличии информации и данных, необходимых для поддержания эффективного функционирования и мониторинга процессов;
- фиксировать в виде отчетов о качестве результаты измерений, осуществляемых в ходе управления процессом, для предоставления доказательств эффективного функционирования и мониторинга процессов.

Предприятие должно создать **процесс идентификации требований заказчика**:

- полноту требований заказчика к продукции;
- требования, не установленные заказчиком, но необходимые для целей соответствия;
- обязательства по отношению к продукции, включая регламентирующие и законодательные требования;
- требования заказчика относительно пригодности продукции и/или услуги, ее поставок и технического сопровождения.

Организация должна внедрить меры по **осуществлению взаимосвязи с заказчиком** с целью выполнения его требований связанные с:

- информацией о продукции;
- обращением (прохождением) запросов и заказов, включая поправки;
- жалобами заказчика и действиям, относящимся к несоответствующей продукции;
- реакцией заказчика относительно функционирования продукции.

Предприятие должно **планировать и управлять проектированием и/или разработкой продукции**, подготавливать планы проектирования, которые включают:

- этапы процесса проектирования и/или разработки;
- требуемые действия по анализу, проверке и утверждению;
- полномочия и ответственность за действия по проектированию и/или разработке.

Внутреннее взаимодействие между различными группами, вовлеченными в проектирование и/или разработку, должно управляться таким образом, чтобы обеспечить эффективную взаимосвязь и четкую ответственность специалистов (см. таблицу 3.1).

Входные данные для проектирования и разработки должны включать:

- эксплуатационные требования заказчика или рынка;
- применяемые нормативные и законодательные требования;
- применяемые требования по охране окружающей среды;
- требования, вытекающие из прежних аналогичных проектов,
- любые другие требования, существенные для проектирования и разработки.

Эти входные данные должны быть проанализированы на адекватность и полноту; нечеткости или противоречивости требований должны быть устранены.

Выходные данные процесса проектирования и/или производства должны быть зарегистрированы в форме, дающей возможность проверки их по отношению к входным требованиям:

- соответствовать входным требованиям для проектирования и/или разработки;
- содержать или давать ссылку на критерий приемки продукции и/или услуги;
- определять характеристики продукции и/или услуги, которые являются существенными с точки зрения безопасности и правильного использования.

Документы, содержащие выходные данные проектирования и/или разработки, должны быть утверждены до их применения. На соответствующих этапах должен проводиться **систематический анализ проекта** для:

- оценки возможности выполнения требований к качеству;

- идентификации проблем, если таковые имеются, и выработки предложений по разработке решений.

В состав *участников анализа проекта* должны включаться представители служб, связанных с анализируемым этапом проектирования. Результаты анализа проекта и последующих действий должны быть зарегистрированы. Должна быть спланирована и выполнена проверка проекта и/или разработки, обеспечивающая уверенность в том, что выходные данные соответствуют входным требованиям. Результаты проверки и последующие действия должны быть зарегистрированы.

Утверждение проекта должно проводиться с целью подтверждения того, что конечная продукция способна отвечать особым требованиям для конкретных условий использования заказчиком. Когда это возможно, утверждение должно быть спланировано и выполнено до начала поставки или применения продукции. Там, где невозможно осуществить полное утверждение до начала поставки или применения, должно быть предпринято частичное утверждение выходных данных проекта в максимально возможном с точки зрения практики объеме. Результаты утверждения и последующих действий должны быть зарегистрированы.

Изменения проекта или модификация должны быть утверждены уполномоченным персоналом и зарегистрированы до их внедрения, следует определить влияние изменений на:

- взаимодействие между элементами проекта и/или разработки;
- взаимодействие между составными частями конечной продукции;
- имеющуюся продукцию и на функционирование ранее поставленной продукции;
- необходимость проведения повторной проверки или утверждения для всех или части выходных данных проектирования и/или разработки.

Предприятие должно *спланировать и управлять производственными и сервисными операциями*, включая те, которые предпринимаются после первоначальной поставки, посредством:

- предоставления технических условий, определяющих характеристики продукции, которые должны быть достигнуты;

- предоставления четко понимаемых производственных требований или инструкций для тех видов деятельности, где они необходимы для достижения соответствия продукции;
- применения и поддержания в технически исправном состоянии оборудования, необходимого для соответствующего производства, монтажа или проведения технического обслуживания;
- обеспечения, надлежащих производственных условий;
- предоставления и использования подходящего измерительного и контрольного оборудования;
- внедрения надлежащих действий по мониторингу или проверке;
- подходящих методов для выпуска, поставки и/или монтажа продукции.

Предприятие должно обеспечить **идентификацию статуса продукции** по отношению к требуемым измерениям и действиям по проверке и, где это целесообразно, должна идентифицировать продукцию подходящими способами в ходе всех процессов. Это должно применяться и к составным частям продукции, где их взаимодействие влияет на соответствие требованиям.

Предприятие должно определить и установить **процессы измерения функционирования системы менеджмента качества** (см. рис. 3.5). Удовлетворенность заказчика должна использоваться в качестве одного из измеряемых параметров результатов действия системы, как способ оценки текущего соответствия системы. Организация должна проводить мониторинг информации об удовлетворенности и/или неудовлетворенности заказчиков. Методы получения и использования такой информации и данных, а также измеряемые параметры должны быть определены. Результаты измерений должны использоваться для поддержания в рабочем состоянии и/или улучшения этих процессов. Доказательства проведения требуемых измерений и мониторинга и соответствия принятым критериям должны быть зарегистрированы. Данные отчеты должны указывать уполномоченных лиц, ответственных за выпуск продукции.

Предприятие должно **постоянно улучшать систему менеджмента качества**, установить общесистемную процедуру, которая определяет использование политики качества, целей, результатов внутреннего аудита, анализа данных, корректирующих и предупреждающих действий и анализа со стороны руководства в целях под-

держки постоянных улучшений. Установить процесс, направленный на сокращение или исключение причин несоответствий для предотвращения повторения несоответствий.

Общесистемная процедура для *процесса проведения корректирующих действий* должна определять требования по:

- идентификации несоответствий (включая жалобы заказчика);
- определению причин несоответствий;
- оценке необходимости действий для обеспечения уверенности в том, что несоответствия не повторятся;
- реализации всех действий, обусловленных необходимостью обеспечения уверенности в том, что несоответствия не повторятся;
- регистрации результатов предпринятых действий;
- анализу эффективности и регистрации предпринятых корректирующих действий.

Стандарт **ISO 9003:2004** – Руководство по применению стандарта **ISO 9001:2000** к программным средствам – предназначено для регламентирования менеджмента при приобретении, поставке, разработке, применении, сопровождении *сложных программных средств* и их обслуживании. Структура и разделы стандарта полностью соответствуют **ISO 9001:2000**, не содержат ограничений и изменений его требований, расширяет и конкретизирует их применительно к проектированию и производству сложных программных средств. Предлагается его использовать для установления и конкретизации процессов и систем менеджмента качества на соответствие требованиям к комплексам программ:

- как часть коммерческого контракта с другими организациями при производстве программных средств;
- при представлении полезного программного продукта для рынка;
- для использования при поддержке процессов организации проектов сложных ПС;
- для учета при встраивании ПС в комплексы аппаратуры;
- при организации сервиса эксплуатации программных продуктов.

В разделах 4 – 8 дополняются и комментируются аналогичные разделы **ISO 9001:2000** с учетом особенностей управления проектированием и производством в жизненном цикле сложных комплексов программ. Значительное внимание уделено конкретизации требова-

ний к системам менеджмента качества (гл. 4), обязанностям руководства проектами (гл. 5), управлению ресурсами (гл.6), процессам проектирования и производства (гл. 7), а также измерениям, анализу и усовершенствованию (гл. 8) сложных комплексов программ. Полное или частичное применение рекомендаций стандарта **ISO 90003** целесообразно в различных ситуациях, независимо от технологии, модели жизненного цикла, процессов разработки, последовательности действий и организационной структуры предприятия. Его рекомендуется применять как поддержку процессов программной инженерии, совместно со стандартами **ISO 12207, ISO 15504, ISO 9126, ISO 14598, ISO 15939**, с которыми коррелированы его концепции и ряд конкретных положений.

Лекция 4

СТАНДАРТЫ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНЫХ СРЕДСТВ ДЛЯ СЕРТИФИКАЦИИ СИСТЕМ КАЧЕСТВА ПРЕДПРИЯТИЙ

Подготовка стандартов жизненного цикла программных средств для производства

Во время формирования и адаптации стандартов жизненного цикла и производства конкретного проекта системы и программного средства следует выявить и определить замысел или потребность в новой или усовершенствованной системе технологических требований и процессов производства. Общие потребности производства формулируются с учетом таких факторов, как функции, стоимость, критичность и реализуемость программного продукта. Процессы заказа и конкретизации проекта должны быть использованы для установления технологических и/или эксплуатационных возможностей производственной системы [4, 9, 12].

Должны быть *определены предварительные характеристики программного средства* (рис. 4.1):

- потенциальное число и сложность компонентов программной конфигурации продукта;
- типы, размер и критичность программного средства;
- допустимый технический риск программного комплекса;
- типы, комплектность и носители подготовки документов;
- необходимость новой разработки, изменения или повторного применения компонентов прототипов программного средства;
- требования к основным характеристикам качества комплекса программ.

Следующие *факторы могут влиять* на заказ, производство, эксплуатацию или сопровождение в ЖЦ программного средства:

- различия в стратегиях и процедурах, принятых в разных организациях – соисполнителях;
- особенности стратегий различных проектов в цикле заказ –

поставка;

- размер и сложность проекта;
- требования к системе и программному продукту.

Подготовка стандартов жизненного цикла программных средств для производства:

- определение предварительных характеристик проекта программного средства;
- утверждение функциональных, технических, стратегических и экономических аспектов системы и программного продукта;
- выбор процессов производства и всего жизненного цикла в зависимости от характеристик конкретного проекта, используемых стандартов и технологии;
- определение конкретной среды, стандартов процессов производства и характеристик проекта программного средства;
- выбор политики заказчика и поставщика, связанной с безопасностью и управлением риском;
- анализ и адаптация соответствующих требований и процессов в жизненном цикле ПС для удовлетворения конкретным требованиям стандартов;
- сборка результатов адаптации стандартов для каждого конкретного вида практической деятельности специалистов;
- оценка изменений деятельности специалистов в соответствии с условиями договора на модификацию стандартов жизненного цикла программного средства.

Рис. 4.1

Требования к системе являются основой для дальнейшего ее разбиения по компонентам системы, таким как технические средства, компьютеры, программные средства и персонал пользователей. Процессы заказа, поставки и разработки могут быть использованы для анализа и определения требований к системе, принятия общих проектных решений по системе и предварительных требований для компонентов системы, включая программные средства. Процесс производства может быть использован для анализа, демонстрации, аттестации, тестирования и макетирования соответствующих требований и пилотных решений с использованием стандартов.

Проектирование и производство охватывают этапы создания, интеграции (сборки), тестирования и оценки системных технических средств, компьютеров, программных средств, оборудования, персонала подсистем, его обучения и компонентов сопровождения. Выходными результатами этой работы, являются макет системы, достаточно близкий к заданному, документы, необходимые для последующей проверки на соответствие требованиям и стандартам. Данная работа может включать однократное или многократное использование процесса разработки, скоординированное с другими компонентами и процессами системы. Результатами являются исходные **требования к программному средству**, его системный проект и соответствующие программные компоненты.

Выбор процессов производства и всего жизненного цикла зависит от характеристик конкретного проекта, используемых стандартов и состояния технологии. Работы и задачи процессов жизненного цикла отвечают на вопросы – **что делать**, а не на вопросы – **как делать**. Такое свойство процессов ЖЦ ПС предоставляет заказчику широкие возможности для установления **требований к производству конечного продукта** и, в то же время, позволяет разработчику создавать и применять соответствующие методы, способы и инструментарий для его производства.

Адаптация и определение конкретной среды, стандартов процессов разработки и характеристик проекта программного средства, должны учитывать:

- процессы, стратегии и процедуры, которые уже применяются на предприятии;
- является ли изменяемый процесс важным для достижения целей предприятия и конкретного проекта;
- учтен ли повышенный деловой риск системы и комплекса программ при модификациях процессов жизненного цикла;
- какая модель и профиль стандартов процессов жизненного цикла системы и/или комплекса программ уже используется;
- является ли система критичной по безопасности и в какой степени;
- будут ли использованы новые технологии и инструментарий производства.

Какой бы не была причина для применения процессов и стандартов ЖЦ ПС, **стратегия их внедрения** состоит из типовых шагов:

- создание плана внедрения процессов и профиля стандартов жизненного цикла;
- адаптация стандартов ЖЦ к конкретному проекту;
- формализация метода внедрения процессов и профиля стандартов жизненного цикла;
- утверждение метода внедрения и применения стандартов ЖЦ ПС.

Должны быть определены соответствующие **политики и процедуры** заказчика и поставщика, с которыми следует согласовать особенности программного проекта, связанные с безопасностью и управлением риском. Необходимо определить соответствующие законы и подзаконные акты, включая документы, относящиеся к внешней среде, общей безопасности и конфиденциальности системы, влияющие на программный продукт. Такие политики и процедуры следует учитывать при разработке, эксплуатации и сопровождении программного средства. Если имеются готовые политики безопасности, в них необходимо включить работы по анализу требований из процесса производства и эксплуатации системы.

На соответствующем уровне детализации целесообразно установить **подсистемы и компоненты конфигурации** системы, а также ее характеристики, особенно те, которые относятся к программному средству. При определении этих характеристик следует отметить, какие из них являются критичными при эксплуатации системы. Характеристики системного уровня, относящиеся к программному средству и подлежащие учету, включают:

- межсистемные и внутрисистемные интерфейсы;
- интерфейсы пользователя;
- оценки влияния ошибок программного средства на безопасность системы;
- оценки предполагаемых вычислительных мощностей, производительности и временных ограничений;
- наличие программ, реализованных техническими средствами.

Для систем реального времени, работа которых в значительной степени зависит от правильного функционирования программных продуктов и своевременной выдачи результатов, необходим наиболее тщательный надзор и контроль производства. При разработке программного средства может иметь место технический риск. Если к программному средству предъявлены требования по безопасности,

защите или другие критические требования, то особенно необходимо применять стандарты, строгое определение технических требований, особенно тщательное проектирование, тестирование и оценка качества, а также независимая верификация и аттестация.

Анализ и адаптация соответствующих требований и процессов в жизненном цикле ПС состоит в выборе метода удовлетворения конкретным **требованиям стандартов**, программными решениями или изменениям существующей практической деятельности в предприятии. В этом случае под практической деятельностью в жизненном цикле ПС понимается способ решения задачи конкретным лицом при выполнении им повседневной работы в соответствии с положениями стандартов. При адаптации процессов жизненного цикла для конкретного проекта в процесс его разработки должны быть внесены следующие **дополнительные работы**:

- по анализу модификации требований стандартов;
- по проектированию и документированию комплекса программ;
- по сборке комплекса программ;
- по оценке достигнутых характеристик качества программного продукта.

Процессы по анализу модификации требований стандартов состоят из следующих задач, которые разработчик должен выполнить или обеспечить их выполнение в соответствии с условиями конкретного договора при подготовке жизненного цикла комплекса программ:

- существующая практическая деятельность должна быть проанализирована с целью определить наиболее эффективные методы реализации изменений системы;
- анализ должен учитывать обзор деловой деятельности, структуру предприятия, а также полномочия и обязанности каждого подразделения, вовлеченного в проект, и документально оформленные;
- практическая деятельность должна быть оценена с учетом перечисленных критериев, а результаты этих оценок должны быть документально оформлены.

Проектирование и документирование процессов ЖЦ ПС включают задачи, при решении которых разработчику следует:

- определить, в соответствии с интересами участников процесса, информационные потоки процесса, связанные с практической дея-

тельностью;

- создать состав и предварительные версии технологических документов;
- подготовить методические материалы, используемые при реализации договора, для обучения специалистов соответствующей практической деятельности;
- необходимо выполнять регулярные обследования степени удовлетворения запросов пользователей и анализировать их результаты в целях определения степени пригодности производственной системы.

Базовые стандарты жизненного цикла программных средств

Стандарт **ISO 12207** является базовым, на основе которого формируется и развивается *профиль стандартов обеспечения проектирования, производства, сопровождения и управления конфигурацией сложных программных средств* – рис. 4.2. Компоненты стандарта **ISO 12207** и архитектура некоторых его разделов непосредственно использованы при детализации структуры и содержания стандартов: управления проектами ПС – **ISO 16326**; сопровождения комплексов программ – **ISO 14764**; управления конфигурацией ПС – **ISO 15846**. В ряде стандартов приводятся перекрестные ссылки, и рекомендуется использовать отдельные фундаментальные положения стандарта **ISO 12207**.

Стандарт **ISO 12207:1995** – Процессы жизненного цикла программных средств – наиболее полно на уровне международных стандартов отражает *жизненный цикл, технологию разработки и процессы обеспечения качества сложных программных средств*. Жизненный цикл ПС представлен набором этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующих ведение разработки на всех стадиях от подготовки технического задания до завершения испытаний версий и окончания эксплуатации программного продукта. В ЖЦ включаются описания исходной информации, способов выполнения операций и работ, устанавливаются требования к результатам и правилам их контроля, а также к содержанию технологических и эксплуатационных документов. Определяется организационная структура коллективов, распре-

деление и планирование работ, а также контроль за реализацией ЖЦ ПС.



Рис. 4.2

Стандарт **ISO 12207** определяет *набор процессов*, используемых для больших и сложных программных проектов, может быть адаптирован и применен к программному проекту любого типа, меньшего размера и сложности. Процессы, работы и задачи в стандарте описаны в наиболее общей естественной позиционной последовательности, аналогичной каскадной модели ЖЦ ПС. Стандарт может использоваться как непосредственный директивный, руководящий или как рекомендательный документ, а также как организационная база при

создании средств автоматизации соответствующих технологических этапов или процессов производства программных продуктов. Для реализации положений стандарта должны быть выбраны инструментальные средства, совместно образующие взаимосвязанный комплекс технологической поддержки и автоматизации ЖЦ и не противоречащие предварительно скомпонованному набору нормативных документов.

Стандарт определяет архитектуру, процессы, разделы и подразделы ЖЦ ПС, а также перечень базовых работ и детализирует содержание каждой из них. Архитектура жизненного цикла программных средств в стандарте базируется на **трех крупных компонентах** (рис. 4.3):

- основные процессы жизненного цикла ПС и определяющие работы (раздел 5);
- вспомогательные процессы и работы, поддерживающие жизненный цикл ПС (раздел 6);
- организационные процессы и управление жизненным циклом ПС (раздел 7).

Разделы 5, 6, 7 стандарта состоят из ряда подразделов, в которых подробно раскрывается содержание каждой работы и комментируются особенности их выполнения. Рекомендации к каждому подразделу состоят в среднем из 3–6 пунктов – работ (действий). Общее число работ и комментариев к ним в стандарте свыше 220.

В **разделе 5** изложены основы ЖЦ и рекомендации по подготовке, разработке, эксплуатации и сопровождению программных средств. Процессы приобретения и/или подготовки к созданию ПС должны начинаться с инициализации проекта, анализа концепции, анализа рынка продуктов, выработки требований и состава поддерживающих документов, создания предварительного плана действий проекта. Следует оценить отклики фирм на предложения по созданию проекта, заключить контракт, спланировать жизненный цикл, организовать поддержку разработки отчетами, а также сдачи и завершения проекта.

Основные работы по созданию сложного комплекса программ рекомендуется начинать с определения состава сопровождающих документов, выбора средств конфигурационного управления и обеспечения качества, а также выбора методов и средств технологического обеспечения разработки всей информационной системы.

**Стандарт ISO 12207:1995 –
Процессы жизненного цикла программных средств**

**5. Основные процессы
жизненного цикла**

- 5.1. Приобретение – заказ
- 5.2. Поставка
- 5.3. Разработка –
производство
- 5.4. Эксплуатация
- 5.5. Сопровождение

**7. Организационные процессы
жизненного цикла**

- 7.1. Управление проектом
- 7.2. Создание инфраструктуры
- 7.3. Усовершенствование
- 7.4. Обучение

**6. Вспомогательные процессы
жизненного цикла**

- 6.1. Документирование
- 6.2. Конфигурационное
управление
- 6.3. Обеспечение качества
- 6.4. Верификация
- 6.5. Контроль соответствия
– аттестация
- 6.6. Совместный анализ
- 6.7. Аудит
- 6.8. Решение проблем –
устранение дефектов

Рис. 4.3

Следует проанализировать и формализовать системные требования и критерии качества системы: функциональные, пользовательские, защищенности, интерфейсов с внешней средой, сопровождаемости. На этой базе проектируется архитектура всей системы, выделяются и анализируются требования к программным средствам. Рекомендуется при определении и формировании характеристик качества ПС руководствоваться стандартом **ISO 9126** и предложенной в нем номенклатурой показателей (см. лекцию 7). Кодирование и тестирование каждого компонента ПС должно быть оформлено совокупностью документов, удостоверяющих соответствие компонента первичной спецификации, содержащих тесты и результаты тестирования.

Рекомендуется разрабатывать план работ, включающий комплексирование компонентов, тестирование по всем разделам требований и показателям качества, а также документирование плана, результатов интеграции, использованных тестов, критериев оценки и полученных результатов. Далее ПС следует подвергать квалификационному (аттестационному) тестированию по всем разделам требований контракта, при широком варьировании тестов, изменениях значений критериев, а также тестировать полноту и адекватность технологической и пользовательской документации реальному программному продукту. Проверенный таким образом комплекс программ интегрируется в вычислительные средства информационной системы, средства визуализации и телекоммуникации. После объединения всех средств система должна подвергаться квалификационному тестированию и испытаниям на всю совокупность требований к системе, а также производится оформление и проверка полного комплекта документации.

Подготовленный специалист должен освоить все процедуры поддержки эксплуатации и применения, а также тестирования ПС при функционировании, на соответствие критериям оперативного использования документации. Работы по поддержке пользователей состоят в помощи и консультации их при обнаружении дефектов или ошибок при применении ПС в составе информационной системы. Эти работы взаимодействуют с работами, обеспечивающими сопровождение ПС. Специалисты анализируют сообщения об ошибках и предложения на модификацию ПС, селективируют их на соответствие требованиям контракта и оценивают целесообразность проведения изменений. Подготовленные изменения тестируются и проверяются по критериям, определенным в документации. Далее планируется распространение проведенных изменений или новой версии пользователям. Некоторые версии с определенной совокупностью изменений должны планироваться для ликвидации и прекращения сопровождения. Вспомогательные технологические работы, поддерживающие жизненный цикл ПС, и рекомендации по их выполнению изложены в **разделе 6**. Процессы документирования ПС должны охватывать планирование и обеспечение документирования, рекомендации по стандартизации, проектированию и разработке, а также по производству, конфигурационному управлению и сопровождению комплекта документации на ПС. Конфигурационное управление предлагается вклю-

чать в общий план управления проектом с процедурами конфигурационной идентификации, контроля, учета, отчетности и развития конфигурации. Для обеспечения гарантий качества следует использовать планирование, методологию, процедуры и стандарты поддержки качества ПС в соответствии с контрактом с учетом доступных ресурсов. Рекомендуются планирование и выполнение работ в процессе всего жизненного цикла ПС реализовать на основе требований стандарта **ISO 9001** (см. лекцию 3). Верификация ПС должна включать её организацию, планирование и техническое обеспечение. Представлена структура раздела контракта на верификацию, содержание процесса, состав требований, проектирование процесса тестирования, обобщение и документирование результатов. Валидация – удостоверение правильности (аттестация) должна гарантировать полное соответствие программного продукта спецификациям, требованиям и документации на ПС и возможность его надежного функционирования и безопасного применения пользователем. Рекомендуются ее выполнять путем тестирования и испытаний во всех возможных ситуациях исходных данных и проводить независимыми специалистами.

Управление проектом должно быть сосредоточено, в основном, в подготовке и обеспечении планирования и управления ресурсами, персоналом, аппаратурой, программными средствами и инструментарием. Процессы ревизии – аудита служат для установления соответствия реальных работ и отчетов требованиям, планам и контракту. Отмечается, что ревизоры не должны иметь прямой зависимости от разработчиков ПС, они определяют состояние работ, использование ресурсов, соответствие документации спецификациям и стандартам, корректность тестирования. В процессе решения задач должны выявляться и регистрировать проблемы и дефекты последующего применения программных средств и их функционирования. Каждый дефект или ошибка должны быть определены, идентифицированы, описаны, проанализированы и устранены в процессе сопровождения в соответствии с контрактом.

Организации жизненного цикла ПС посвящен *раздел 7*. Она включает основные работы по управлению проектом, производством и средствами для обеспечения процессов по разработке, эксплуатации и сопровождению. Процессы формирования инфраструктуры должны состоять из выбора и установления аппаратных и программных средств, технологии, стандартов и обслуживания, используемых для

разработки, сопровождения и обеспечения эксплуатации ПС. Инфраструктура должна модифицироваться и сопровождаться в соответствии с изменениями требований к разработке и подлежит конфигурационному управлению. Процессы совершенствования жизненного цикла ПС состоят в установлении, оценивании, измерении, контроле и корректировке процессов жизненного цикла конкретных ПС. Совершенствование ЖЦ ПС должно учитывать требования пользователей и развитие определенной технологии. Процессы обучения определяются требованиями к проекту, должны учитывать необходимые ресурсы, управление и технические средства. Должны быть разработаны и представлены пользователю материалы, облегчающие обучение эксплуатации ПС.

Изложены рекомендации по преобразованию и адаптации базовой структуры этого международного стандарта для конкретного проекта (приложение А) и руководство по их выполнению в ЖЦ ПС (приложение В).

Стандарт **ISO 12207:AMENDMENT 1:2002** – Изменения (улучшения, редакция) стандарта **ISO 12207** – изложены на базе этого стандарта и набора Приложений новых процессов и рекомендаций. Все Приложения базируются на обширной таблице **E1**, в которой представлены рекомендуемые источники процессов жизненного цикла ПС, их наименование и тип, детализирующиеся в последующих Приложениях **F**; **G**; **H** этого документа. Новые, расширенные комментарии и рекомендации к каждому процессу жизненного цикла сложных ПС структурированы и изложены в составе базовых подразделов: *цель* и *результаты*, что конкретизирует и существенно улучшает возможность и удобство применения пользователями процессов стандарта **ISO 12207**. В Приложении **F** подробно прокомментированы существующие в стандарте **ISO 12207** процессы: базовые (5 раздел), вспомогательные (6 раздел) и организационные (7 раздел). В Приложении **G** детально изложены рекомендуемые дополнительные *новые* процессы, отсутствовавшие в этих стандартах: обучение и управление специалистами, качеством и повторным использованием программных компонентов, оценка и организация человеческих ресурсов, технологическое обеспечение проектов ПС. Приложение **H** содержит подробные рекомендации по организации и реализации комплексирования и сборки крупномасштабных комплексов программ.

Стандарт **ISO 12207:AMENDMENT 2:2004** – Изменения (улучшения, редакция) стандарта **ISO 12207** – содержит редакционные изменения и дополнения описаний некоторых процессов, представленных в **AMENDMENT 1**. Эти корректировки в основном относятся к процессам управления: требованиями, изменениями и конфигурацией ПС, повторным применением программных компонентов и инфраструктурой процессов.

В стандарте **ISO 15504:1-5:1998 – 2004** – Оценка (аттестация) процессов жизненного цикла программных средств: Ч.1. Основные понятия и вводное руководство. Ч.2. Эталонная модель процессов и их зрелости. Ч.3. Проведение аттестации. Ч.4. Руководство по проведению аттестации. Ч.5. Модель аттестации и руководство по показателям. На базе концепции методологии **СММІ** (см. лекцию 3) подробно представлены *оценивание и совершенствование процессов жизненного цикла программных средств*. Он предназначен для предприятий, занимающихся планированием, управлением, контролем, а также совершенствованием приобретения, поставки, разработки, эксплуатации, сопровождения и оценивания качества крупномасштабных программных средств. Стандарт расширяет, детализирует и предоставляет базу для аттестации, поддержки и реализации на предприятии процессов жизненного цикла ПС, регламентированных стандартом **ISO 12207**. Рубрикации процессов в этих двух стандартах подобны. В стандарте **ISO 15504** модернизирован и расширен состав организационных процессов и подробно детализированы работы во всех стандартизируемых процессах жизненного цикла ПС. Поэтому оба стандарта целесообразно *применять совместно* при конкретизации процессов жизненного цикла реальных комплексов программ. Стандарт предоставляет структурный подход к *аттестации качества процессов ЖЦ ПС*, проводящейся предприятием с целью:

- выяснения состояния и качества его собственных процессов для их усовершенствования;
- определения пригодности процессов для выполнения определенного требования или класса требований заказчика к качеству ПС;
- определения качества и пригодности процессов конкретного предприятия для определенного договора или класса договоров на ЖЦ ПС.

Аттестация реализации ЖЦ ПС направлена на обеспечение адекватности управления процессами и должна учитывать внешнюю

среду, в которой выполняются аттестуемые процессы. Чтобы предприятие могло улучшить качество своей продукции, оно должно иметь проверенный, последовательный и надежный метод для аттестации состояния своих процессов, а также иметь средства использования ее результатов как часть Программы усовершенствования процессов ЖЦ ПС. Использование аттестации процессов внутри предприятия должно способствовать выработке культуры постоянного совершенствования и повышения характеристик качества ПС, а также соответствующих механизмов поддержания этой культуры и оптимизации использования ресурсов. Это должно приводить к появлению **зрелых предприятий**, максимально восприимчивых к возрастающим требованиям потребителя и рынка, имеющих минимальную стоимость полного жизненного цикла своей продукции и, как результат, максимально удовлетворяющих требования конечного пользователя к характеристикам качества ПС.

Покупателям и заказчикам ПС выгодно использование аттестации процессов ЖЦ при определении зрелости поставщика, что:

- уменьшает неопределенность при выборе поставщиков программных комплексов за счет того, что риски, связанные с реальной зрелостью подрядчика, выявляются еще до заключения договора;
- позволяет заранее предусмотреть необходимые меры на случай возникновения рисков события;
- предоставляет количественные критерии выбора при сопоставлении потребностей бизнеса, требований и оценочной стоимости проекта с реальной зрелостью конкурирующих поставщиков;
- приводит к общему пониманию необходимости использования результатов аттестации для усовершенствования процессов и оценки зрелости поставщика при прогнозировании характеристик ЖЦ ПС.

Анализ результатов в свете потребностей предприятия способствует выявлению сильных и слабых сторон используемых процессов, а также присущих им рисков. Это, в свою очередь, помогает определить, эффективны ли эти процессы для достижения заданных целей проекта ПС, а также выявить существенные причины недостаточного качества продукции, риск превышения бюджет или сроков. Все вместе позволяет расставить приоритеты при совершенствовании процессов. Определение зрелости связано также с анализом и выявлением рисков, связанных с выполнением проекта, использующего

выбранные процессы. Стандарт **ISO 15504** связан с другими международными стандартами, он дополняет некоторые стандарты и другие модели для оценки зрелости, качества и эффективности предприятий и процессов ЖЦ ПС.

Руководства по применению базовых стандартов жизненного цикла программных средств

Стандартом **ISO 15271:1998** – Руководство по применению ISO 12207 – поддержано *практическое использование стандарта ISO 12207*. Он содержит подробные рекомендации по внедрению, применению в проектах ПС, а также при организации работ и реализации требований стандарта **ISO 12207**. Изложены рекомендации по формализации методов внедрения, по подготовке и утверждению плана работ, по адаптации требований стандарта к конкретным особенностям характеристик ЖЦ и внешней среды проекта ПС и по сопровождению пилотного проекта.

Перечислены и прокомментированы особенности и факторы, которые следует учитывать при применении стандарта в различных проектах, связанные: с используемой моделью жизненного цикла системы и проекта ПС; со стратегией и процедурами функционирования заинтересованных организаций заказчика и поставщика; с характеристиками системы и типом комплекса программ; с особенностями его сопровождения. Кроме того, рекомендуется принимать во внимание объем комплекса программ проекта и количество вовлеченных специалистов, критичность области применения ПС и возможные технические риски. Обращается внимание на возможные особенности применения стандарта в различных типах предприятий: на ситуации и причины, которые следует рассматривать при внедрении стандарта, на степень необходимой жесткости административного управления проектом при его использовании. Иллюстрируется общая модель жизненного цикла системы и особенности в ней части модели, отражающей ЖЦ ПС. Описаны рекомендации по проведению общесистемных работ для ПС в целом: по исследованию и определению концепции; по проектированию и разработке; по созданию и производству; по распространению и продаже; по сопровождению и поддержке проекта системы.

Стандартом **ISO 16326:1999** – Руководство по применению **ISO 12207** при административном управлении проектами – регламентиро-

ваны **процессы управления проектированием**. Основным содержанием в **ISO 16326** является раздел, в котором цитируется каждое требование п. 7.1 из стандарта **ISO 12207** и приводятся подробные рекомендации по их реализации. Детально изложены работы по планированию и процедуры выполнения процесса административного управления на различных этапах жизненного цикла ПС. Кроме того, представлены таблицы соответствия требований **ISO 12207** детальным рекомендациям стандарта **ISO 10006:1997** по административному управлению сложными техническими проектами.

Подготовка и определение области управления должны начинаться с установления требований к реализуемому процессу. Администраторы отвечают за управление продуктом, проектом, работами и задачами процессов, таких как заказ, поставка, разработка, эксплуатация, сопровождение или вспомогательные процессы. Затем администратор должен определить возможности реализации процесса, проверяя наличие, соответствие и достаточность ресурсов, выделенных для выполнения и управления процессом (специалистов, материалов, технологии и условий), а также реальность сроков реализации процесса. При планировании администратор должен подготовить планы для:

- выполнения процессов, которые должны содержать описания соответствующих работ и задач, а также обозначения создаваемых программных компонентов и продуктов;
- установление графиков своевременного решения задач;
- оценки трудозатрат и ресурсов;
- распределение задач по исполнителям и их обязанностей;
- установление используемых в процессе критериев управления качеством;
- обеспечение условий и инфраструктуры выполнения процесса.

Администратор должен осуществлять текущий контроль и надзор за выполнением процесса, подготавливая как внутренние отчеты о развитии процесса, так и внешние отчеты для заказчика в соответствии с условиями договора. Он должен исследовать, анализировать и устранять дефекты, обнаруженные при выполнении процессов. Все обнаруженные дефекты и результаты их устранения должны быть документально оформлены, а также в установленные сроки подтвер-

ждены полная реализация процесса и выполнение утвержденных планов.

Администратор должен обеспечить проверку и оценку качества программных продуктов и планов на адекватность требованиям, а также проверять результаты оценок продуктов, работ и задач, реализуемых в ходе процесса, на соответствие целям и утвержденным планам. После создания всех запланированных программных продуктов администратор должен определить степень их соответствия критериям, установленным в договоре или организационной процедуре. Все представленные результаты и документация должны быть сохранены в архиве.

Многие современные модели жизненного цикла ориентированы преимущественно на сложные, критические ПС обработки информации и управления в реальном времени. К таким ПС предъявляются наиболее высокие требования к качеству функционирования, они создаются и развиваются большими коллективами специалистов в течение длительного времени. В этих *стандартах ЖЦ рекомендуется* руководителям программных проектов постоянно влиять на ход работ, санкционируя определенные работы или приостанавливая их, если они могут повлиять на другие области деятельности, *определять методологию и технологию реализации проекта*, необходимые для:

- прогнозирования и соответствующего предотвращения или минимизации неблагоприятного воздействия потенциальных проблем проекта;
- принятия временных и/или постоянных решений в жизненном цикле ПС;
- решения возникающих проблем и устранения дефектов проекта ПС;
- принятия ответственности за проект в целом, его процессы, работы, ресурсы, продукты и результаты.

Процессы управления проектом, работы и задачи необходимо выполнять неоднократно – итерационно, чтобы реализовать требования и цели проекта. Основываясь на выбранной модели жизненного цикла программного средства, некоторые процессы, работы и задачи можно выполнять одновременно, при этом они могут быть взаимосвязаны или скоординированы в зависимости от этапов проекта. Руководитель должен отвечать за управление проектом, работами и за-

дачами соответствующих процессов, такими как заказ, поставка, разработка, эксплуатация, сопровождение или вспомогательные процессы.

Стандарты сопровождения и управления конфигурацией программных средств

Стандарт **ISO 12207** – является основным международным нормативным документом, регламентирующим сопровождение и изменения программных средств. Этим задачам в стандарте **ISO 12207** посвящены разделы: **5.5. Процесс сопровождения** и **6.8. Процесс решения проблем** – устранения дефектов в составе раздела Вспомогательных процессов. При реализации этих задач целесообразно применять практически всю совокупность профиля стандартов жизненного цикла ПС, которые, по существу, являются развитием и детализацией этого основного стандарта. Непосредственно сопровождение регламентирует стандарт **ISO 14764**.

В соответствии с требованиями заказчика по развитию и модификации программного продукта в жизненном цикле *должен быть организован процесс его сопровождения* (см. п. 5.5 стандарта **ISO 12207**). После активизации процесса следует разработать план сопровождения и соответствующие процедуры, а также выделить конкретные ресурсы для сопровождения. После поставки заказчику программного продукта сопроводитель, в соответствии с договором и предложением о модификации или отчетом о дефекте, должен изменить соответствующие программы, данные и документы. Рекомендуется проводить регулярный контроль с целью проверки корректности выходных результатов конкретных работ по сопровождению. Работы, обеспечивающие сопровождение ПС, представленные в *разделе 5.5 стандарта ISO 12207*, включают:

- подготовку процесса;
- анализ проблем и изменений;
- внесение изменений;
- проверку и приемку при сопровождении;
- перенос на другие платформы;
- снятие с эксплуатации и ликвидацию продукта.

В *стандарте ISO 14764:1999* – Сопровождение программных средств – каждый перечисленный выше процесс, детализирован, в виде подробных комментариев и рекомендаций. *Целью планирова-*

ния сопровождения является подготовка плана работ по сопровождению и обеспечение ресурсов, необходимых для проведения этих работ после передачи программного продукта на сопровождение. Планирование начинается после определения концепции сопровождения ПС и завершается разработкой плана сопровождения, используемого в качестве основы при сопровождении. Этот план должен быть подготовлен сопроводителем во время первичной разработки базовой версии ПС и включать в себя процессы анализа предложений пользователя по внесению изменений в программный продукт. **Общий план сопровождения должен определять:**

- причины необходимости сопровождения;
- состав исполнителей работ по сопровождению;
- роли и обязанности каждого субъекта, вовлеченного в сопровождение;
- как должны быть выполнены основные процессы и работы;
- какие имеются и необходимы ресурсы для сопровождения;
- методы и средства организации работ по управлению, выпуску продукта и синхронизации работ;
- перечень всех проектных результатов и продуктов, подлежащих поставке заказчику;
- критерии завершения соответствующей деятельности, работ и задач;
- состав отчетных материалов по этапам, затратам и графикам проведения работ;
- периодичность и способы выдачи отчетных материалов;
- состав отчетных материалов по проблемам и устраненным дефектам;
- место проведения сопровождения;
- время начала и длительность сопровождения.

Стандартом **ISO 14764** рекомендуется сопроводителям формализовать **конкретный план сопровождения ПС** из общего состава процессов жизненного цикла, который уточнить и адаптировать с учетом объема и особенностей проекта в составе разделов:

- описание сопровождаемой системы, в которую входит ПС;
- концепция сопровождения комплекса программ; описание уровня сопровождения системы и ПС; установление длительности процессов сопровождения; адаптация стандартизированных процессов сопровождения;

- организационные работы по сопровождению, роли и обязанности специалистов;
- ресурсы: состав специалистов; инструментальные средства; технические средства; документы и планы;
- процессы – как должна быть выполнена конкретная деятельность;
- определение уровня обучения, необходимого для сопровождаителей и для пользователей;
- протоколы и отчеты по сопровождению; контрольные данные, собранные при работах по сопровождению.

При **подготовке процесса** персонал сопровождения должен разработать, документировать и выполнить планы и процедуры для проведения действий и задач сопровождения. Персонал сопровождения должен устанавливать процедуры для получения, записи и сообщений о дефектах, проблемах и запросах на изменения от пользователей и для обеспечения обратной связи к пользователям. Для обеспечения создания эффективных планов и процедур сопровождения сопровождаитель должен:

- выполнить оценку сопровождаемой системы;
- гарантировать официальное подтверждение принятия на себя обязанностей сопровождаителя программного продукта;
- провести анализ доступных ресурсов для сопровождения;
- оценить и согласовать с заказчиком финансирование и стоимость сопровождения;
- установить требования к процессу передачи программного продукта сопровождаителю;
- определить подлежащие реализации процессы сопровождения;
- документально оформить процесс сопровождения в виде планов и процедур, согласованных с заказчиком.

Стратегии сопровождения должны быть ориентированы на людские и материальные ресурсы, необходимые и доступные для обеспечения развития и модификаций программного продукта. Этот анализ является исходным при разработке стратегии сопровождения. Процесс разработки изменений включает в себя ряд работ, связанных с планированием сопровождения ПС. Эти виды деятельности должны быть определены в стратегии сопровождения программного продукта.

Анализ проблем и модификаций включает оценку влияния на организацию, существующую систему и интерфейсы системы для: корректировок, усовершенствований, профилактики или адаптации ПС к новой среде; определения размера модификации, стоимости, времени на модификацию; риска влияния на производительность, надежность или безопасность. Персонал сопровождения должен дублировать или верифицировать проблему – дефект. Базируясь на анализе, персонал обслуживания должен разработать варианты выполнения модификаций. Сопровождающий должен документально оформить проблему, модификационный запрос, результаты их анализа и варианты реализации изменений. Следует получать утверждение выбранного варианта модификации согласно договору (см. таблицу 3.1).

Основой для проведения работы по анализу дефектов и изменений являются: официальное предложение о модификации или отчет о дефекте, системные и/или проектные документы и нормативные документы. Исходные данные для проведения работы включают: предложения о модификациях или отчеты о дефектах; последнюю базовую версию программного продукта; информационный архив проекта программного средства; информацию о состояниях конфигурации ПС, БД и системы; функциональные требования; требования к интерфейсам.

Реализация модификации содержит анализ и определение, какие документы, компоненты программного средства и версии требуют изменений, что должно быть документировано. Требования Процесса Разработки (ISO 12207 п. 5.3) должны быть дополнены следующим образом:

- должны быть определены и документированы испытания и критерии оценки для тестирования и оценивания качества модифицированных и неизмененных компонентов программного средства и единиц конфигурации системы;
- должна быть гарантирована полная и правильная реализация новых и модифицированных требований, гарантировано, что первоначальные неизменные требования были не искажены и результаты испытаний документированы.

Для того чтобы подтвердить актуальность представленных отчетов о дефектах, сопроводитель должен **продублировать и верифицировать** возникшие проблемы – дефекты, выполнив следующие этапы решения данной задачи:

- разработать стратегию верификации и квалификационного тестирования для проверки устранения конкретной проблемы – дефекта;
- реализовать управление конфигурацией представленной версией программного продукта;
- ввести в действие (инсталлировать) представленную версию программного продукта;
- провести тестирование для проверки проблемы – дефекта, предпочтительно с использованием копий представленных тестовых данных;
- документально оформить результаты квалификационного тестирования.

Если конкретная проблема не может быть повторена сопровождателем, он должен проверить правила, политики и документы ЖЦ ПС на предприятии. На основе проведенного анализа сопровождатель должен разработать **варианты реализации изменения**:

- присвоить соответствующий приоритет проблеме (дефекту) или предложению о модификации;
- установить наличие возможностей и средств для решения проблемы;
- оценить масштаб и трудоемкость данной модификации;
- разработать варианты реализации конкретного изменения;
- определить влияние данных вариантов на функциональную пригодность и технические средства системы;
- выполнить анализы риска для каждого варианта.

До внесения изменений в систему и программный продукт в соответствии с договором с заказчиком сопровождатель должен **согласовать выбранный вариант корректировки**, выполнив следующее:

- представить результаты анализов на согласование в соответствующие группы Совета по управлению конфигурацией;
- участвовать в обсуждениях с заказчиком рассматриваемой корректировки;
- обновить после согласования, состояние, содержание и статус предложения о модификации;
- обновить, после согласования, конкретные требования к программному продукту, если соответствующая заявка на корректировку носит характер модернизации и совершенствования объекта.

Проектирование архитектуры модификаций в плане трансформируется в его общую структуру и определяет функции и компоненты модифицированного программного средства. Основными особенностями данной работы среди процессов ЖЦ ПС, влияющими на сопровождаемость, являются выбор структуры программы, разбиение ее на компоненты (модули) и поток данных, циркулирующий между ними. При модификациях важно использовать знания коллектива специалистов по обработке данных, относящиеся к возможности использования частей существующих программ или библиотек, доказавших высокое функциональное качество.

Сопроводителю следует определить процедуры для: получения, документирования и контроля отчетов о дефектах и предложений о модификациях от пользователей; обеспечения обратной связи с пользователями. Каждая, возникающая проблема и/или дефект, должны быть документально оформлены и введены в процесс анализа изменений, для чего следует:

- разработать схему классификации и присвоения приоритетов для предложенных модификаций и описаний дефектов;
- разработать процедуры проведения целевых анализов изменений;
- определить процедуры представления предложенных модификаций и описаний дефектов оператором;
- определить организацию обратной связи с пользователями при анализе изменений;
- определить, как пользователей будут обслуживать в период реализации сопровождения;
- определить, как будут введены предлагаемые модификации в базу данных учета состояний изменений и используемых ресурсов.

Контроль за рассматриваемыми работами следует проводить посредством процесса *совместного анализа* (см. стандарт **ISO 14764**). В конце работ должен быть проведен *анализ риска*. На основании выходных результатов анализа может быть пересмотрена предварительная оценка требуемых ресурсов и с привлечением пользователей или заказчика принято решение о целесообразности перехода к работе по внесению изменений в базовую версию программного продукта. Результатами этой работы являются: анализ влияния изменений;

рекомендуемый вариант и согласованные изменения; обновленные и исправленные документы.

При внесении изменений в ПС сопроводитель разрабатывает и тестирует конкретные изменения программного продукта. Исходными данными для проведения работы при внесении изменений должны быть: базовая версия программного продукта; согласованные с заказчиком предложения о модификации; согласованные документы на реализацию корректировки; отчет о влиянии корректировки и выходные результаты работы по анализу изменений. Сопроводитель должен выполнять анализ **использования процессов разработки комплекса программ** при внесении изменений (см. п. 5.3 и 5.5 стандарта **ISO 12207**). После согласования корректировки сопроводителю следует провести анализ и определить, какие документы, программные модули и их версии требуют изменения.

Результаты испытаний корректировок должны быть документально оформлены. Контроль за рассматриваемой работой должен быть проведен посредством процесса совместного анализа. Результатами данной работы являются: обновленные планы, документы и процедуры тестирования; измененные исходные программы; отчеты о квалификационном тестировании; показатели, характеризующие качество внесенных изменений. Обновленные документы должны включать подробный отчет о проведенном анализе; обновленные требования; обновленные планы, процедуры и отчеты о тестировании; обновленные учебные материалы. Сопроводитель должен провести проверки каждого внесенного изменения совместно с заказчиком, утвердившим изменение в целях подтверждения целостности и работоспособности измененной системы:

- отслеживание реализованных предложений о модификации и отчетов о дефектах относительно требований предыдущей базовой версии проекта и программных кодов;
- проверку соблюдения стандартов на ЖЦ ПС и системы;
- проверку того, что изменены только нужные компоненты программного средства;
- контроль обновления документов версии программного продукта;
- проверку полноты проведения тестирования и отчетов о тестировании.

Сопроводитель должен **документально оформить** и представить заказчику:

- отчеты о проблемах (дефектах) и предложения о модификациях; результаты их анализа и варианты реализации изменений;
- результаты приемочных испытаний, верификации, аттестации и измерений характеристик качества новой версии программного продукта;
- отчеты об обеспечении характеристик качества программного продукта и результаты эксплуатационного тестирования;
- замечания заказчика и результаты взаимодействия с ним по устранению дефектов версии программного продукта;
- комплект актуальных проектных документов и документов результатов сопровождения;
- оценки корректности реализованной политики, графика и Программы квалификационного тестирования версии программного продукта;
- правильность оценок необходимых и использованных ресурсов;
- официальные рекомендации с указаниями о целесообразных последующих модификациях и создании новых версий ПС.

Снятие программного средства с эксплуатации и сопровождения должно быть подготовлено анализом, обосновывающим это решение. В анализе следует определить и экономически обосновать: возможность сохранения устаревшей версии комплекса программ, а также необходимость создания и применения новой версии программного продукта. Разработка нового ПС может проводиться для: усовершенствования модульности; упрощения сопровождения; обеспечения стандартизации или независимости от поставщика-продавца компонентов.

Перед прекращением сопровождения следует определить влияние снятия программного продукта с сопровождения на пользователей, установить программный продукт, заменяющий снимаемый (при его наличии) и определить обязанности по любым оставшимся вопросам последующей поддержки применения ПС. Пользователи должны получить уведомление о планах и работах по снятию с сопровождения и эксплуатации.

Стандарт **ISO 15846:1998** – Конфигурационное управление программными средствами – обобщает, детализирует и развивает основ-

ные концептуальные положения, представленные в стандартах **ISO 12207** и **ISO 10007**. Стандарт **ISO 10007:1995** – Руководящие указания при управлении конфигурацией – содержит общие рекомендации по управлению конфигурацией, которые целесообразно применять на протяжении всего жизненного цикла различных систем и видов продукции, с тем, чтобы обеспечить наглядность функциональных и физических характеристик и управление ими.

Основной задачей управления конфигурацией является документальное оформление и обеспечение полной наглядности текущей конфигурации продукции и выполнения требований к её физическим и функциональным характеристикам. Другая задача заключается в том, чтобы все лица, работающие над проектом, в любой момент его жизненного цикла использовали достоверную и точную информацию. Управление конфигурацией следует организовать так, чтобы персонал знал свои обязанности и имел достаточно независимости и полномочий для выполнения поставленных задач. Управление конфигурацией является управленческой дисциплиной, использующей техническое и административное руководство для разработки, производства и поддержки объектов конфигурации.

Шесть (6-ой – 11-ый) разделов начинаются с цитирования соответствующих шести базовых требований раздела 6.2 стандарта **ISO 12207**:

- подготовка процесса;
- определение конфигурации;
- контроль конфигурации;
- учет состояния конфигурации;
- оценка конфигурации;
- управление выпуском и поставка.

Процесс управления конфигурацией является процессом применения административных и технических процедур на всем протяжении жизненного цикла программных средств для: обозначения, определения и установления состояния базовой версии программных продуктов в системе; управления хранением, обращением и поставкой объектов.

Подготовка процесса включает разработку плана управления конфигурацией. План должен определять: работы по управлению конфигурацией; процедуры и график выполнения данных работ; организацию, ответственных за выполнение данных работ; связь данной

организации с другими предприятиями, например, по разработке и сопровождению программных средств. План должен быть документально оформлен и выполнен.

Определение конфигурации состоит в создании схемы обозначения программных объектов и их версий (объектов программной конфигурации), которые контролируются при реализации проекта. Для каждого программного объекта и его версий должны быть определены: документация, в которой фиксируется состояние его конфигурации; эталонные версии и другие элементы обозначения.

Контроль конфигурации, при этом должны быть выполнены: обозначение и регистрация заявок на внесение изменений; анализ и оценка изменений; принятие или непринятие заявки; реализация, верификация и выпуск измененного программного объекта. Для каждого изменения следует отслеживать аудиторские проверки, посредством которых анализируется каждое изменение, его причина и разрешение на его внесение. Должны быть выполнены контроль и аудиторская проверка всех доступных контролю программных продуктов, которые связаны с критическими функциями безопасности или защиты.

Учет состояний конфигурации включает подготовку протоколов управления и отчетов о состоянии, которые отражают состояние и хронологию изменения контролируемых программных объектов, состояние их конфигурации. Отчеты о состоянии должны учитывать количество изменений в данном проекте, последние версии программных объектов, обозначения выпущенных версий, количество выпусков и сравнения программных объектов различных выпусков.

Оценка конфигурации при этом должны быть определены и обеспечены: функциональная законченность программных объектов с точки зрения реализации установленных к ним требований; физическая завершенность программных объектов с точки зрения реализации в проекте и в программах всех внесенных изменений.

Управление выпуском и поставка включает официальный контроль выпуска и поставки программных продуктов вместе с соответствующей документацией. Оригиналы программ и документации должны сопровождаться в жизненном цикле. Программы и документация, связанные с обеспечением критических функций безопасности или защиты, должны обрабатываться, храниться, упаковываться и поставляться в соответствии с установленными правилами.

Процесс решения проблем (устранения дефектов) по стандарту **ISO 12207** (п.6.8) является процессом анализа и решения проблем (включая обнаруженные несоответствия), независимо от их происхождения или источника, которые обнаружены в ходе выполнения разработки, эксплуатации, сопровождения или других процессов. Целью данного процесса является обеспечение своевременного, ответственного и документируемого анализа и решения всех обнаруженных проблем (дефектов) и определения причин их возникновения.

Подготовка процесса устранения дефектов – процесс должен быть циклически замкнутым, обеспечивающим:

- своевременное документирование и ввод всех обнаруженных дефектов в процесс решения проблем;
- определение, анализ и возможное устранение причин их возникновения;
- учет и документирование состояний проблем;

Решение обнаруженных проблем: при выявлении проблем в программном продукте или работе должен быть подготовлен отчет, описывающий каждую выявленную проблему. Отчет по проблеме должен являться составной частью вышеописанного процесса, охватывая вопросы: выявления проблем; их исследования, анализа и решения, а также причин их возникновения; определения тенденций, способствующих возникновению проблем.

Модификация, учет и тиражирование версий **требует больших затрат**. Поэтому при выпуске каждой новой базовой версии разработчики стремятся обеспечить преемственность ее функций и компонентов с предыдущими версиями, а также рассматривается возможность и подготавливается решение для возможного прекращения модификаций некоторой устаревшей версии ПС или ее конкретных компонентов. Для реализации на практике приведенных выше концепций и процедур, требований и планов сопровождения и управления конфигурацией программных средств необходимы организационные мероприятия, гарантирующие участникам проектов определенную **культуру, дисциплину разработки и выполнения модификаций**. Такая организационная система должна обеспечивать специалистам разной квалификации и роли в проекте, возможность взаимодействия при решении требуемых комплексных задач, для накопления, хранения и обмена упорядоченной информацией о состоянии и изменениях компонентов проекта.

Лекция 5

ПОДГОТОВКА ПРОИЗВОДСТВА ПРОГРАММНЫХ СРЕДСТВ И СИСТЕМЫ КАЧЕСТВА ПРЕДПРИЯТИЯ К СЕРТИФИКАЦИИ

Подготовка технологии производства программных продуктов и системы качества предприятия к сертификации

Целеустремленная, производственная деятельность разработчиков-поставщиков должна быть направлена на *удовлетворение требований заказчиков и пользователей программных продуктов* при их применении по прямому назначению – рис. 5.1.

Подготовка производства программных средств и системы качества предприятия к сертификации:

- подготовка технологии производства и системы качества предприятия программных продуктов к сертификации:
 - основная цель современных технологий обеспечения жизненного цикла комплексов программ;
 - методы стандартизированного описания требований и оценивания характеристик качества производственных процессов;
 - цели применения выборки из всей совокупности действующих стандартов и их адаптации в профиль стандартов;
- адаптация базовых стандартов управления производством **ISO 12207** и системой качества **ISO 90003** программных средств для сертификации:
 - задачи адаптации стандартов и применения ключевых характеристик проекта;
 - адаптация базового стандарта **ISO 90003** на основе адаптированной версии стандарта **ISO 12207**.

Рис.5.1

Скоординированное применение методов, стандартов и средств в процессах производства, развития и применения комплексов программ позволяет исключать многие виды дефектов или значительно ослаблять их влияние. Тем самым **экономические характеристики** и уровень достигаемого качества становятся **предсказуемыми и управляемыми**, непосредственно зависящими от ресурсов, выделяемых на их достижение, а главное, от системы качества и эффективности технологии, используемых на всех этапах жизненного цикла.

Основными целями упорядочивания, регламентирования этапов и процессов производства в жизненном цикле программных комплексов являются:

- снижение трудоемкости, длительности, стоимости и **улучшение всех характеристик** программных продуктов;
- повышение качества разрабатываемых и/или применяемых компонентов и программных комплексов в целом при их производстве, сопровождении и эксплуатации;
- обеспечение возможности расширять комплексы программ по набору прикладных функций, совершенствовать и масштабировать их в зависимости от изменения решаемых задач, внешней среды и потребностей заказчика;
- обеспечение переносимости прикладных программ и данных между разными аппаратными и операционными платформами и повторного использования программных компонентов в различных проектах.

Методической основой технологии производства комплексов программ, регламентирующей деятельность специалистов, является типовой технологический процесс создания промышленной продукции. Он отражается **набором этапов и операций** в последовательности их выполнения и взаимосвязи, обеспечивающих упорядоченное ведение работ на всех стадиях от инициирования проекта и подготовки технических требований до завершения испытаний или применения версии программного продукта. Индустриализация технологий создания комплексов программ базируется на стандартизации процессов производства, их структурного построения и интерфейсов с операционной и внешней средой. Для этого с самого начала разработки должны определяться **задачи, состав и этапы работ**, необходимые для достижения конечной цели, а также требуемые для их выполнения материальные и людские **ресурсы**. Технические и управ-

ленческие проверки экономических характеристик, анализ качества результатов промежуточных работ и компонентов, а также корректности их взаимосвязей должны обеспечивать руководителям и всем разработчикам уверенность достижения требуемого конечного результата производства – **получение продукта требуемого качества**. Для этого необходима **подготовка технологии** производства программного продукта:

- приобретение или разработка и освоение технологии, структуры, свойств и экономики технологических процессов, состава и форм отчетных документах о компонентах и процессах производства продукта;
- предварительный выбор и освоение среды проектирования, методов, инструментальных средств автоматизации и стандартов производства;
- формирование базы данных производства, концептуальное, логическое и физическое распределение информационных и программных компонентов и документов проекта;
- определение организационной структуры и числа специалистов, а также потребностей в субподрядных организациях для реализации проекта;
- разработка предварительного плана проектирования и производства с учетом возможного технического, экономического и планового рисков;
- подготовка комплекта руководств для детального проектирования, программирования и тестирования компонентов комплекса программ, согласование словарей понятий и идентификаторов конфигурации продукта.

Достижение высокого качества комплексов программ существенно зависит от **качества технологии, стандартов и инструментальных средств**, используемых разработчиками для обеспечения всего жизненного цикла. Оценивание достоинств (зрелости) технологической базы позволяет прогнозировать возможное качество программных продуктов и ориентировать заказчика и пользователей при выборе разработчика и поставщика для определенного проекта с требуемыми характеристиками. Поэтому определение уровня технологической поддержки процессов жизненного цикла, организационного и инструментального обеспечения производства, непосредственно

связано с прогнозированием реальных или возможных экономических характеристик и качества конкретного комплекса программ.

В современных автоматизированных технологиях создания и совершенствования сложных комплексов программ, с позиции обеспечения их качества можно выделить **базовые методы и средства**, позволяющие:

- создавать программные модули и функциональные компоненты высокого, гарантированного качества;
- предотвращать дефекты проектирования за счет систем обеспечения качества, эффективных технологий и инструментальных средств автоматизации жизненного цикла производства комплексов программ;
- предотвращать, обнаруживать и устранять различные дефекты и ошибки проектирования, разработки и сопровождения программ путем верификации и систематического тестирования на всех этапах жизненного цикла комплекса программ;
- удостоверить достигнутые значения качества функционирования программного продукта в процессе их **испытаний и сертификации** перед передачей заказчику и в регулярную эксплуатацию пользователям.

Унификация процессов производства комплексов программ оправдана, если она обеспечивает экономию времени создания и/или применения продукта. Каркас процессов комплекса программ – это структура, организующая процессы взаимодействия компонентов, используемых при построении функций, образующих версии программных продуктов. Построение каждой части архитектуры обычно включает множество процессов, даже в наиболее простых продуктах. Разработка каркаса процессов целесообразна для версий программных продуктов, когда затраты могут эффективно окупаться при производстве множества вариантов – версий продуктов. Чтобы быть полезными, формальные процессы продукта необходимо адаптировать к специфике конкретного проекта. Поскольку каждое предприятие и каждый продукт являются во многих отношениях уникальными, невозможно сформулировать процесс структурирования, который удовлетворит нужды всех проектов.

Ключ к эффективному использованию процесса структурирования состоит в том, чтобы сделать его специфичным для данного проекта, чтобы он содержал **только используемые структуры**, органи-

зованные в соответствии с нуждами проекта. Изменения, порожденные адаптацией такого рода, часто довольно существенны и дают результат, мало похожий на первичную структуру. Структурирование должно содержать детальную информацию о проекте: конфигурирование компонентов комплекса, наборы инструкций разработчикам, указатели на формирование документации и программные интерфейсы. Кроме того, необходимы, имена ключевых контактных специалистов, необходимые для выполнения важнейших шагов процесса, отслеживания дефектов или сборок, фиксации изменений, контроля стиля кодирования и взаимного просмотра кода, а также многие другие детали, специфичные для взаимодействия компонентов любого крупного проекта.

Возрастание сложности и ответственности современных задач, решаемых крупными системами, а также возможного ущерба от недостаточного качества программного продукта, значительно повысило актуальность освоения методов *стандартизированного описания требований и оценивания характеристик качества* производственных процессов, компонентов и продуктов на различных этапах жизненного цикла. Выявилась необходимость систематизации реальных характеристик качества, применения стандартов для выбора из них необходимой номенклатуры и требуемых значений для конкретных производств комплексов программ. Обещания разработчиков в контрактах с заказчиками создать высококачественные программные продукты в согласованные сроки во многих случаях *не выполняются*, как вследствие различий в понимании требуемого качества, так и вследствие неумения оценить ресурсы, необходимые для достижения заданного заказчиком качества программного продукта. *Стратегической проблемой* в жизненном цикле современных систем стало обеспечение требуемого качества крупных программных продуктов при реальных ограничениях на использование ресурсов [3, 4, 12].

Для сокращения стоимости и улучшения качества комплексов программ стандарты производства, жизненного цикла и системы обеспечения качества ПС *следует адаптировать к характеристикам предприятия или индивидуального проекта*. Должны быть определены характеристики окружения проекта, которые могут воздействовать на адаптацию. Некоторыми из этих характеристик могут быть: модель жизненного цикла; функции жизненного цикла текущей информационной системы; требования к системе и программному

обеспечению; организационные линии поведения, процедуры и стратегия предприятия; размер, критичность и типы системы, а также программного средства; количество задействованного персонала и сторон в проекте. Должны быть запрошены у заказчика или потенциальных пользователей те исходные характеристики организаций, на которые могут повлиять решения по адаптации. В процесс адаптации должны быть включены особенности: пользователей; поддерживающего персонала; руководителей контракта; поставщиков; потенциальных покупателей. Процессы, работы и задачи ЖЦ необходимо селектировать в стандартах с учетом риска, размера, критичности проекта. Следует выбрать процессы, работы и задачи, которые должны быть обязательно выполнены. Они должны включать в себя перечень документов, которые нужно разработать и информацию о персональной ответственности за них. Выбранные процессы, работы и задачи, но не обеспечиваемые данным стандартом, следует оговаривать в самом контракте.

Организационные процессы жизненного цикла следует оценивать, чтобы определить, будут ли они соответствовать выбранным процессам, работам и задачам ЖЦ ПС. При адаптации стандартов должны учитываться, в частности, следующие факторы: стоимость, планирование, производительность, размер и интерфейс с человеком. Все решения по адаптации должны быть документированы вместе с обоснованием их целесообразности.

В стандартах на жизненный цикл и системы качества ПС содержание этапов, работ и результирующих их документов **отражено на методологическом и концептуальном уровне**. Методы и средства реализации каждой работы в этих стандартах не раскрываются и адресуются к специальным, детализирующим нормативным документам различного уровня. Ряд характерных особенностей этапов и работ принципиально не позволяет создать полную гамму международных стандартов, поддерживающих каждый этап ЖЦ ПС. Например, быстро оснащающиеся различными методами и средствами автоматизации, этапы системного анализа, моделирования и предварительного проектирования не позволяют стабилизировать основу этих процессов, достаточную для формализации на уровне международных стандартов.

Наиболее обеспечены стандартами рутинные этапы программирования компонентов и документирования программных средств. Ес-

тественно, отсутствуют и вряд ли вскоре будут созданы детальные стандарты для самых творческих этапов системного анализа, предварительного и детального проектирования, на которых доля творческих работ по трудоемкости превышает 80%, а технические работы определяются фирменными инструкциями для применяемых средств автоматизации. Также велик уровень творческого труда (70-80%) на этапах интеграции, комплексной отладки и испытаний ПС в целом. В этой области степень стандартизации не превышает 50%, и в значительной степени покрывается общими положениями стандартов на ЖЦ ПС. Стандартизованы работы, выполняемые большим числом специалистов относительно невысокой квалификации, и почти не стандартизованы особо сложные творческие работы, требующие наивысшей квалификации специалистов.

При создании сложных проектов ПС и обеспечении их ЖЦ *целесообразно применять выборку из всей совокупности действующих стандартов – профиль*, а имеющиеся весьма обширные пробелы в стандартизации заполнять технологическими документами и инструкциями, регламентирующими применение выбранных или созданных средств автоматизации разработки ПС. В результате на начальном этапе проектирования следует сформировать весь комплект документов – профиль, разного уровня формализации и утверждения, обеспечивающий регламентирование всех этапов и работ при создании определенных проблемно-ориентированных ПС. Для реализации положений этих документов должны быть выбраны инструментальные средства, совместно образующие взаимосвязанный комплекс технологической поддержки и автоматизации ЖЦ и не противоречащие предварительно скомпонованному набору нормативных документов.

При разработке системы качества предприятия и применении ее в процессе производства программных продуктов, а также *для сертификации на соответствие стандартам серии ISO 9000, необходимо подготовить, адаптировать и внедрить следующий комплект международных стандартов: ISO 12207; ISO 90003; ISO 10013; ISO 10005*. На основе этих стандартов должен быть разработан детальный технологический процесс обеспечения ЖЦ и система обеспечения качества программных средств, поддержанные комплексом инструментальных средств и должностных инструкций для практической реализации положений адаптированных международных стандартов. Это позволит *сформировать систему качества пред-*

приятая, гарантирующую высокое качество проектов и всего жизненного цикла сложных комплексов программ, готовых для сертификации.

Адаптация для сертификации базовых стандартов управления производством ISO 12207 и системой качества ISO 90003 программных средств

*Основные рекомендации по адаптации стандартов, регламентирующих ЖЦ и системы качества ПС, отражены в Приложениях А и В к стандарту ISO 12207 (см. рис. 5.1). В Приложении А изложены общие нормативные положения, которые более подробно раскрываются в Приложении В, из которого ниже приводятся **основные фрагменты**. В этом Приложении рекомендуются **два уровня адаптации** стандартов ЖЦ ПС:*

- адаптация первого уровня к данной деловой проблемно - ориентированной области; например, авиационной, атомной, медицинской, военной, национальной или организационной;
- адаптация второго уровня, которую следует осуществлять для каждого предприятия, специфического проекта или контракта.

В стандарте отмечается, что лица, включенные в любую службу жизненного цикла проекта или процесса, должны проводить оценки и селекцию характеристик программных продуктов и работ. Эти оценки следует выбирать и адаптировать в зависимости от области действия, величины, комплексности и критичности проекта или организации.

В разделе В.3 Приложения стандарта ISO 12207 изложены рекомендации, которые содержат:

Оценки внутреннего процесса осуществляются персоналом, выполняющим назначенные задачи процесса во время своих повседневных работ.

Верификацию и проверки достоверности осуществляются покупателями, поставщиками или независимой стороной для того, чтобы провести верификацию или установить достоверность качества продуктов разной глубины зависимости от проекта. Эти оценки не дублируют и не заменяют другие оценки, а, напротив, дополняют их.

Объединенные наблюдения и ревизии осуществляются на совместном форуме контролирующих и наблюдаемых сторон для того,

чтобы оценить статус и согласованность продуктов и работ на этапе предварительного соглашения по планированию.

Гарантия качества - сертификация осуществляется лицами, независимыми от персонала, напрямую ответственного за разработку программного продукта и за выполнение процесса. Целью этого является представление независимой гарантии согласованности программных продуктов и процессов с требованиями контракта и с привязкой к установленным планам.

Улучшение осуществляется организацией для эффективного управления и самоусовершенствования ее процессов. Проводится без учета требований проекта или контракта.

В параграфах раздела **В.4. Вопросы адаптации и применения** в общих чертах проводится широкое рассмотрение задач адаптации и применения ключевых характеристик проекта. Ни рассмотрение, ни характеристики не являются исчерпывающими и представляют только ограниченную точку зрения.

Организационные работы – определяется, какие из организационных работ уместны и применимы: работы, связанные с компьютерными языками, сохранностью и безопасностью; требования к резерву аппаратуры и управление риском. Следует выделить и сохранить пункты данного стандарта, относящиеся к этим организационным службам.

Стратегия приобретения – определяется, какие из стратегий приобретения компонентов уместны и применимы для данного проекта: типы контракта; включение субучастников контракта и агентов по верификации и по проверке достоверности; уровень связи покупателя с участниками контракта и развитие возможностей участников контракта.

Концепция поддержки – определяется, какие из концепций поддержки уместны и применимы: ожидаемая длительность поддержки; предполагаемый уровень изменений; кто будет осуществлять поддержку - покупатель или персонал поддержки. Если программный продукт предполагает иметь длительный жизненный цикл поддержки, или ожидаются значительные изменения, то следует рассмотреть все требования к документации. Рекомендуется иметь автоматизированную разработку документации.

Характеристики уровня системы – определяется, какие из характеристик уровня системы необходимы и применимы: количество

элементов программного обеспечения, типы, размер и критичность программных продуктов, технические риски. Если программный продукт имеет много программных элементов, компонентов и единиц, то Процесс Разработки следует тщательно адаптировать к каждому программному элементу. Следует рассмотреть все требования к интерфейсам и интеграции. Определяется, какие типы программных продуктов включены, так как различные типы программных продуктов могут требовать различных решений по адаптации. Рассмотрены некоторые примеры условий адаптации. Если система зависима от качества, корректного выполнения и от завершения в срок программного продукта, тогда больший контроль управления должен быть обеспечен во время тестирования, ревизий, верификаций, проверок достоверности и т.д.

Разработка программного продукта может иметь технические риски. Если используется несовершенная технология, разрабатываемый программный продукт является беспрецедентным или сложным или он содержит критические требования, такие как сохранность и безопасность, то может потребоваться строгая спецификация, конструирование, тестирование и оценка. Могут оказаться важными независимая верификация и проверка достоверности – сертификация.

Детальную структуру каждого документа системы качества и адаптацию базового стандарта ISO 90003 целесообразно разрабатывать или уточнять предприятием на основе адаптированной версии стандарта ISO 12207 в соответствии с их традициями, используемой технологией и особенностями проектируемого ПС. Некоторые документы автоматизировано формируются инструментальными средствами, поддерживающими ЖЦ ПС, которые определяют их структуру и содержание. Ряд дополнительных документов при проектировании ПС формируется современными CASE-средствами. В результате номенклатура, структура и содержание документов системы качества может значительно варьироваться и расширяться. Формализация структуры и типового содержания каждого документа должна позволять контролировать результаты и качество выполненных работ.

Формирование Руководства по планированию качества программных средств для сертификации на основе стандарта ISO 10005

Так же как остальные базовые стандарты систем качества программных средств, стандарт **ISO 10005:1995** для эффективного применения *следует адаптировать к характеристикам объектов и среды применения конкретного предприятия или проекта* – рис. 5.2. Разработка Плана качества выгодна предприятиям для повышения уверенности, что установленные требования будут выполнены, процессы находятся в управляемом состоянии и обеспечена необходимая деятельность специалистов. На основе этой деятельности организация может лучше понять возможности для улучшения процессов.

Прежде всего, необходимо оставить в версии этого стандарта разделы и положения, которые использовать непосредственно для обеспечения качества производства конкретных программных средств. Из первичного текста стандарта ниже выделены минимально необходимые рекомендации для создания Плана обеспечения качества ЖЦ ПС. Могут быть исключены обычные первые разделы стандартов: область применения, нормативные ссылки и определения, а также информативные приложения.

Адаптация стандарта для конкретных предприятий или проектов может выполняться путем сокращения и конкретизации некоторых представленных ниже положений. После этого План качества должен быть сформирован и оформлен как самостоятельный регламентирующий документ, согласован с заказчиком, утвержден руководством предприятия и доведен до сведения всех участников проекта для практического применения. При сертификации системы качества предприятия должно проверяться наличие и практическое использование всех положений утвержденной версии Плана качества.

План качества должен показывать, как общие документированные процедуры поставщика привязаны и применимы к дополнительным процедурам, относящимся к определенной продукции или проекту, для достижения поставленных целей в области качества ПС.

Уполномоченная группа, в которую должны входить представители всех заинтересованных служб предприятия поставщика, *должна проанализировать План качества на предмет адекватности и официально утвердить его*.

Формирование Руководств по планированию и документированию менеджмента качества программных средств на основе стандартов:

- формирование Руководства планирования менеджмента системы качества программных средств **ISO 10005** для сертификации:
 - идентификация потребности в плане качества;
 - процесс подготовки и сфера действия плана менеджмента качества программных средств;
 - содержание плана менеджмента качества предприятия;
 - документирование плана менеджмента качества программных средств;
 - политика и цели в области качества, обязательства предприятия в отношении качества программных средств;
 - ответственность и полномочия предприятия в отношении менеджмента качества программных средств;
 - ресурсы для обеспечения менеджмента качества;
- формирование Руководства документирования системы качества программных средств **ISO 10013** для сертификации:
 - цели и политика документирования системы менеджмента качества программных средств;
 - структура и содержание документации системы менеджмента качества программных средств;
 - область применения, ответственность и полномочия специалистов системы качества программных средств;
 - документированные процедуры, структура и содержание системы менеджмента качества;
 - подготовка документов системы менеджмента качества программных средств;
 - утверждение, выпуск и управление документами системы менеджмента качества программных средств.

Рис. 5.2

Если План качества оговорен контрактом, его следует, как правило, представлять до начала соответствующих работ. Предприятие должно *идентифицировать потребность в Плане качества*. Во многих случаях план качества может быть полезным и необходимым:

- при определении действий, предусмотренных системой менеджмента качества организации в конкретной ситуации;

- для целей обеспечения соответствия законодательным и обязательным требованиям или требованиям потребителя;
- для демонстрации выполнения требований к качеству для внутренних и/или внешних потребителей;
- в деятельности организации по выполнению требований к качеству и достижению целей в области качества;
- для оптимизации использования ресурсов в соответствии с целями в области качества.

Если предприятие принимает решение о разработке плана качества, то должны быть идентифицированы необходимые для подготовки плана качества **входные данные**, область применения плана качества и область применения других аналогичных документов. При определении области применения плана качества следует учитывать следующие факторы:

- процессы и характеристики качества, которые должны быть включены в план качества для конкретной ситуации;
- требования потребителей или других заинтересованных сторон (внутренних или внешних) по включению процессов, не применяемых в конкретной ситуации, но необходимых для повышения уверенности в выполнении этих требований;
- степень поддержки плана качества другими документами системы менеджмента качества.

При **подготовке плана менеджмента качества** должно быть четко **установлено лицо**, ответственное за его разработку и выполнение. План качества должен быть подготовлен при непосредственном участии лиц, вовлеченных в работу по конкретной ситуации внутри и/или вне предприятия. В плане качества должно быть указано, как необходимые действия должны быть оформлены: указаны непосредственно в плане или путем ссылки на соответствующие документированные процедуры или другие документы. Большая часть документов, используемых при подготовке плана качества, может содержаться в документации системы менеджмента качества организации, включая руководство по качеству и документированные процедуры. Эти документы, при необходимости, могут быть специально подобраны, **адаптированы и/или дополнены**.

При подготовке плана качества предприятие должно согласовать и определить функции, ответственность и полномочия персонала в пределах организации, а также при работе с потребителем, госу-

дарственными органами или другими заинтересованными сторонами. Лица, ответственные за план качества, должны обеспечивать знание и понимание персоналом целей и проблем в области качества, а также умение использовать средства управления, применяемые в соответствии с планом качества. Содержание и форма плана качества должны быть совместимыми с областью применения и входными данными для плана качества и потребностями пользователей. Уровень детализации плана качества должен быть согласован с требованиями потребителей, методами выполнения и сложностью деятельности организации.

Предприятие может представлять план качества в любой форме, например в виде текста, таблицы, матрицы документа, карты процесса, блок-схемы работы или руководства. Эти документы могут быть представлены в электронном и/или бумажном виде. План качества может включать в себя несколько документов, каждый из которых представляет собой план качества отдельного аспекта конкретного производства.

Область применения должна быть заявлена в плане качества, включая:

- установление цели и ожидаемого результата в конкретной ситуации;
- определение аспектов конкретной ситуации, к которым будет применен план качества, включая особые ограничения по применению;
- установление параметров области действия (например, число измерений, состояние рынка, доступность ресурса или статус сертификата на систему менеджмента качества).

В плане качества должны быть установлены **цели в области качества** программных средств для конкретной ситуации и способы их достижения. Цели в области качества для конкретной ситуации могут быть установлены:

- характеристик качества в конкретной ситуации;
- важных вопросов, связанных с удовлетворенностью потребителя или других заинтересованных сторон;
- возможностей улучшения методов работы.

В плане качества конкретной производства должны быть идентифицированы **специалисты, ответственные** за следующие действия:

- обеспечение планирования, внедрения, управления и мониторинга продвижения действий, требуемых системой менеджмента качества или контрактом;

- определение последовательности и взаимодействия процессов, применимых в конкретной ситуации;
- установление требований к обмену информацией между всеми вовлеченными функциональными подразделениями, поставщиками и потребителями, а также урегулированию проблем, возникающих при взаимодействии между ними;
- наделение соответствующих лиц полномочиями принятия решения об исключении отдельных требований системы менеджмента качества;
- управление корректирующими и предупреждающими действиями;
- анализ и одобрение изменений или отклонений от плана качества.

Для **документации и данных**, применимых в конкретной ситуации, в плане менеджмента качества следует указывать;

- способ идентификации документов и данных;
- лиц, анализирующих и утверждающих документы и данные;
- лиц, которым будут направляться документы и данные или уведомление об их доступности;
- способ получения доступа к документам и данным.

План качества должен содержать **перечень установленных отчетов** и методы их поддержания в рабочем состоянии. Такие отчеты могут включать в себя: записи по анализу проекта, контролю и испытаниям, измерению процессов, график работ, графические материалы, расписание заседаний. Предприятие должно рассмотреть следующие вопросы;

- способ, место и время ведения отчетов;
- требования договоров, законодательные и обязательные требования и способы их выполнения;
- носители, на которых должны вестись отчеты – записи (документальная копия или электронные носители);
- способ установления и выполнения требований к четкости, хранению, изменению места хранения, распоряжению и конфиденциальности записей;
- методы, используемые для своевременного обеспечения доступности записей – отчетов;
- записи, предоставляемые потребителю, время и средства их предоставления.

План качества должен устанавливать *тип и число ресурсов*, необходимых для успешного выполнения плана. Эти ресурсы могут включать в себя человеческие ресурсы, инфраструктуру и производственную среду. Если имеются ограничения в доступности отдельных видов ресурсов, то в плане качества может быть указан способ выполнения требований (за счет использования конкурентной продукции, проектов, процессов или контрактов).

Аудиты используются для нескольких целей:

- мониторинг внедрения и результативность планов качества;
- мониторинг и верификация соответствия установленным требованиям;
- наблюдение за поставщиками;
- независимая объективная оценка удовлетворения потребностей потребителей или других заинтересованных сторон.

План качества необходимо анализировать на предмет его адекватности и результативности. План качества должен быть *формально одобрен ответственным лицом* или группой лиц, состоящей из представителей соответствующих функциональных подразделений предприятия. В соответствии с договором план качества может быть представлен потребителю для анализа и принятия или в процессе консультаций при подготовке договора, или после заключения договора. После заключения договора план качества должен быть проанализирован и при необходимости доработан и пересмотрен в соответствии с изменениями требований, которые могли произойти в процессе консультаций при подготовке договора. Предприятие несет ответственность за *мониторинг соответствия* текущей деятельности планам качества.

Формирование Руководства по документированию для сертификации системы менеджмента качества программных средств на основе стандарта ISO 10013

Обязательным требованием стандартов серии **ISO 9000** является документирование системы менеджмента качества предприятий. Настоящий стандарт направлен на обеспечение процессного подхода при разработке и внедрении системы менеджмента качества и улучшении ее результативности. Для результативного функционирования предприятие должно идентифицировать многочисленные взаимосвязанные виды деятельности и управлять ими. При документировании системы менеджмента качества предприятие может определить любое число и со-

став документов, необходимых для демонстрации результативного планирования, функционирования, управления и постоянного улучшения *системы менеджмента качества и ее процессов*. Документирование может охватывать всю деятельность предприятия или отдельные его аспекты. Требования, устанавливаемые в документации, зависят от вида и характера продукции и процессов, условий контракта, установленных законодательных и обязательных требований [15].

Содержание документов системы менеджмента качества и установленные в ней требования должны соответствовать стандартам в области качества. Планирование качества является одним из основных видов деятельности при внедрении системы менеджмента качества. Документы по планированию качества могут регламентировать планирование, подготовку к внедрению системы менеджмента качества, включая организационные мероприятия, графики выполнения работ и ресурсы, необходимые для достижения целей в области качества (см. рис. 5.2).

Классификация документации системы менеджмента качества может быть построена на основе структуры производственных процессов предприятия, структуры внедряемого стандарта качества или их комбинации. Структура документов системы менеджмента качества может быть иерархической. Подобная структура способствует внедрению, поддержанию в рабочем состоянии и лучшему пониманию персоналом требований к документации системы менеджмента качества.

Степень документирования системы менеджмента качества предприятия может различаться в зависимости от следующих факторов:

- размера и видов объектов деятельности организации;
- сложности процессов и форм их взаимодействия;
- компетентности персонала.

Определения системы менеджмента качества должны соответствовать терминам и определениям по **ISO 9000** со ссылкой на этот стандарт или общепринятые термины и определения. Система менеджмента качества обычно включает в себя *следующие документы*:

- политику и цели в области качества;
- руководство по качеству;
- документированные процедуры производства;

- рабочие инструкции специалистам;
- планы качества;
- технические условия продуктов;
- записи – отчеты.

Документирование производства и системы менеджмента качества должно **позволять осуществлять**:

- описание системы менеджмента качества предприятия;
- обеспечение необходимой информацией взаимодействующих подразделений с целью лучшего понимания взаимосвязей между ними;
- доведение до сведения персонала обязательств со стороны руководства в области качества;
- содействие в обеспечении осведомленности персонала об актуальности и важности его деятельности;
- обеспечение взаимопонимания между персоналом и руководством предприятия;
- установление порядка выполнения работ для достижения установленных требований;
- обеспечение объективных свидетельств выполнения установленных требований;
- обеспечение четкой и результативной структуры выполняемых производственных действий;
- обеспечение базы для подготовки вновь нанимаемого персонала и необходимой переподготовки всего персонала организации через запланированные интервалы времени;
- обеспечение последовательности выполнения операций на основе документов по процессам;
- повышение доверия к предприятию со стороны потребителей на основе документированных производственных процедур системы;
- предоставление заинтересованным сторонам информации о возможностях организации;
- создание основы для проведения аудита системы менеджмента качества;
- обеспечение основы для оценивания и постоянной поддержки результативности системы менеджмента качества в соответствии с установленными требованиями.

Политика и цели в области качества должны быть документально оформлены в виде самостоятельных документов или включены

в руководство по качеству. Каждая организация должна разработать свое руководство по качеству. Настоящий стандарт допускает гибкий подход при определении структуры, формата, содержания или методов документирования системы менеджмента качества предприятия. Для небольших организаций в руководство по качеству целесообразно включать полное описание системы менеджмента качества со всеми документированными процедурами, требуемыми **ISO 9001**. Для крупных организаций может быть разработано несколько руководств по качеству и/или может быть создана более сложная иерархическая структура документов.

Руководство по качеству должно содержать **область применения системы** менеджмента качества, обоснование и детали любых исключений, документированные процедуры или ссылки на них, описание процессов системы менеджмента качества. Руководство по качеству должно включать в себя информацию о предприятии, в том числе ее наименование, адрес и контактную информацию. В руководстве по качеству должна быть сделана ссылка на стандарт, на основе которого разрабатывается система менеджмента качества. Руководство по качеству должно содержать точные сведения о его статусе, датах рассмотрения, утверждения и пересмотра.

Руководство по качеству должно включать в себя **политику и цели в области качества** или ссылки на них. Решение о включении политики и/или целей в области качества в руководство по качеству принимает высшее руководство организации. Показатели результативности выполнения целей могут быть включены в другие документы системы менеджмента качества предприятия. Политика в области качества должна содержать обязательства соответствия установленным требованиям и постоянного повышения результативности системы менеджмента качества. В Руководстве по качеству должна быть приведена **структура организации**. Ответственность, полномочия и взаимодействия могут быть приведены в организационных схемах, картах процессов и/или рабочих инструкциях. Руководство по качеству должно содержать описание системы менеджмента качества и ее применения в предприятии, а также описание процессов и их взаимодействия, должны быть также включены документированные процедуры или ссылки на них.

Структура и формат документированных процедур должны быть представлены организацией в виде текстов, схем, карт процессов,

таблиц, их комбинации или в другой приемлемой форме в зависимости от потребностей организации. Документированная процедура должна содержать необходимую информацию и иметь уникальную идентификацию для обеспечения прослеживаемости. В документированных процедурах могут быть сделаны ссылки на рабочие инструкции, определяющие порядок выполнения работ. Документированные процедуры обычно описывают деятельность, включающую в себя несколько функций, в то время как рабочие инструкции обычно относятся к заданиям в рамках одной функции.

Организация должна **идентифицировать ответственность и полномочия** и/или функциональные обязанности специалистов, а также взаимосвязь персонала с процессами и действиями, входящими в процедуру. Ответственность и полномочия могут быть представлены в виде карты процесса или текста. Уровень детализации документов зависит от сложности вида деятельности, используемых методов, уровня квалификации и подготовки персонала, необходимых для выполнения работ. Независимо от уровня при описании видов деятельности должны быть рассмотрены следующие аспекты:

- определение потребностей предприятия, ее потребителей и поставщиков;
- представление процессов в виде текста и/или карт процесса по видам деятельности;
- определение необходимых работ и распределение ответственности, назначение ответственных лиц, рамки их функциональных обязанностей, цели, место, время и способ выполнения этих работ;
- описание управления процессами и идентифицированными видами деятельности;
- определение потребностей в ресурсах, необходимых для выполнения установленной деятельности;
- определение документов, регламентирующих виды деятельности.

План по качеству является частью документации системы менеджмента качества или других планов и/или Программ предприятия. План по качеству используют в документированной системе менеджмента качества организации для установления порядка применения системы качества в конкретной ситуации. План по качеству идентифицирует и документирует способ выполнения предприятием уникальных требований к продукции, процессу, проекту или особым требова-

ний контракта. План по качеству может включать в себя уникальные процедуры, рабочие инструкции и/или записи,

Предприятие должно использовать *внешние документы* и установить методы управления ими в рамках системы менеджмента качества. Примерами таких документов являются чертежи потребителей, технические условия, законодательные и обязательные требования, стандарты, кодексы и руководства по эксплуатации. Это необходимо для обеспечения заинтересованности персонала, а также лучшего понимания специалистами установленных требований.

Анализ и отмена (при необходимости) существующих в предприятии документов, а также используемых ссылок может значительно сократить сроки разработки документов системы менеджмента качества. Кроме того, анализ действующих документов может помочь выявить области несоответствий в системе менеджмента качества для устранения, которых должны быть внесены в документы необходимые изменения.

Предприятия, *внедряющие или планирующие внедрение* системы менеджмента качества, должны:

- определить процессы, необходимые для результативного внедрения системы менеджмента качества;
- определить последовательность и взаимодействие этих процессов;
- документировать процессы, насколько это необходимо, для обеспечения их результативного функционирования и управления.

На основе анализа процессов предприятие должно определить необходимое число документов системы менеджмента качества. *Действия по документированию* системы менеджмента качества включают:

- определение перечня документов системы менеджмента качества, необходимых в соответствии с требованиями выбранного организацией стандарта;
- сбор данных о действующей системе менеджмента качества и процессах с помощью анкетирования или интервью;
- определение документов действующей системы менеджмента качества с целью определения возможности ее использования;
- организация обучения персонала, занятого разработкой документов, программа обучения должна включить порядок документирования, изучение требований стандартов системы менеджмента качества

и/или других необходимых требований;

- организация запроса и получения дополнительных документов и ссылочных документов из функциональных подразделений;
- определение структуры и формата разрабатываемых документов;
- разработка структурной схемы (карт) процессов, на которые распространяется система менеджмента качества;
- проведение анализа блок-схемы и карт процессов и внедрение возможных улучшений;
- аттестация документов по результатам опытного внедрения;
- рассмотрение и утверждение документов до их выпуска.

До выпуска документы должны поступить на **рассмотрение уполномоченным лицам** предприятия с целью контроля ясности, точности и правильности их изложения и построения. Чтобы учесть сложившуюся в организации практику, предполагаемые пользователи также должны иметь возможность ознакомиться с документами и представить свои замечания. Выпускаемые документы должны быть утверждены ответственными за их внедрение лицами. При рассылке документов уполномоченные лица должны обеспечить поступление к соответствующему персоналу документов, содержащих необходимые для его работы данные. Правильной рассылке и управлению документацией может, например, способствовать присвоение учетных номеров каждому экземпляру документов, предназначенных для определенных пользователей.

Предприятие должно установить процесс инициирования, разработки, рассмотрения, контроля и регистрации изменений документов. Внесение изменений должно проходить ту же процедуру рассмотрения и утверждения, что и при первоначальной разработке документов. При управлении выпуском документов и внесением в них изменений очень важно обеспечить надлежащее утверждение документов уполномоченными лицами. Установленная процедура должна обеспечивать применение только надлежащих документов. Пересмотренные документы должны быть заменены последней версией. С целью обеспечения пользователей надлежащими версиями утвержденных документов рекомендуется составить перечень основных документов и установить статус их пересмотра.

Требования к элементам систем качества должны быть установлены в соответствующих базовых стандартах, применяемых данным предприятием – адаптированные версии **ISO 90003** и **ISO12207**. Ре-

комендуется, чтобы во всех применимых случаях описание элементов системы качества осуществлялось в той же последовательности, что и в выбранном базовом стандарте. После выбора соответствующего стандарта каждое предприятие определяет приемлемые для себя элементы системы качества и, на основании требований этих элементов в стандарте, определяет, как оно намерено применять, совершенствовать и управлять каждым выбранным элементом.

Лекция 6

СЕРТИФИКАЦИЯ ПРОЦЕССОВ ПРОИЗВОДСТВА ПРОГРАММНЫХ ПРОДУКТОВ И СИСТЕМ КАЧЕСТВА ПРЕДПРИЯТИЙ

Требования к квалификации аудиторов по сертификации производства и системам качества программных продуктов

В стандарте **ISO 19011** сформулированы общие требования и рекомендации к *оценке компетентности аудиторов* производства и менеджмента систем качества продукции, которые целесообразно применять при сертификации программных средств. В них входят личные качества и способность применять знания и навыки при сертификационных испытаниях процессов производства – рис. 6.1 [4,7].

- Требования к квалификации аудиторов по сертификации производства и систем менеджмента качества предприятия:**
- общие требования к личным характеристикам, к компетентности, знаниям и навыкам аудиторов производства и систем качества программных средств;
 - требования к квалификации аудиторов по сертификации производства программных средств:
 - понимание принципов работы проверяемого предприятия и производства программных средств:
 - факторы, необходимые при выборе главных аудиторов и аудиторов для выполнения конкретных заданий сертификации производства программных средств;
 - обязанности главного аудитора качества производства программных средств;
 - обязанности аудиторов качества производства программных средств.

Рис. 6.1

Для того, чтобы аудиторы имели возможность выбора и систематического проведения аудита надлежащим образом, они должны быть готовы к **выполнению следующих действий**:

- применению принципов, процедур и методов аудита;
- результативному планированию и организации работ;
- проведению аудита в течение установленного срока;
- сбору данных посредством результативного опроса, выслушивания, наблюдений и анализа документов, записей и данных;
- проверке точности собранных данных;
- подтверждению достаточности и приемлемости свидетельств аудита для подкрепления выводов аудита и заключений;
- оценке факторов, влияющих на достоверность выводов и заключений по результатам аудита;
- подготовке отчетов по аудиту;
- сохранению конфиденциальности данных.

Для применения **критериев аудита**, знания и навыки в этой области аудиторов должны охватывать [7]:

- стандарты по системе менеджмента качества, применяемые процедуры или другие документы по системам менеджмента, используемые в качестве критериев аудита;
- применение ссылочных документов к различным ситуациям при аудите;
- системы информации и методы санкционирования доступа, обеспечения безопасности, рассылки и управления документами, данными и записями.

Для понимания **принципов работы проверяемого предприятия и производства** знания и навыки аудиторов в этой области должны охватывать:

- размеры, структуру, функции предприятия и взаимосвязи подразделений внутри него;
- общие бизнес-процессы производства продукта и соответствующую терминологию;
- культурные и социальные обычаи проверяемого предприятия;
- применяемые законы, технические регламенты и другие требования, относящиеся к предмету производства;
- контракты и договоры предприятия;

- международные соглашения и конвенции.

Руководители аудиторских групп должны иметь дополнительные знания и навыки по руководству аудитом для результативного и эффективного его проведения. Руководитель аудиторской группы должен быть подготовлен к выполнению следующих действий:

- планированию аудита и результативное использование ресурсов в ходе аудита;
- представлению аудиторской группы при взаимодействии с заказчиком аудита и проверяемой организацией;
- организации и направления работы членов аудиторской группы;
- руководству аудиторской группой для получения заключения по результатам аудита;
- предупреждению и разрешению конфликтов;
- подготовке и завершению отчета (акта) по аудиту.

Аудиторы производства и системы менеджмента качества должны обладать знаниями и навыками в следующих областях:

- принципы менеджмента качества программных средств и их применение,
- терминологию в отрасли производства программных продуктов;
- технические характеристики процессов и продукции производства;
- процессы и практику производства программных средств и в области экономики.

У аудиторов должны быть необходимое образование, опыт работы, обучения аудиту и опыт его проведения. Практический опыт работы должен быть в технической сфере, сфере управления или профессиональной области, включая опыт принятия решений, разрешения проблем и обмена информацией с другим управленческим или специальным персоналом, сотрудниками того же уровня, потребителями и/или другими заинтересованными сторонами.

При выборе главных аудиторов и аудиторов для выполнения конкретных заданий по сертификации производства программных средств необходимо, чтобы уровень их подготовки соответствовал следующим требованиям (см. рис. 6.1):

- знание состава и содержания профиля стандартов системы качества программных средств, в соответствии, с которыми должны

быть проведены испытания;

- наличие профессиональных квалификационных оценок или технической экспертизы аудиторов в определенной прикладной области;
- достаточный размер и состав аудиторской группы;
- навыки управления группой аудиторов;
- личные качества, необходимые для общения с проверяемыми производствами и специалистами;
- отсутствие реального или предполагаемого столкновения интересов.

Службы управления Программами сертификации производства должна постоянно оценивать деятельность своих аудиторов и экспертов либо путем наблюдения за проведением испытаний, либо другими средствами. Получаемая информация должна использоваться для улучшения выбора аудиторов и их деятельности и для выявления их неправильных действий. Проверки, проводимые разными аудиторами, должны приводить к аналогичным выводам, когда одна и та же процедура проверяется в одинаковых условиях. Должны быть установлены **методы для оценки и сравнения деятельности аудиторов** с целью достижения согласованности в их деятельности: семинары по обучению; сравнения деятельности аудиторов; ротация аудиторов из одной группы в другую.

Главный аудитор является, в конечном счете, ответственным за проведение всех этапов сертификационных испытаний. Следует, чтобы он обладал способностями и опытом в управленческой деятельности и имел достаточную компетенцию для принятия решений, связанных с проведением проверок и любых наблюдений. Кроме того, **в обязанности главного аудитора входит:**

- участие в выборе других членов группы для сертификации;
- составление Программы проведения проверок, подготовка рабочих документов и инструктирование бригады по сертификационным испытаниям процессов и/или продуктов;
- определение требований к деятельности, связанной с сертификацией включая квалификационные требования к аудиторами;
- оценка документов, относящихся к деятельности в области существующей системы качества с целью установления их адекватности производству;
- представление группы по испытаниям руководству со сторо-

ны проверяемого производства;

- информирование проверяемого производственного коллектива о наличии значительных несоответствий и о существенных затруднениях в ходе проверок системы качества;

- в четкой и ясной форме предоставление отчетов о результатах сертификационных испытаний.

Необходимо, чтобы аудиторы сохраняли беспристрастность и не были подвержены влияниям, которые могли бы неблагоприятно отразиться на их объективности. Следует, чтобы все лица и все организации, занимающиеся проверкой, уважали независимость и честное отношение к делу аудиторов, а также поддерживали их.

В обязанности аудиторов входит:

- передача и разъяснение проверяемым производственным коллективам требований к испытаниям;

- планирование и эффективное выполнение возложенных на них обязанностей;

- запись результатов наблюдений и подготовка отчетов о результатах испытаний;

- контроль за эффективностью выполнения корректирующих действий после проведения проверок;

- обеспечение сохранности документации, относящейся к испытаниям для ее предоставления по требованию, обеспечение ее конфиденциальности, обработка секретной информации с соблюдением требований инструкций;

- сотрудничество с главным аудитором и оказание помощи в его деятельности.

Необходимо, чтобы аудиторы: зарекомендовали себя объективными; собирали и анализировали достоверные и достаточно полные данные для подготовки выводов, относящихся к проверяемой системе качества; могли бы установить, что процедуры, документы и другая информация, описывающая или обеспечивающая требуемые компоненты системы качества, являются известными, понятными, имеющимися в распоряжении и используемыми коллективом проверяемого предприятия.

Управление программами сертификационных испытаний должно осуществляться главными аудиторами, обладающими практическими знаниями в области методов проверки конкретных систем качества производства программных средств. Служба управления сер-

тификацией должна нанимать аудиторов, квалификация которых соответствуют рекомендациям стандарта **ISO 19011**. Эти аудиторы должны быть утверждены оценочной комиссией (органом сертификации), приемлемой с точки зрения службы управления программами испытаний и соответствующей рекомендациям, представленным в этом стандарте.

Подготовка и документирование организации процессов сертификации производства и системы качества предприятия

Наличие и полнота *технологической документации* при сертификации проверяются на соответствие требованиям действующих национальных и отраслевых стандартов, руководящих и других документов, устанавливающих требования к содержанию и составу конструкторской и технологической документации данного предприятия. При проверке учитываются *общие требования нормативных документов* на продукцию сертифицируемого производства (рис. 6.2). В составе организационно – распорядительных документов, связанных с функционированием производства и системой менеджмента качества предприятия (СМКП), проверяются [4]:

- положения о подразделениях и должностные инструкции специалистов;
- руководство по менеджменту качества производства;
- стандарты предприятия, инструкции и другие документы, связанные с реализацией процессов производства и СМКП;
- стандарты предприятия, инструкции и другие документы, связанные с выполнением специальных требований к производству по обеспечению безопасности продукции.

Положение о подразделениях и должностные инструкции должны содержать сведения, касающиеся функций подразделений, обязанностей и полномочий специалистов, связанных с реализацией процессов СМКП. Руководство по менеджменту качества производства должно содержать:

- краткую характеристику предприятия;
- перечень процессов производства и СМКП предприятия;
- обоснование исключений из области применения СМКП;
- описание процессов СМКП и их взаимодействия или ссылку на соответствующие документы предприятия;

- порядок оценки результативности процессов СМКП или ссылку на соответствующий документ предприятия.

***Подготовка и документирование организации
процессов сертификации производства и системы
менеджмента качества предприятия:***

- общие требования нормативных документов на продукцию сертифицируемого производства;
- положение о подразделениях и должностные инструкции, обязанности и полномочия специалистов, связанных с реализацией процессов производства;
- особенности и состав документации процессов и результатов сертификации – аудита технологии и системы качества производства программных средств:
 - заявка на проведение сертификации производства программных средств;
 - регистрационная карта сертифицируемого производства программных средств;
 - заключение по результатам рассмотрения заявки на сертификацию;
 - задание на проведение сертификационных испытаний;
 - план сертификации производства программного средства;
 - заключение по результатам сертификационных испытаний производства программного средства;
- документы сертификационной лаборатории – Программа испытаний и методики аудита – испытаний производства должны содержать:
 - определение объекта сертификационных испытаний;
 - цель испытаний – аудита;
 - требования к Программе сертификационных испытаний;
 - требования к документации испытаний;
 - средства автоматизации и порядок испытаний;
 - методики испытаний по каждому разделу требований производства;
 - формы отчетности выполнения и результатов испытаний;
- отчет о проверках организации сертификационных испытаний – аудита производства и системы качества программного продукта;
- передача главным аудитором отчета о проверках и организации сертификационных испытаний производства и системы качества программного продукта клиенту – заявителю.

Рис. 6.2

Орган по сертификации должен проверить и оценить идентифицированные предприятием процессы СМКП и представленные объективные свидетельства их результативности.

Как отмечалось в ведении сертификационные испытания процессов производства комплексов программ и их результатов – программных продуктов имеют одну и ту же цель удостоверение качества продукции, но **принципиальные различия** в сертификационных испытаниях процессов производства и сертификационных испытаниях качества конечного программного продукта, которые непосредственно отражаются на свойствах продукции, интересующей заказчика и потребителей. При испытаниях производственных процессов и систем качества критерии их оценки трудно или даже невозможно описать точными числовыми метриками. Большинство характеристик реальных систем качества отражается фактами наличия и использования в них некоторых свойств и реализации определенных требований стандартов Системы сертификации. Поэтому в ряде случаев целесообразно применять **бальные оценки** наличия в оценках процессов производства и в системах качества таких свойств и факторов выполнения установленных требований, а также обобщение совокупностей подобных бальных оценок.

Метод бальной оценки систем качества и состав анализируемых основных характеристик должен быть согласован с органом сертификации **в определенной проблемной области**. Предварительно должны быть определены и согласованы между заказчиком и поставщиком допустимые уровни таких суммарных бальных оценок для признания процессов производства предприятия пригодными для применения. В составе критериев сертификации производства можно выделить следующие группы показателей:

- общая оценка состояния производства и системы качества программного продукта;
- полнота применения методов поэтапного измерения качества в жизненном цикле продукции;
- уровень средств автоматизации для обеспечения качества продукции;
- полнота и состав комплекта нормативной документации, поддерживающей применение системы качества производства;
- эффективность и результаты эксплуатации системы качества производства.

В каждой из выделенных групп можно определить 5 – 10 характеристик, для более детальной оценки их наличия или отсутствия в данной группе, и для последующей интегральной численной оценки системы качества. Органом по сертификации совместно с заказчиком должны быть утверждены уровни интегральных оценок, достаточные для допуска систем качества к эксплуатации на определенных предприятиях.

В [2,11] определены *состав и содержание документов, организующих процессы взаимодействия* заявителя (поставщика или заказчика) и сертификационных органов и лабораторий при подготовке и проведении испытаний наиболее сложных объектов и процессов (см. рис. 6.2). В более простых ситуациях эти документы целесообразно адаптировать с учетом особенностей объектов и условий испытаний в основном в направлении сокращения их состава и объема.

Документы заявителя. Исходный документ – *Заявка на проведение сертификации* – должен содержать следующую информацию:

- координаты заявителя;
- представителя заявителя, ответственного за связь с органом по сертификации;
- наименование органа по сертификации;
- виды сертификационных испытаний;
- состав и наименования нормативно – технических документов, на соответствие которым должны быть проведены сертификационные испытания;
- обязательства заявителя об оплате рассмотрения заявки и проведения сертификации;
- обязательства заявителя по выпуску сертифицированной продукции в соответствии с образцом, прошедшим сертификацию;
- личные подписи заявителя и его представителя.

Предъявляемые на сертификацию программное средство, его компоненты или система качества предприятия должны представляться в комплекте с соответствующей документацией. В случае различной комплектации поставки пользователю предъявляемого объекта, заявитель должен в «Договоре на проведение сертификации» оговорить условия и варианты комплектности поставки на сертификацию.

Документы органа (лаборатории) по сертификации (см. рис. 6.2). Внутренний документ – **Регистрационная карта** сертифицируемого производства или системы качества предназначен для организации учета и хранения оперативной и архивной информации о сертифицируемой продукции, ее компонентах, процессах и результатах их сертификации, и должен содержать следующие виды информации:

- служебную: регистрационные номер заявки, классификационные признаки производства или продукции;
- исходные данные о заявителе, организация, реквизиты заявки и Договора на проведение сертификации;
- исходную о сертифицируемых ПС или системе качества: условное обозначение, наименование, краткие характеристики;
- исходную о видах испытаний и составе соответствующей нормативной документации;
- исходную об испытательной лаборатории, проводящей сертификационные испытания;
- выходную о результатах испытаний: реквизиты протоколов испытаний и краткие выводы по результатам испытаний;
- выходную о выданном сертификате: реквизиты сертификата и срок действия.

Документ органа по сертификации **Заключение по результатам рассмотрения заявки** должен содержать аргументированное решение о принятии предъявленной продукции или системы качества для проведения сертификации и информацию об условиях проведения сертификации (договорно – правовые и финансовые условия). В случае отказа в принятии на сертификацию данный документ должен содержать мотивированное обоснование решения.

Исходный документ **Задание на проведение сертификационных испытаний** является основанием для проведения сертификации предъявленной продукции или системы качества, и должно содержать разделы:

- основание для проведения работ;
- требования к сертификационным испытаниям предъявленного объекта или производства;
- требования к оформлению результатов испытаний.

В разделе «Основание для проведения работ» перечисляют документы, на основании которых проводятся сертификационные ис-

пытания (реквизиты Заявки на проведение сертификации и Договора на проведение сертификации). В разделе «Требования к сертификационным испытаниям» должны быть указаны виды сертификационных испытаний, обозначение и наименование нормативной документации, на соответствие которым проводятся испытания, и состав требований, подлежащих проверке. В разделе «Требования к оформлению результатов испытаний» должны быть представлены виды и комплектность документов, предъявляемых испытательной лабораторией в орган по сертификации по результатам проведенных испытаний.

План сертификации программного средства или его производства предназначен для использования сертифицирующей организацией с целью определения, что предлагаемый заявителем жизненный цикл ПС и/или его компонентов соответствует требованиям контракта, технического задания и спецификаций заказчика. Этот план в соответствии со стандартами должен включать:

- обзор системы, включающий описание ее функций и их распределения между аппаратными и программными средствами, архитектуру, используемые процессоры, аппаратно-программные интерфейсы и особенности обеспечения безопасности;
- обзор программного средства: функции ПС, концепция формирования структуры, обеспечение безопасности и резервирование, стратегия синхронизации и планирования исполнения ПС и компонентов;
- задачи сертификации: описание сертификационного базиса, доказательства соответствия результатов разработки требованиям стандартов по сертификации, процесс оценки безопасности системы;
- жизненный цикл программного средства: используемая модель жизненного цикла, которая будет контролироваться в соответствующих детальных планах, а также ответственность за процессы ЖЦ системы и процессы сертификационного взаимодействия;
- документы жизненного цикла ПС, которые должны быть разработаны и контролироваться процессами жизненного цикла ПС, отношения между документами, представляемыми на рассмотрение сертифицирующей организации, формы документов и способы, посредством которых документы ЖЦ становятся доступными для сертифицирующей организации;

- план – график: раздел описывает средства заявителя, которые будут использоваться для обеспечения прозрачности работ для сертифицирующей организации по процессам жизненного цикла программного средства с целью планирования просмотров.

В разделе – «Средства и методы испытаний» – должны быть указаны технические и программные инструментальные средства, используемые во время испытаний, и порядок проведения испытаний. В разделе – «Методики испытаний» приводятся описания проверок с указанием ожидаемых результатов испытаний. Они должны содержать разделы: объекты и цели испытаний; оцениваемые показатели качества; условия и порядок испытаний; методы обработки, анализа и оценки результатов испытаний; средства автоматизации испытаний; отчетность.

Документ *Регистрационная карта испытаний* – предназначен для организации учета и хранения оперативной и архивной информации о процессах и результатах испытаний и включает следующие виды информации: служебную (регистрационный номер, классификационные признаки и т. д.); исходную об испытуемом объекте; исходную о видах испытаний; выходную о результатах испытаний.

Основными выходными документами испытательной лаборатории являются *Протоколы испытаний*, предназначенные для оформления окончательных результатов сертификационных испытаний и выдачи «Сертификата соответствия» уполномоченным органом. Каждый протокол испытаний должен содержать информацию: служебную (регистрационный номер, дату, реквизиты утверждения, сведения о сертификационной лаборатории); исходную (краткие сведения об испытанном объекте и заявителе, основание для проведения испытаний); выходную (краткие сведения о проведенных испытаниях, включая сведения об испытаниях структурных компонентов комплекса программ) с указанием реквизитов протоколов испытаний функциональных компонентов; сведения о нормативной документации, на соответствие которым проверялись продукция или процессы; итоговую (краткие сведения о результатах испытаний, выводы и предложения).

Документ *Отчет о проведении испытаний* предназначен для подробного описания результатов проведенных испытаний. Основная часть отчета содержит сведения о видах проведенных испытаний и их результатах для отдельных структурных и функциональных компо-

нентов системы качества производства в целом с подробным описанием полученных результатов.

Документ *Заключение по результатам сертификационных испытаний* разрабатывается экспертами органа и содержит обобщенные сведения о результатах испытаний и обоснование целесообразности или невозможности выдачи сертификата, включая рекомендации по доработке комплекса программ или системы качества. Основным выходным документом органа по сертификации является *Сертификат соответствия*, предназначенный для документального удостоверения соответствия предъявленных заявителем продуктов или системы качества производства установленным требованиям. Форма и правила заполнения сертификата соответствия должны отвечать требованиям, установленным национальным органом по сертификации.

Документы сертификационной лаборатории (см. рис. 6.2) следует определять и адаптировать в зависимости от специфики конкретного производства и/или систем качества и видов проводимых испытаний. Обязательным является наличие нормативных документов и требований, на соответствие которым согласно заявке проводится сертификация. Документы *Программа испытаний и Методики испытаний* должны содержать:

- определение объекта испытаний;
- цель испытаний;
- требования к Программе испытаний;
- требования к документации;
- средства автоматизации и порядок испытаний;
- методики испытаний по каждому разделу требований;
- формы отчетности.

В разделе «Объект испытаний» должны быть указаны обозначение, наименование, область применения и состав испытываемого производства или системы качества. Состав испытываемого объекта и цель проведения испытаний должны указываться с точностью до отдельных структурных компонентов. «Требования к документации» должны отражать состав эксплуатационной документации, представляемой на испытания, и предъявляемые к ней требования по содержанию. Для сертификации систем качества, кроме адаптированных базовых стандартов серии **ISO 9000** и **ISO 12207**, заявителем должны предъявляться рабочие версии Программы качества предприятия и

Руководств по качеству и управлению конфигурацией на предприятии.

Организация сертификационных испытаний производства и систем качества программных средств

Предприятие, имеющее потребность в проведении сертификационных испытаний производства и системы качества (заявитель), должно создать *условия и представить документы для обеспечения процессов испытаний*. Служба управления Программами проверок должна определить те разделы требований стандартов, на соответствие которым предполагается проверять производство и систему качества, и разработать инструкции, позволяющие эффективно производить такие испытания. После получения и проверки комплектности и качества документации специалистами испытательной лаборатории следует провести экспертизу степени реального применения системы качества на предприятии.

Испытания начинаются с составления *Программы проверки производства и системы качества предприятия*, которая должна служить рабочим планом проведения последующих работ. Программа является внутренним рабочим документом испытательной лаборатории и должна содержать перечень работ, детализируемый в соответствии со спецификой предприятия, и включающий в себя анализ полноты и качества представленных исходных документов и степени их практической реализации при производстве и поставке программных средств. Экспертиза применения процедур системы качества осуществляется испытательной лабораторией на рабочих местах предприятия, обеспечивающего ЖЦ ПС. Проверки проводятся по наличию на рабочих местах специалистов-разработчиков соответствующих документов и по полноте использования их положений и рекомендаций. Анализы состояния производства и внутренние проверки системы качества, процессов и/или продукции должны проводиться *персоналом – аудиторами, независимыми* от лиц, непосредственно ответственных за выполнение этих работ.

При составлении Программы испытаний аудиторам следует рассмотреть адекватность методов и документов, представленных проверяемым предприятием, требованиям системы качества (Руководству по качеству или аналогичным документами). Если исследование обнаружит несоответствие требованиям системы, описанной проверяемым, к дальнейшим действиям приступать не следует без решения

возникших вопросов в интересах клиента, инспекторов и, если возможно, проверяемого предприятия.

Сертификация производства и систем качества *на месте* предприятия рекомендуется [2, 4] из *следующих этапов* – рис. 6.3.

- организация работ;
- заочная, предварительная оценка качества производства;
- подготовка к аудиту – испытаниям качества на месте производства;
- проведение аудита – испытаний качества на месте производства и подготовка акта по результатам аудита – испытаний;
- завершение сертификации качества производства, выдача и регистрация сертификата;
- инспекционный контроль сертифицированного производства.

Организация работ – основанием для начала работ может служить письмо – обращение, направленное заказчиком в произвольной форме в орган по сертификации. Орган по сертификации проводит предварительную регистрацию письма – обращения (заявки) и его анализ для определения возможности проведения сертификации с учетом:

- оценки соответствия области деятельности заказчика и области аккредитации органа по сертификации;
- наличия у органа по сертификации необходимой информации для планирования аудита – испытаний;
- наличия у органа по сертификации возможности проведения работ в сроки, предпочтительные для заказчика, и наличия соответствующих ресурсов.

При положительном решении орган по сертификации регистрирует письмо – обращение (заявку) и извещает об этом заказчика. В случае положительного решения о принятии заявки на сертификацию производства орган по сертификации и заказчик заключают договор на проведение сертификации производства.

После представления заказчиком оформленной заявки, запрошенных сведений и документов, распоряжением руководства органа по сертификации назначают руководителя группы аудита (председателя комиссии) и формируют группу аудита (комиссию по сертификации).

Организация сертификационных испытаний производства и систем менеджмента качества программных средств:

- условия и документы для обеспечения процессов сертификационных испытаний программных средств;
- Программа и методики испытаний – аудита производства и системы менеджмента качества предприятия;
- организация работ по сертификации производства и системы менеджмента качества программных средств;
- первоначальное заседание комиссии по сертификации;
- заочная оценка качества производства программного средства;
- подготовка к аудиту – испытаниям на месте производства программных средств;
- одобрение Программы и методик испытаний и передача ее аудиторам;
- проведение аудита – испытаний на месте производства и подготовка акта по результатам аудита предприятия и качества производства программных средств:
 - источники информации о состоянии производства;
 - протоколы и отчет под руководством главного аудитора о результатах испытаний качества производства;
 - критерии оценки качества производства на соответствие требованиям комиссии auditors;
 - результаты аудита, выводы и рекомендации комиссии, оформленные в виде акта;
 - расширение или сужение области сертификации производства;
- завершение сертификации, выдача и регистрация сертификата по результатам аудита – испытаний производства программных средств;
- инспекционный контроль состояния сертифицированного производства программных средств.

Рис. 6.3

Целью первоначального заседания комиссии по сертификации является:

- представление членов группы аудиторов высшему руководству проверяемого производства;
- рассмотрение задач и объема проводимых испытаний;

- представление краткого описания методов и процедур, которые будут использоваться для проведения испытаний;
- определение официальных взаимоотношений между группой аудиторов и проверяемым предприятием;
- предоставление в распоряжение группы аудиторов необходимых средств и оборудования;
- определение времени и даты проведения заключительного и всех промежуточных собраний группы по испытаниям совместно с высшим руководством проверяемого предприятия;
- разъяснение любых непонятных пунктов Программы испытаний.

Для **сбора данных** используются интервью, изучение документов, наблюдение за деятельностью и обстановкой в рассматриваемых областях системы качества и производства. Выявляются факты несоответствия, если они кажутся важными, даже в том случае, если они не оговорены в контрольном перечне, и проводится их анализ.

В ходе испытаний главный аудитор может вносить изменения в распределение обязанностей между аудиторами и в Программу проверок с одобрения клиента и согласия проверяемого предприятия, если это является необходимым для оптимального достижения целей проверок. Следует все **результаты наблюдений**, полученные в ходе испытаний, подтверждать документами. Выявленные отклонения следует идентифицировать в соответствии со специальными требованиями стандартов или других документов, на которые можно сделать ссылку в процессе проверки. Главный аудитор должен проанализировать полученные результаты наблюдений вместе с ответственным руководителем проверяемого предприятия. Все результаты о несоответствиях должны быть подтверждены руководством проверяемого производства.

Заочную оценку производства – выполняет назначенная комиссия по сертификации до выезда на предприятие и включает: проверку области сертификации производства и области применения системы менеджмента качества предприятия (СМКП); оценку соответствия качества продукции требованиям заказчика, потребителей и обязательным требованиям; проверку документов СМКП. Заочная оценка завершается оформлением письменного заключения, в котором наряду с выявленными замечаниями формулируют вывод о возможности

или невозможности аудита системы менеджмента качества производства и на месте предприятия.

Подготовка к аудиту на месте производства включает предварительное взаимодействие с проверяемой организацией, которое проводит председатель комиссии с целью согласования порядка доступа к соответствующим документам, определения представителей проверяемой организации (лиц, сопровождающих экспертов). Председатель комиссии подготавливает план испытаний – аудита, включающий:

- цели и объемы аудита;
- сроки и график проведения аудита;
- область аудита, включая идентификацию структурных подразделений проверяемого предприятия;
- проверяемые подразделения и процессы СМКП;
- идентификацию представителей проверяемой организации (лиц, сопровождающих экспертов);
- требования конфиденциальности.

План аудита должен быть доведен до сведения проверяемого предприятия до начала аудита на месте. Если комиссия состоит из нескольких экспертов, председатель комиссии, руководствуясь планом аудита, по согласованию с членами комиссии распределяет между ними обязанности по аудиту конкретных подразделений, видов деятельности, процессов и процедур СМКП проверяемой организации. При распределении обязанностей учитывают необходимость соответствия компетентности экспертов и технических экспертов проверяемым видам деятельности организации согласно плану аудита.

Клиенту – заявителю необходимо **одобрить Программу и методики испытаний** и передать ее аудиторам и проверяемому предприятию. Следует составить гибкую Программу проверки, способную учитывать изменения, основанные на информации, получаемой в ходе проверок, и обеспечивать эффективное использование инструментальных средств, которая включает:

- идентификацию лиц предприятия, несущих прямую ответственность за выполнение поставленных целей и объем испытаний;
- идентификацию нормативных ссылок (таких как стандарты на используемую систему качества, Руководство по системе качества проверяемой организации);
- идентификацию членов группы по испытаниям;

- дату и место проведения испытаний;
- идентификацию организационных подразделений предприятия, подлежащих проверкам;
- дату и продолжительность, предусмотренные для проведения каждого этапа испытаний;
- календарный план встреч с руководством проверяемого предприятия;
- правила распространения отчета об испытаниях и предусмотренная дата публикации результатов.

Если руководитель проверяемого производства имеет возражения, относящиеся к пунктам Программы, следует сообщить об этом главному аудитору. Эти возражения необходимо разрешать между главным аудитором и проверяемым руководителем производства и, в случае необходимости, клиентом – заявителем перед проведением проверки. Специальные подробности Программы испытаний должны быть доведены до сведений проверяемого в ходе проведения испытаний только в том случае, если их преждевременное раскрытие не будет препятствовать получению объективных данных.

Каждому аудитору следует поручать проверку определенных элементов системы качества или функциональных участков производства. Такое распределение обязанностей должен проводить главный аудитор, консультируясь с заинтересованными аудиторами. Документы, необходимые для облегчения исследований аудиторов и записи заключительных результатов, могут включать:

- контрольный перечень для оценки каждого элемента системы качества;
- формы для составления отчетов результатов испытаний;
- формы для записи объективных данных, подтверждающих заключительные выводы аудиторов.

Рабочие документы следует составлять таким образом, чтобы не ограничивать деятельность или дополнительные исследования аудиторов, которые могут оказаться необходимыми в процессе получения информации. Рабочие документы, содержащие конфиденциальную или производственную информацию, являющуюся собственностью фирмы, должны быть надлежащим образом защищены организацией, проводящей испытания.

Проведение аудита на месте и подготовка акта по результатам аудита начинается с предварительного совещания, которое

проводится под руководством председателя комиссии с участием членов комиссии, руководства и ведущих специалистов проверяемого предприятия. Целью **предварительного совещания** является:

- подтверждение возможности реализации плана аудита на месте;
- краткое изложение используемых методов и процедур аудита;
- установление официальных процедур взаимодействия между членами комиссии и сотрудниками проверяемой организации.

В ходе аудита председатель комиссии периодически информирует проверяемую организацию о ходе аудита. Информацию, полученную в ходе аудита и свидетельствующую о наличии непосредственного риска нарушений требований к качеству продукции и нарушения требований к производственным процессам, немедленно доводят до сведения руководства проверяемого предприятия. В качестве **источников информации** используют:

- интервью с работниками проверяемой организации;
- собственные наблюдения экспертов за деятельностью персонала и функционированием процессов производства;
- данные обратной связи от заказчика и потребителей;
- документы системы менеджмента качества производства регламентирующего характера, такие как руководство по менеджменту качества производства, стандарты (процедуры) предприятия, регламенты, положения, инструкции;
- данные обзоров, анализов функционирования системы менеджмента качества.

Полученная и проверенная информация по объектам аудита или свидетельства аудита должна быть **сопоставлена с критериями** аудита для получения выводов, которые могут **указывать на соответствие или несоответствие** СМКП критериям аудита. Свидетельства должны быть обобщены с указанием мест наблюдений, функций, процессов и требований, которые были проверены. Несоответствия и подтверждающие их свидетельства аудита должны быть зарегистрированы.

В ходе аудита производства все обнаруженные отклонения объектов аудита от требований **ISO 9001** и документов предприятия должны быть тщательно рассмотрены и классифицированы комиссией в зависимости от степени несоответствия рассматриваемого объекта проверки. Наблюдения, сделанные в ходе аудита, классифицируют

с целью выполнения проверяемой организацией корректирующих действий (для устранения причин несоответствий), адекватных последствиям выявленных несоответствий, принятия органом по сертификации решения о выдаче, подтверждении, приостановлении или отмене действия сертификата, а также расширения или сужения области сертификации. **При наличии несоответствий:**

- на заключительном совещании комиссия официально представляет руководству проверяемого предприятия зарегистрированные несоответствия;
- предприятие проводит анализ причин несоответствий и разрабатывает план корректирующих мероприятий с представлением его в орган по сертификации.

Главный аудитор должен представлять результаты испытаний с учетом той значимости, которую они имеют, по его мнению, а также заключения группы с учетом эффективности системы качества, что гарантирует их соответствие целям по обеспечению качества.

Отчет об испытаниях готовится под руководством главного аудитора, ответственного за его точность и полноту. Отчет должен точно отражать характер и содержание проверок, датирован и подписан главным аудитором. Следует, чтобы отчет, в зависимости от применения, содержал:

- объем и задачи проверок;
- подробности Программы проверок, идентификацию членов группы по проверкам и проверяемого, даты проведения проверок и идентификацию проверяемой организации;
- идентификацию нормативных документов, в соответствии с которыми проводились проверки;
- результаты наблюдений о несоответствиях;
- оценку группой по проверке степени соответствия системы качества стандартам и документации;
- оценку способности системы и руководства предприятия решать задачи по обеспечению качества;
- перечень рассылок отчета о проверке производства;
- акт о завершении сертификационных испытаний производства и системы качества.

Выполнение запланированных корректирующих действий комиссия проверяет с выездом или без выезда на предприятие. Председатель комиссии несет ответственность за подготовку и содержание

акта по результатам аудита. До проведения заключительного совещания комиссия анализирует выявленные несоответствия и оформляет акт по результатам аудита с указанием несоответствий, их категории и рекомендациями органу по сертификации для принятия решения о выдаче (невыдаче) сертификата соответствия производства. **Результаты аудита**, выводы и рекомендации комиссия оформляет в виде **акта**, который должен содержать:

- идентификацию органа по сертификации;
- идентификацию организации – заказчика;
- цель и область аудита;
- основание для проведения аудита;
- время и место проведения аудита;
- состав комиссии по сертификации с идентификацией председателя и членов комиссии;
- идентификацию нормативной базы аудита;
- результаты аудита;
- выводы комиссии;
- адреса рассылки акта.

Отчет и проект акта об испытаниях **передается клиенту – заявителю** главным аудитором. На клиента возлагается обязанность предоставить копию отчета о проверках высшему руководству проверяемого предприятия. Следует определить и согласовать с проверяемым предприятием любую дополнительную рассылку отчета. Отчеты о проверках, содержащие конфиденциальную или производственную информацию, являющуюся собственностью фирмы, должны быть соответствующим образом защищены организацией, проводящей проверку, и клиентом.

Документы по испытаниям следует сохранять в соответствии с соглашением между клиентом – заявителем, проверяющей организацией и проверяемым предприятием, соблюдая установленные требования. Проверка считается законченной, когда отчет о ней (возможно **сертификат соответствия**) предоставляется клиенту – заявителю. Проверяемый несет ответственность за определение и инициирование любых корректирующих действий, необходимых для исправления несоответствия или устранения причины, которая может его вызвать.

Заключительное совещание аудиторов проводится под руководством председателя комиссии с целью представления выводов и

заключений по аудиту СМКП. К совещанию должен быть подготовлен проект акта по результатам аудита. На заключительном совещании должны присутствовать руководство и ведущие специалисты проверяемого предприятия, заказчика и все члены комиссии. По окончании проверок и перед подготовкой отчета следует, чтобы группа по сертификации провела **совещание вместе с высшим руководством** проверяемого предприятия и лицами, ответственными за выполнение рассматриваемых действий. Основной целью этого заседания является представление результатов наблюдений, полученных в ходе испытаний, высшему руководству проверяемого предприятия в виде ясно сформулированных результатов.

Акт подписывают председатель комиссии, члены комиссии и представляют для ознакомления и подписи руководителю проверяемого предприятия или его представителю. Экземпляры акта являются собственностью проверяемой организации и органа по сертификации, при этом члены комиссии и проверяемая организация должны строго соблюдать требования конфиденциальности.

Завершение сертификации, выдача и регистрация сертификата считаются выполненными, если проведены все работы, предусмотренные планом аудита, и акт по результатам аудита подписан сторонами и разослан. Документы по сертификации производства хранят в органе по сертификации в соответствии с правилами, установленными в соответствующих документах органа по сертификации. Комиссия и руководство органа по сертификации не должны раскрывать содержание документов и другой информации, полученной во время аудита, а также актов по результатам аудита любой другой стороне без согласия проверяемой организации и заказчика.

Критерием принятия решения о соответствии производства и СМКП, установленным требованиям является отсутствие в акте аудита сведений о несоответствиях или выполнение проверяемым предприятием корректирующих мероприятий по устранению несоответствий в согласованные сроки. При этом орган по сертификации должен признать корректирующие мероприятия результативными. Контроль выполнения корректирующих действий по установленным несоответствиям орган по сертификации систем качества планирует и осуществляет после получения письменного отчета проверяемой организации об устранении несоответствий.

При положительном решении орган по сертификации оформляет **сертификат соответствия производства** установленного образца. В органе по сертификации, на сертификате проставляют регистрационный номер, затем сертификат регистрируют в реестре органа по сертификации. Обычно **срок** действия сертификата соответствия производства – **три года**.

Инспекционный контроль качества сертифицированного производства может быть плановым и внеплановым. При плановом инспекционном контроле общий объем проверки должен составлять не менее 1/3 количества процессов СМКП, включая совокупность обязательных элементов, проверяемых при каждом инспекционном контроле. Объекты аудита при внеплановом инспекционном контроле определяют в зависимости от причины, вызвавшей необходимость инспекционного контроля.

Расширение или сужение области сертификации производства проводят при изменении номенклатуры выпускаемой продукции или изменении условий производства имеющейся номенклатуры продукции.

Состав и содержание документации производства и системы качества предприятия зависят от характеристик процессов проектирования, разработки и модификации программных средств, а также от требований к их качеству и особенностей технологической среды. Поэтому необходимый комплект документов для каждого предприятия или проекта следует выбирать и адаптировать применительно к этим характеристикам. Оцениваемыми при сертификации показателями системы качества являются наличие соответствующих документов и практическое выполнение требований международных стандартов серии **ISO 9000** и должностных инструкций специалистами предприятия – разработчика. Клиент – заявитель должен подготовить и предъявить проверяющей испытательной лаборатории, согласованный между заказчиком и разработчиком и утвержденный, комплект документов для проверки их достоверности, достаточности состава и качества процессов изготовления в соответствии с нормативными документами.

Ориентировочный комплект основных документов сертификации качества технологии и производственных процессов состоит из трех групп, содержание которых по структуре подобен, но

несколько отличается от комплекта документов при **сертификации программных продуктов** (см. лекцию 11):

- базовые нормативные документы системы качества в соответствии с номенклатурой и содержанием серии стандартов **ISO 9000**, а также подготовленные руководителями предприятия на их основе Программа, Руководство и инструкции, предъявляемые испытателям – сертифицираторам технологии и системы качества производства;

- исходные документы, характеризующие конкретное производственное предприятие и жизненный цикл программного средства, подготавливаемые руководством предприятия для сертификации технологии и системы качества производства;

- отчетные документы испытателей – аудиторов, отражающие результаты испытаний – сертификации технологии производства и системы качества предприятия, представляемые органу сертификации, заявителю и руководству проверяемого предприятия.

Перечень и приблизительное содержание групп этих документов могут быть ориентированы на общий случай проверки систем качества предприятий, реализующих и поддерживающих жизненный цикл сложных программных средств, базирующихся на изложенных выше международных стандартах. Комплект документов может сокращаться и адаптироваться по согласованию между заявителем, испытателями и руководством проверяемого предприятия в соответствии с характеристиками и особенностями производства программных средств.

В отчетной **технологической документации**, обеспечивающей многолетнее сопровождение, конфигурационное управление и сертификацию процессов производства программного продукта, необходимы требования, тесты, тексты программ и описания алгоритмов комплекса программ. Это приводит к большому объему документации, которая может составлять около 100 страниц совокупности документов на тысячу строк программного продукта. Статистический анализ объема документации для производства программных продуктов различных классов дал широкий разброс числа страниц на единицу объема программ [15, 17]. Однако выявились некоторые средние характеристики, которые близки к указанным оценкам. Подтверждена наиболее высокая документированность тиражируемых многоверсионных программ реального времени, особенно в тех случаях,

когда необходима высокая безопасность и надежность функционирования программного продукта (до 200 страниц на тысячу строк).

При оценках можно предполагать средний объем *технологической документации* ~ 50 – 100 страниц на тысячу строк. При этом затраты на документацию практически пропорциональны размеру комплекса программ. Часть документации не всегда подлежит массовому тиражированию и поставке каждому Пользователю, что позволяет несколько снижать удельные затраты на ее подготовку. Однако необходимость ее тщательной отработки, и высокого соответствия текущей версии программного продукта, приводит к большим затратам, как при первичном изготовлении технологических документов, так и при их модификации в процессе последующего сопровождения.

Часть 3

СЕРТИФИКАЦИЯ ПРОГРАММНЫХ ПРОДУКТОВ

Лекция 7

ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ХАРАКТЕРИСТИКАМ И КАЧЕСТВУ ПРОГРАММНЫХ ПРОДУКТОВ

Общие требования к качеству функционирования программных продуктов

Цели, методы и *схема процессов сертификации программного продукта* должны обеспечивать высокое качество результатов производства и его полное соответствие требованиям заказчика и пользователей. В учебнике они изложены в *пяти лекциях*, общий состав которых содержит (рис. 7.1):

- *формирование* функциональных и конструктивных *требований* к программным продуктам, требования к допустимым рискам и проверку их корректности;
- *организацию* сертификационных испытаний программных продуктов на соответствие требованиям, предварительный выбор стратегии, планирование и оценки затрат на испытания;
- *подготовку* сертификационных испытаний программных продуктов, выбор методов, требования к тестам и процессы генерации динамических тестов внешней среды в реальном времени;
- Программы, методики и *процессы испытаний* программных продуктов, а также эксплуатационной документации на соответствие требованиям к программным продуктам;
- *завершение* испытаний, оформление отчетной документации и утверждение акта результатов сертификационных испытаний программных продуктов на соответствие требованиям, тиражирование и внедрение сертифицированных версий программных продуктов.

Схема сертификационных испытаний программных продуктов***Формирование функциональных требований к программному продукту:***

- особенности требований заинтересованных лиц в программных продуктах;
- требования к надежности функционирования, функциональной безопасности, производительности и эффективности использования ресурсов ЭВМ программным продуктом;
- требования к допустимым рискам применения программного продукта и эксплуатационным характеристикам.

Организация сертификационных испытаний программного продукта на соответствие требованиям:

- цели, задачи и процессы сертификационных испытаний;
- стратегии, планирование, графики тестирования и ресурсы предварительных испытаний на соответствие требованиям;
- оценки затрат на испытания программного продукта.

Подготовка сертификационных испытаний программного продукта:

- требования к квалификации испытателей программного продукта;
- подготовка тестов для испытаний программного продукта;
- требования к тестам для испытаний модулей и компонентов;
- требования к генерации и компонентам динамических тестов внешней среды для испытаний продукта в реальном времени;
- средства обработки результатов и оценки эффективности динамической генерации тестов для испытаний в реальном времени.

Сертификационные испытания программного продукта на соответствие требованиям:

- порядок сертификационных испытаний программного продукта;
- программа и методики испытаний программного продукта;
- испытания соответствия функциональных и конструктивных характеристик программного продукта заданным требованиям;
- испытания надежности, функциональной безопасности и динамических характеристик программного продукта.

Удостоверение качества и завершение сертификационных испытаний продукта на соответствие требованиям:

- испытания для сокращения и ликвидации опасных рисков;
- испытания эксплуатационной документации на соответствие требованиям к программному продукту.
- завершение испытаний и оформление документации результатов, утверждение акта сертификационных испытаний;
- тиражирование и внедрение сертифицированных версий программного продукта.

Рис. 7.1

Для **сертификации программных продуктов**, прежде всего, необходимо понимать их основные свойства, уметь выделить и формализовать требуемые характеристики. Базовые международные стандарты определяют основные требования к качеству комплексов программ для их сертификации (см. Приложение). Непосредственно к характеристикам качества программных средств относится **группа стандартов**, в которую входят нормативные документы, регламентирующие формализацию требований к метрикам качества комплексов программ, методы и процессы их выбора и измерения. Эта группа является важнейшей для достижения и гарантирования высокого качества **программного продукта**. Тщательное **специфицирование требований к качеству программного продукта – ключевой фактор обеспечения их сертификации и адекватного применения**. Это может быть достигнуто на основе выделения и определения подходящих характеристик качества с учетом целей использования и функциональных задач комплекса программ [3, 9, 12].

Основой для формирования требований к характеристикам комплексов программ является анализ важнейших свойств, отражающих качество их функционирования, определивших содержание этой лекции (рис. 7.2). При этом под **качеством функционирования понимается** множество свойств, обуславливающих пригодность программного продукта обеспечивать надежное и своевременное представление требуемой информации системе, заказчику и/или потребителю для его дальнейшего использования по назначению. В соответствии с принципиальными особенностями конкретного комплекса программ при проектировании должны выбираться номенклатура и значения показателей качества, необходимых для его эффективного применения пользователями, которые отражаются в технической документации и в **спецификациях требований** на конечный программный продукт.

Основой для формирования функциональных показателей, оцениваемых при сертификации, является анализ свойств, характеризующих качество функционирования создаваемого программного средства с учетом технологических возможностей разработчика. При этом под качеством функционирования понимается множество свойств, обуславливающих пригодность системы обеспечивать надежное и своевременное представление требуемой информации для ее дальнейшего функционального использования.

Формирование требований к функционированию программных продуктов:

- системная эффективность целевого применения программных продуктов определяется степенью удовлетворения потребностей заказчика и пользователей;
- каждое требование должно отражать отдельно распознаваемую, измеряемую сущность программного продукта;
- для контроля качества и корректности на значимость задач, следует сопоставлять конкретные требования и сформулированные цели разработки программного продукта;
- функциональную пригодность определяют: требования к количественным, измеряемым характеристикам и требования к структурным характеристикам, определяющим архитектуру комплекса программ;
- для обеспечения качества программного продукта необходим набор требуемых характеристик, свойств, их мер и значений качества для определенных потребителей программного продукта;
- следует учитывать, что требования обычно инициируются заказчиком с объемом функций, превышающим тот, который можно реализовать при выделенных ресурсах;
- ограниченные ресурсы для реализации требований функциональной пригодности, могут негативно отражаться на конструктивных характеристиках продукта.

Рис. 7.2

При выборе минимума стандартизируемых показателей качества целесообразно учитывать следующие принципы:

- ясность и измеряемость значений показателей;
- соответствие установившимся понятиям и терминологии;
- возможность последующего уточнения и детализации;
- отсутствие перекрытия между используемыми показателями;
- должны быть выделены характеристики, которые позволяют оценивать комплекс программ с позиции заказчика, пользователя, разработчика и управляющего проектом.

Для сертификации качества функционирования рекомендуется следующая система основных функциональных показателей для

обеспечения **функциональной адекватности программного средства**:

- соответствие реальных функциональных возможностей ПС декларируемым в программной документации;
- соответствие реальных функциональных возможностей аппаратных средств декларируемым в технической и эксплуатационной документации;
- обеспечение точности обработки безошибочной и актуальной исходной информации с помощью ПС для получения в итоге достаточно достоверной используемой информации;
- соответствие требованиям функциональных стандартов в области взаимодействия, сопровождаемости и переносимости ПС на различные вычислительные платформы в его жизненном цикле;
- соответствие требованиям нормативных документов по выполнению мер обеспечения защищенности информационных и программных ресурсов ПС от несанкционированного доступа.

Для оценки качества представления информации и выполнения функциональных технологических операций рекомендуются характеристики, определяющие для пользователей **доступность информационных и программных ресурсов ПС**:

- средняя наработка на отказ программно-технических средств системы;
- среднее время восстановления ПС и системы после отказа;
- коэффициент готовности ПС и системы;
- среднее время реакции ПС на запрос и/или на выполнение технологических операций;
- вероятность своевременного представления запрашиваемой информации и/или выполнения технологических операций;
- вероятность сохранения актуальности информации в базе данных на момент ее использования в системе;
- вероятность предотвращения несанкционированного доступа к программным и информационным ресурсам системы;
- вероятность сохранения конфиденциальности выходной информации системы.

Системная эффективность целевого применения программных продуктов в стандарте **ISO 9126** определяется степенью удовлетворения потребностей определенных лиц – заказчиков и/или пользователей, которые, в ряде случаев, желательно измерять экономиче-

скими категориями: прибылью, стоимостью, трудоемкостью, предотвращенным ущербом, длительностью применения и т.п. В стандартах эта эффективность отражается основной, обобщенной характеристикой качества – **функциональная пригодность**. В связи с тем, что ее абсолютную величину обычно трудно измерить непосредственно и количественно, то по ряду показателей необходима и возможна качественная оценка свойств и достоинств при применении программного продукта.

Системная эффективность – функциональная пригодность комплекса программ может быть описана количественно или качественно, в виде набора полезных свойств и характеристик программного средства, их отличий от имеющихся у других комплексов программ, а также источников возможной эффективности. Данное требование связано с тем, **какие** функции и задачи должен решать программный продукт для удовлетворения потребностей пользователей, в то время как другие, конструктивные требования главным образом связаны с тем, **как и при каких условиях**, заданные функции могут выполняться с требуемым качеством. Функциональная пригодность – это набор и описания атрибутов, определяющих **назначение, основные, необходимые и достаточные функции программного продукта**, заданные техническим заданием и спецификациями требований заказчика или потенциального пользователя.

Номенклатура и значения всех остальных показателей качества непосредственно определяются требуемыми функциями программного средства и, в той или иной степени, влияют на выполнение этих функций. Поэтому выбор функциональной пригодности, подробное и корректное описание ее свойств, являются основными исходными данными для установления требуемых значений всех остальных стандартизированных показателей качества.

При сертификации наибольшее значение для реального обеспечения качества программного продукта имеет **полнота соответствия** исходных требований к функциям и характеристикам, пространствам выполненных проверок при тестировании, адекватными им по содержанию и сценариям тестами. В результате совокупности **требований к тестам** могут применяться **как эталоны и вторая адекватная форма описания требований к комплексу программ** для сквозной верификации спецификаций требований к тестам сверху вниз, а также сами подвергаться верификации на корректность соот-

ветствия исходным требованиям к функциям и характеристикам компонентов программ разного уровня.

Функциональные требования – определяют поведение программного продукта, который должен быть создан разработчиками для предоставления возможности выполнения заказчиком и/или пользователями своих обязанностей в контексте пользовательских требований. Выбор требований к характеристикам при проектировании программных средств начинается с **определения исходных данных**. Для корректного выбора и установления требований к характеристикам качества, прежде всего, необходимо определить основные особенности программного продукта (Рис. 7.3).

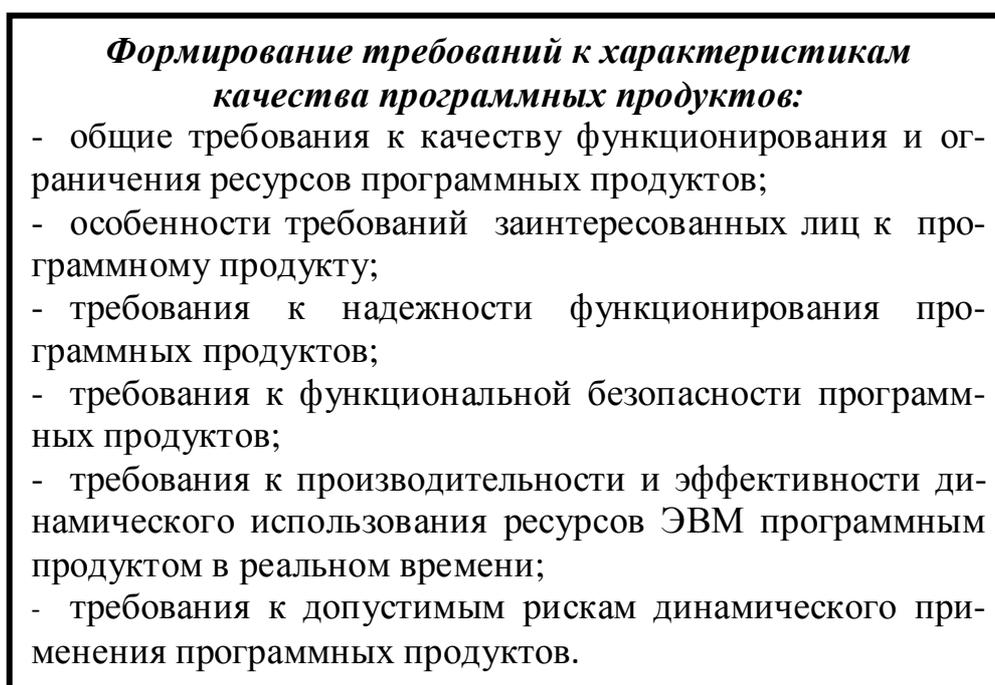


Рис. 7.3

На основе этих данных должен формироваться общий **набор требуемых характеристик, свойств, их мер и значений качества для определенных потребителей программного продукта** (рис. 7.3):

- цели создания программного продукта и назначение комплекса функциональных задач;
- общие требования к функциям и характеристикам комплекса задач программного средства;

- перечень объектов и характеристик внешней среды применения программного продукта (технологических объектов управления, подразделений предприятия и т.п.), при управлении которыми должен решаться комплекс задач;
- периодичность и продолжительность решения комплекса задач;
- связи и взаимодействие комплекса задач с внешней средой и другими компонентами системы;
- распределение функций между персоналом, программными и техническими средствами при различных ситуациях решения требуемого комплекса функциональных задач.

Проекты, как правило, инициируются заказчиком с **объемом функциональных возможностей, превышающим** тот, который разработчик может реализовать, обеспечив приемлемое качество при заданных ресурсах. Тем не менее, необходимо ограничиваться, чтобы иметь возможность предоставить в срок **достаточно целостный и качественный программный продукт**. Существуют различные методы задания очередности выполнения (приоритеты) требований и понятие базового уровня – совместно согласованного представления о том, в чем будут состоять ключевые функции системы, как продукта проекта – понятие состава требований, задающих ориентир для принятия решений и их оценки [1, 9, 17].

Если **масштаб проекта и сопутствующие требования заказчика превышают реальные ресурсы**, придется ограничиваться в функциях и качестве комплекса программ. Поэтому следует определять, что **обязательно должно** быть сделано в первой или очередной версии системы и программного продукта при имеющихся ресурсах проекта. Привлечение заказчика к итерационному решению проблемы управления масштабом и функциями повышает взаимные обязательства сторон, способствует росту взаимопонимания и доверия между заказчиком и разработчиками.

Улучшение каждой, вспомогательной – **конструктивной характеристики качества**, требует некоторых затрат ресурсов, которые в той или иной степени должны отражаться на основной характеристике качества – на функциональной пригодности. Требования к конструктивным характеристикам имеют значение для проекта постольку, поскольку они обеспечивают требуемое качество реализации основного назначения и функций ПС. При выборе требований кон-

кретных мер и шкал конструктивных характеристик качества следует учитывать возможные затраты ресурсов на их достижение и на результирующее повышение функциональной пригодности, желательно, в сопоставимых экономических единицах, в тех же мерах и масштабах. Поэтому для каждого проекта необходимо ранжировать характеристики и их атрибуты (приоритеты) и выделять, прежде всего, те требования, которые могут в наибольшей степени улучшить функциональную пригодность для конкретных целей.

Ограниченные ресурсы для реализации требований функциональной пригодности, могут негативно отражаться на конструктивных характеристиках: на надежности, безопасности, пропускной способности, качестве взаимодействия с внешней средой и с пользователями, качестве документации и других эксплуатационных факторах. Таким образом, **требования к характеристикам качества проекта комплекса программ и допустимых рисков** должны быть проанализированы и согласованы для установления в договоре между заказчиками и разработчиками исходных компромиссных значений, пригодных для эффективной разработки проекта ПС с требуемыми функциями. Для решения этих задач необходимо управление характеристиками качества и управление рисками с целью обеспечения разработчиками требуемой заказчиком функциональной пригодности.

Время или допустимая длительность разработки является невозможным ресурсом. Этот ресурс все больше определяет требования к качеству комплексов программ в процессе их разработки и сопровождения. Жесткие требования заказчиков к срокам реализации проектов, естественно, ограничивают разработчиков и испытателей. Увеличение числа привлекаемых для этого специалистов только в некоторых пределах позволяет ускорять разработку. Радикальный способ увеличения реальных затрат на разработку и испытания в ограниченное время состоит в систематизации, планировании и автоматизации на всех этапах жизненного цикла комплекса программ.

Обычно заказчики и разработчики первоначально устанавливают требования к каждой характеристике без учета относительных затрат на их достижение, а также без детального анализа их совместного влияния на полную функциональную пригодность у потребителей. Это может приводить к значительным перекосам и **несбалансированным значениям требований** к отдельным, взаимосвязанным характеристикам качества, на которые не рационально используются

ограниченные ресурсы проекта, или к не адекватно низким их значениям. В проектах ПС это может угрожать значительным повышением стоимости и/или снижением конкурентоспособности создаваемого программного продукта из-за недостаточного уровня отдельных показателей качества.

Особые классы функциональных задач решаются **динамическими системами и программными продуктами реального времени**. Такие современные функциональные задачи и системы быстро растут по сложности, по размеру комплексов программ и потребной производительности вычислительных машин. Они широко применяются во всех отраслях промышленности, в управлении народным хозяйством и в оборонной технике. Для динамического решения таких задач необходима разработка и применение **специального класса** программных средств для управления вычислительными процессами – **операционных систем реального времени**. Они должны оптимизировать динамическое использование различных ресурсов и производительности ЭВМ при интенсивных потоках информации из внешней среды от различных объектов, при разнообразии последовательностей и длительностей динамического решения отдельных функциональных задач.

Значение **реального времени** в таких системах при **решении функциональных задач** должно быть унифицировано и приведено к **единой глобальной или территориальной шкале** на пространстве использования однотипных систем с точностью, зависящей от динамических характеристик функционирующих объектов. При обмене информацией между компонентами таких систем все сообщения о состоянии наблюдаемых объектов должны снабжаться реальным временем в соответствии с единой шкалой. Несинхронный обмен информацией об объектах между различными ЭВМ в таких системах может потребовать организации в них механизмов прерываний вычислений для обеспечения сохранения реального времени приоритетного решения определенных функциональных задач и вычислительных процессов.

Требования к операционным системам реального времени для автоматизации управления и обработки информации о динамических объектах состоят в следующем:

- в процессах управления и решения функциональных задач в ЭВМ должно использоваться единое реальное время систем управле-

ния и обработки информации динамических объектов, а также внешней среды;

- потоки сообщений из внешней среды для обработки могут быть независимыми, несинхронными, различными по интенсивности, содержанию, реальному времени формирования и поступления для обработки;

- информация в сообщениях от источников и объектов внешней среды должна содержать реальное время, к которому относятся сообщения, их координаты и характеристики состояния;

- реальное время решения различных функциональных задач должно эффективно упорядочиваться в соответствии с установленной дисциплиной диспетчеризации, определенной при производстве системы, и реальным временем приема информации из внешней среды;

- алгоритмы и программы функциональных задач системы могут различаться по длительности решения и важности для пользователей, должны учитывать реальное время получения результатов решений и исходной информации;

- для эффективного использования ограниченных ресурсов производительности и оперативной памяти ЭВМ целесообразно применять приоритеты и временное прерывание исполнения некоторых программ в соответствии с выбранными дисциплинами последовательности решения различных функциональных задач;

- информация и сообщения для потребителей и внешней среды могут выдаваться асинхронно, в соответствии с установленной дисциплиной и содержать значения реального времени, которому соответствуют данные в сообщениях.

Исследован, решен и практически апробирован ряд **крупных научно-технических задач создания операционных систем и программных продуктов реального времени** для обработки информации на различных ЭВМ. В них функциональные алгоритмы и программы могут обеспечивать для операторов визуализацию информации об объектах, ее прием, обработку и корректировку в ЭВМ **в соответствии с реальным временем деятельности человека**. Операторы могут фиксировать состояния, характеристики и пространственные координаты каждого объекта с идентификатором на графическом индикаторе, и вводить их в ЭВМ.

Требования к характеристикам комплексов программ, определяющие их функциональную пригодность, *различаются на:*

- *требования к количественным, измеряемым характеристикам*, непосредственно влияющим на оперативное функционирование и возможность применения программного продукта в системе, в которые входят требования к надежности, безопасности, производительности, допустимым рискам применения;
- *требования к структурным характеристикам*, определяющим архитектуру комплекса программ, влияющие на возможности его модификации и сопровождения версий, на мобильность и переносимость на различные платформы, на документированность, удобство практического освоения и применения программного продукта вне оперативного функционирования.

Формализация требований к качеству программных продуктов могут проводиться с двух позиций: с *позиции положительной*, обеспечения эффективности и непосредственной адекватности их характеристик качества – назначению, целям создания и применения программного продукта, а также с *негативной позиции* возможного при этом ущерба – допустимого риска от дефектов и недостатков при применении комплекса программ или системы. *Характеристики качества и риски* объектов и процессов обычно тесно связаны, на них влияют подобные внутренние и внешние факторы, которые с разных сторон отражаются на свойствах систем или комплексов программ. Требуемые и реальные характеристики качества влияют на риски, а остающиеся риски влияют на характеристики качества программного продукта. Повышению характеристик качества проекта обычно *сопутствует* снижение его рисков и наоборот, сокращение рисков способствует улучшению характеристик качества и повышению функциональной пригодности. Их сбалансированное взаимовлияние должно обеспечивать *требуемую функциональную пригодность* программного продукта [3, 10, 12].

В самом начале и в процессе развития проекта, необходимо контролировать *качество и корректность требований*. Преимущества такого подхода состоят в том, что сводятся к минимуму дорогие переделки за счет уменьшения числа дефектов требований, которые можно обнаружить или предотвратить на ранних этапах жизненного цикла проекта или его версии. Определение требований напрямую связано с процедурами проверки и утверждения (аттестации), как это

сформулировано в **ISO12207**. Принято считать, что требования описаны не полностью, если для них не заданы правила – **проверки и аттестации**, то есть, не определены способы контроля корректности и утверждения. Процедуры проверки требований являются отправной точкой для специалистов по качеству, непосредственно отвечающих за соответствие получаемого программного продукта предъявляемым к нему требованиям.

Требования должны быть **верифицируемыми**. Требования, которые не могут быть проверены и аттестованы (утверждены) – это всего лишь **не обязательные пожелания**. Именно так они будут восприниматься разработчиками, даже в случае их высокой значимости для заказчика и пользователей. Если описанное требование не сопровождается процедурами проверки – в большинстве случаев говорят о недостаточной детализации или неполном описании требования и, соответственно, спецификация требований должна быть отправлена на доработку, и если необходимо, должны быть предприняты дополнительные усилия, направленные на конкретизацию требований.

Чтобы протестировать **требования на значимость**, следует сопоставить требования и сформулированные цели разработки программного продукта:

- способствует ли требование достижению целей проекта;
- если исключить требование, помешает ли это достижению целей;
- существуют ли другие требования, которые зависят от данного.

В спецификации необходимо объективно определить все, что должен делать продукт, а также те условия внешней среды, при которых он должен применяться и функционировать. Наиболее ответственный аспект формирования требований – общение со специалистами, которые вносят требования. При наличии согласованного способа формирования требований все заинтересованные стороны могут принимать участие в процессе определения требований. Как только сформулировано хотя бы одно требование, можно приступать к его **тестированию на корректность** и задавать заинтересованным сторонам подробные вопросы для уточнения и конкретизации тестов. Можно применять различные тесты для **проверки того, что требование существенно** и что все ответственные участники проекта понимают его смысл одинаково.

Особенности требований заинтересованных лиц к программному продукту

При разработке требований к комплексу программ необходимо *выделять и ранжировать по приоритетам заинтересованных лиц*, которым необходимы определенные функции и показатели качества программного продукта с учетом их специализации и профессиональных интересов. Широкая номенклатура характеристик, представленная в стандарте **ISO 9126**, определяет разнообразные требования, из которых следует селектировать и выбирать те, которые необходимы *с позиции потребителей этих данных* [7, 17]:

- *заказчиков*, для которых важно регламентировать и оценивать программный продукт по значениям требований к функциям и характеристикам, заданным и утвержденным в техническом задании и спецификациях требований и определяющих, прежде всего, назначение, функции и сферу применения программного продукта;
- *пользователей*, для которых необходима функциональная пригодность, корректность, надежность и другие показатели качества при оперативном использовании программного продукта по основному назначению;
- *разработчиков*, для которых особенно важны: ясность и конкретность описаний требований к функциям и характеристикам программного продукта, его возможная архитектура и интерфейсы между компонентами и с внешней средой;
- *специалистов сопровождающих и модифицирующих* комплекс программ, которые отдают приоритет характеристикам, поддерживающим сопровождение и конфигурационное управление версиями комплекса и его компонентов;
- *лицам, ответственным за инсталляцию*, внедрение и реализацию программного продукта в различных аппаратных и операционных средах, для которых наиболее важны характеристики обеспечения мобильности.

Первые две группы потребителей характеристик качества, заинтересованы, преимущественно, в реализации основных функций, проявляющихся в процессе использования конечного, готового программного продукта. Для этих потребителей при выборе важно выделять и по возможности формализовать внешние, эксплуатационные характеристики на завершающих этапах ЖЦ версий. К ним относятся, прежде всего, высокие приоритеты для надежности, эффективно-

сти и практичности. Для заказчиков приоритетными могут быть также сопровождаемость и мобильность версий, которые для оперативных пользователей программных продуктов обычно являются второстепенными.

Последние три группы потребителей интересуют преимущественно характеристики комплексов программ на промежуточных этапах ЖЦ, на которых проявляются, в основном, внутренние структурные, технологические свойства программного продукта, влияющие также на сопровождаемость и мобильность. Их можно не представлять в составе эксплуатационной документации для оперативных пользователей и отражать только в технологической документации разработчиков, специалистов по сопровождению и внедрению программ и данных, а также поставлять заказчикам по специальному запросу. Для этих потребителей надежность и практичность отходят на второй план, однако, ресурсная эффективность может оставаться высоко приоритетной.

Приоритеты потребителей при формировании требований к комплексу программ отражаются не только на выделении важнейших для них критериев и ранжировании приоритетов других характеристик, но также на возможности исключения из анализа некоторых характеристик качества, которые для данного потребителя не имеют значения. После определения назначения и функций ПС подготовка исходных данных и концепции проекта должны завершаться **выделением номенклатуры приоритетных функций и характеристик качества**, имеющих достаточное влияние на функциональную пригодность для определенных потребителей.

Команда разработчиков должна применить методы и процессы для того, чтобы **понять решаемую проблему заказчика до начала разработки программного комплекса**. Для этого следует использовать **анализ, выявление и освоение проблемы и интересов заказчика**:

- достигнуть соглашения между заказчиком и разработчиком по определению проблемы, целей и задач проекта;
- выделить основные причины – проблемы, являющиеся их источниками и стоящие за основной проблемой проекта системы и ПС;
- выявить заинтересованных лиц и пользователей, чье коллективное мнение и оценка в конечном итоге определяет успех или неудачу проекта;

- определить, где приблизительно находятся область и границы возможных решений проблем;
- понять ограничения, которые будут наложены на проект, команду и решения проблем.

Понимание потребностей пользователей необходимый организационный этап, так как разработчики редко получают совершенные спецификации требований к создаваемой системе, они должны сами «добывать» информацию, необходимую им для успешной работы. Термин **выявление требований** точно отражает данный процесс, в котором разработчики должны играть активную роль. Чтобы помочь производственной команде решить эти проблемы, лучше понять потребности пользователей и других заинтересованных лиц, целесообразно использовать **методы** [7]:

- интервьюирования и анкетирования – создание структурированного интервью;
- проведение интервью с пользователями и/или заинтересованными лицами;
- подведение итогов совокупности интервью, формулирование наиболее часто упоминавшихся функциональных и архитектурных потребностей заказчика и пользователей;
- совещания, посвященные анализу и синтезу требований – формулирование и определение целей программного продукта; ознакомление с ними всех участников проекта и установление, что они с ними согласны; если это не так, следует остановиться и уточнениями добиться согласия;
- анализ иллюстративных прецедентов в приложении к концепции требований, чтобы их функции были наглядны и понятны;
- по возможности выявление или создание прототипов на основе первичных требований.

Хотя ни один из методов не является универсальным, каждый из них позволяет лучше понять потребности пользователей и тем самым превратить неясные требования, в требования, которые известны и понятны.

Требования к надежности функционирования программных продуктов

Надежность является внутренним свойством системы и программного продукта, заложенным при создании и проявляющимся **во**

времени при функционировании и эксплуатации – стандарт **ISO 9126** [3, 13]. Программа любой сложности и назначения при строго фиксированных исходных данных и абсолютно надежной аппаратуре исполняется по однозначно определенному маршруту и дает на выходе строго определенный результат. Однако случайное изменение исходных данных и накопленной при обработке информации, а также множество условных переходов в программе создают огромное число различных маршрутов исполнения каждого сложного комплекса программ. Источниками ненадежности являются непроверенные тестированием сочетания исходных данных, при которых функционирующие программы дают неверные результаты или отказы. В результате комплекс программ **не соответствует требованиям функциональной пригодности и работоспособности**.

При применении понятий надежности к программным комплексам следует учитывать **особенности и отличия этих объектов от традиционных технических систем**, для которых первоначально разрабатывалась теория надежности:

- не для всех видов программ применимы понятия и методы теории надежности – их можно использовать только к программным продуктам, функционирующим в реальном времени и непосредственно взаимодействующим с внешней средой;
- при разработке и оценке качества программных компонентов к ним не применимы понятия надежности функционирования, если при обработке информации они не используют значения реального времени;
- доминирующими факторами, определяющими значения надежности программ, являются дефекты и ошибки проектирования и разработки, и второстепенное значение имеет физическое разрушение программных компонентов при внешних воздействиях;
- для повышения надежности комплексов программ особое значение имеют методы автоматического сокращения длительности восстановления и преобразования отказов в кратковременные сбои, путем введения в программные средства временной, программной и информационной избыточности;
- непредсказуемость места, времени и вероятности проявления дефектов и ошибок, а также их редкое обнаружение при реальной эксплуатации достаточно надежных программных продуктов, не позволяет эффективно использовать традиционные методы априорного

расчета показателей надежности, ориентированные на стабильные, измеряемые значения надежности составляющих компонентов;

- традиционные методы форсированных испытаний надежности систем путем физического воздействия на их компоненты не применимы для программных продуктов и их следует заменять на методы форсированного воздействия информационных потоков тестов из внешней среды.

Определение степени работоспособности системы предполагает наличие в ней средств, способных установить *соответствие ее характеристик требованиям технической документации*. Для этого должны использоваться методы и средства оперативного контроля и диагностики функционирования программного продукта и системы. Глубина и полнота проверок, степень автоматизации контрольных операций, длительность и порядок их выполнения влияют на работоспособность системы и достоверность ее оценки. Методы и средства диагностического контроля предназначены для установления степени работоспособности системы, локализации отказов, определения их характеристик и причин, и скорейшего восстановления работоспособности.

В международном стандарте **ISO 9126**, при отборе минимума стандартизируемых показателей выдвигались и учитывались следующие принципы: ясность и измеримость значений, отсутствие перекрытия между используемыми показателями, соответствие установившимся понятиям и терминологии, возможность последующего уточнения и детализации. **Надежность**: свойства комплекса программ обеспечивать достаточно низкую вероятность потери работоспособности – отказа, в процессе функционирования программного продукта в реальном времени. Основные атрибуты надежности могут быть объективно *измерены и сопоставлены с требованиями*.

Завершенность: свойство комплекса программ не попадать в состояния отказов вследствие ошибок и дефектов в программах, данных и внешней среде.

Устойчивость к дефектам и ошибкам: свойство программного продукта автоматически поддерживать заданный уровень качества функционирования при проявлениях дефектов и ошибок или нарушениях установленного интерфейса между компонентами и с внешней средой.

Восстанавливаемость: свойство комплекса программ в случае отказа возобновлять требуемый уровень качества функционирования, а также поврежденные программы и данные. После отказа, программный продукт иногда может быть неработоспособен в течение некоторого времени, продолжительность которого определяется его восстанавливаемостью в процессе перезапуска – рестарта. Перезапуск должен обеспечивать возобновления нормального функционирования ПС, на что требуются ресурсы ЭВМ и время. Поэтому полнота и длительность восстановления функционирования после сбоев отражают качество и надежность программного продукта, и возможность его использования по прямому назначению.

Доступность или готовность: свойство программного продукта быть в состоянии выполнять требуемую функцию в данный момент времени при заданных условиях использования. Внешне, готовность может оцениваться относительным временем, в течение которого комплекс программ находится в работоспособном состоянии, в пропорции к общему времени применения.

Понятие **корректной (правильной) программы** может рассматриваться статически вне ее исполнения во времени. Корректность программы не определена вне области изменения исходных данных, **заданных требованиями спецификации**, и не зависит от динамики функционирования программы в реальном времени. Степень некорректности программ определяется вероятностью попадания реальных исходных данных в пространство значений, которое задано требованиями спецификации и технического задания, однако не было проверено при тестировании и испытаниях.

Надежная программа, прежде всего, должна обеспечивать достаточно низкую вероятность отказа в процессе функционирования в реальном времени. Быстрое реагирование на искажения программ, данных или вычислительного процесса и восстановление работоспособности за время меньшее, чем порог между сбоем и отказом, обеспечивают высокую надежность программ. При этом не корректная программа может функционировать абсолютно надежно. В реальных условиях по различным причинам исходные данные могут попадать в пространство значений, вызывающих сбои, не проверенные при испытаниях, а также не заданные требованиями спецификации и технического задания. Если в этих ситуациях происходит достаточно быстрое восстановление, такое что не фиксируется отказ, то такие собы-

тия не влияют на основные показатели надежности – наработку на отказ и коэффициент готовности. Следовательно, надежность функционирования программ **является понятием динамическим**, проявляющимся во времени и существенно отличается от понятия корректности программ.

При оценке реализации требований надежности регистрируются только такие искажения в процессе динамического исполнения программ, которые приводят к потере работоспособности продукта. Реальные исходные данные могут иметь значения, отличающиеся от заданных требованиями и от использованных при применении программ. Непредсказуемость вида, места и времени проявления дефектов в процессе эксплуатации приводит к необходимости создания специальных, дополнительных **систем оперативной защиты** от непредумышленных, случайных искажений вычислительного процесса, программ и данных. Системы оперативной защиты предназначены для выявления и блокирования распространения негативных последствий проявления дефектов и уменьшения их влияния на надежность функционирования комплекса программ до устранения их первичных источников. Для этого в комплекс программ должна вводиться временная, программная и информационная избыточность, осуществляющая оперативное обнаружение дефектов функционирования, их идентификацию и автоматическое восстановление (рестарт) нормального функционирования.

Основным **принципом классификации сбоев и отказов** в программных продуктах при отсутствии их физического разрушения является разделение по временному показателю длительности восстановления после любого искажения программ, данных или вычислительного процесса, регистрируемого как нарушение работоспособности [13]. При длительности восстановления, меньшей заданного порога, дефекты и аномалии при функционировании программ следует относить к **сбоям**, а при восстановлении, превышающем по длительности пороговое значение, происходящее искажение соответствует **отказу**. Временная зона перерыва нормальной выдачи информации и потери работоспособности, которую следует рассматривать как зону сбоя, тем шире, чем более инертный объект находится под воздействием сообщений, подготовленных программным продуктом.

Требования к надежности программных продуктов в значительной степени адекватны аналогичным характеристикам, принятым

для других технических систем. Наиболее широко используется **критерий длительности наработки на отказ**. Для определения этой величины измеряется время работоспособного состояния системы между последовательными отказами или началами нормального функционирования системы после них. Критерий надежности восстанавливаемых систем учитывает возможность многократных отказов и восстановлений. Для оценки надежности таких систем, которыми чаще всего являются сложные программные продукты, кроме вероятностных характеристик наработки на отказ, важную роль играют характеристики функционирования после отказа в процессе восстановления.

Основным **требованием к процессу восстановления** является **длительность восстановления** и ее вероятностные характеристики. Этот критерий учитывает возможность многократных отказов и восстановлений. Обобщение характеристик отказов и восстановлений производится в критерии **коэффициент готовности**. Этот показатель отражает вероятность иметь восстанавливаемую систему в работоспособном состоянии в произвольный момент времени. Значение коэффициента готовности соответствует доле времени полезной работы системы на достаточно большом интервале, содержащем отказы и восстановления.

В реальных системах **требуемая наработка на отказ** программного продукта обычно должна быть соизмерима с наработкой на отказ аппаратуры, на которой исполняется программа. Для систем обработки информации и управления в реальном времени наработка на отказ комплекса программ измеряется десятками и сотнями часов, а для особо важных или широко тиражируемых систем может достигать десятков тысяч часов. При достаточно развитом программном оперативном контроле и восстановлении, наработка на отказовую ситуацию может быть на один-два порядка больше, чем наработка на отказ. Реально очень трудно достигать наработку на отказовую ситуацию на уровне тысяч часов и поэтому для получения высокой надежности программ **требуется использовать средств рестарта**.

Требования к функциональной безопасности программных продуктов

Наиболее полно функциональная безопасность комплексов программ характеризуется **величиной ущерба**, возможного при проявле-

нии дестабилизирующих факторов и реализации конкретных *угроз – рисков*, а также средним временем между проявлениями непредумышленных угроз, нарушающих безопасность. Однако описать и измерить, в общем виде, возможный ущерб при нарушении безопасности для критических ПС разных классов практически невозможно. Поэтому реализации угроз можно характеризовать интервалами времени между их проявлениями, нарушающими безопасность применения ПС, или наработкой на отказы, отражающиеся на безопасности. Это *сближает понятия и требования безопасности с показателями надежности* комплексов программ. Различие состоит в том, что в показателях надежности учитываются все реализации отказов, а в характеристиках функциональной безопасности следует регистрировать только те *случайные катастрофические отказы*, которые отразились на безопасности. Статистически таких отказов может быть в несколько раз меньше, чем учитываемых в значениях надежности. Однако методы, влияющие факторы и реальные значения показателей надежности могут служить ориентирами при оценке функциональной безопасности критических программных продуктов.

При формировании требований к средствам обеспечения функциональной безопасности программных продуктов систем, необходимо учитывать, что такие средства не могут быть абсолютно безупречными и корректными. Непредусмотренные при проектировании некоторые ситуации и ошибки функционирования программ и данных могут быть потенциальными источниками катастроф при применении таких программных продуктов, *влияющими на безопасность их функционирования и применения* [4, 13]. Поэтому при подготовке требований к безопасности программных продуктов целесообразно учитывать и конкретизировать особенности источников возможного нарушения корректности их функционирования:

- дефекты методов, алгоритмов и функционального содержания процессов, принятых для решения задач обеспечения безопасности для корректного применения системы по ее назначению;
- недостаточное качество технологии и производственных процессов, использованных для реализации методов и алгоритмов обеспечения безопасности, определяющих возможность проявления технологических дефектов и ошибок при применении программного продукта;

- нарушения внешней средой, системами и/или пользователями требований к созданным средствам обеспечения безопасного функционирования и применения программных продуктов.

В результате при использовании программного продукта по прямому назначению в реальной внешней среде, при проявлениях ошибок, искажениях исходных данных и других непредсказуемых событиях возможны **опасные отказовые ситуации**, аномалии функционирования и искаженные результаты, нарушающие безопасность системы. Предвидеть заранее все подобные ситуации и протестировать при них сложные ПС оказывается невозможным из-за их огромного количества. Один из видов ущерба при отказовых ситуациях, определяющих функциональную безопасность, состоит в **прерывании его работоспособности на длительное время**. Контроль продолжительности потенциальной работоспособности системы и ПС, а также длительности отказовых ситуаций может **требовать повышения функциональной безопасности** путем автоматизированного сокращения времени неработоспособного состояния системы.

Отказовые ситуации становятся катастрофическими и отражаются на безопасности, если длительность прерывания работоспособного состояния превышает некоторое пороговое значение, специфическое для конкретной системы и внешней среды. Если удастся обнаружить опасную отказовую ситуацию и быстро восстановить работоспособность за время не превышающее заданный порог, то эта ситуация может не отразиться на функциональной безопасности системы. Для этого применяются методы и средства, которые направлены на **автоматическое обнаружение опасных отказов** при реальном функционировании комплексов программ и на достаточно быстрое оперативное автоматическое восстановление нормального вычислительного процесса, текстов программ и данных (**рестарт**). В любых ситуациях, прежде всего, должны исключаться катастрофические последствия и длительные, опасные отказы или в максимальной степени смягчаться их влияние на безопасность результатов.

Введение в программы средств контроля и оперативной защиты, позволяет **скомпенсировать их неполную корректность**, а также снижать влияние на безопасность негативных возмущений различных типов. Выбор метода оперативного восстановления обычно происходит в условиях значительной неопределенности сведений о характере отказовой ситуации и степени ее возможного влияния на работос-

пособность ПС. Кроме того, восстановление работоспособности желательно производить как можно быстро, чтобы отказовую ситуацию можно было свести до уровня кратковременного сбоя.

Требования к обеспечению функциональной безопасности программных продуктов – установлены в международном стандарте **IEC 61508:3**. В нем представлен общий подход к видам деятельности на протяжении цикла обеспечения безопасности для систем, содержащих программируемые электронные компоненты систем (ПЭС), которые используются для выполнения различных функций и, в частности, для обеспечения безопасности систем. Стандарт **IEC 61508-3** содержит:

- метод разработки спецификаций требований по безопасности ПЭС, необходимых для достижения заданной функциональной безопасности;
- основанный на рисках подход, для определения требований к уровням соответствия комплексу требований на функциональную безопасность;
- устанавливаются количественные меры отказов для систем безопасности ПЭС, связанные с уровнями соответствия комплексу требований систем по функциональной безопасности;
- требования для этапов жизненного цикла обеспечения безопасности и деятельности, которая должна проводиться при проектировании и разработке программного продукта, обеспечивающего безопасность;
- требования для информации, относящейся к аттестации программных продуктов, которая должна передаваться организациям, производящим компоновку ПЭС в составе системы;
- требования, которые должны выполнять организации, производящие модификацию программного продукта, обеспечивающего безопасность;
- требования к технологическим средствам производства, таким как инструментальные средства проектирования и разработки, языковые трансляторы, инструментальные средства для тестирования, отыскания и устранения ошибок, средства управления конфигурацией.

Цикл обеспечения требований безопасности при разработке комплекса программ должен быть выбран и задан при планировании безопасности для практических нужд проекта или предприятия. В

деятельность, связанную с циклом обеспечения безопасности, должны быть включены процедуры гарантирования качества и безопасности. Рекомендуется использовать основные положения, комплекс этапов и работ, представленный в стандарте **ISO 12207. Спецификация требований по безопасности программного продукта** должна быть достаточно подробной для того, чтобы конструкция и ее выполнение достигали требуемого соответствия комплексу требований по безопасности, и можно было произвести оценку функциональной безопасности. В спецификации должны быть **требования к функциям безопасности**:

- обеспечивающие достижение и поддержание безопасного состояния программного продукта или системы;
- относящиеся к выявлению отказов; извещения об их наличии и управление отказами в аппаратуре; отказами исполнительных механизмов, а также в программном продукте;
- относящиеся к периодической проверке функций обеспечения безопасности в оперативном режиме и в отключенном состоянии.

Аттестация программного продукта должна быть использована для демонстрации того, что программный продукт **удовлетворяет требованиям по его безопасности**. План аттестации безопасности программного продукта должен содержать требуемые условия окружающей среды, в которой должна производиться аттестация; политику и процедуры оценки результатов аттестации.

Требования к информационной безопасности комплексов программ формализует стандарт **ISO 15408:1-3** [17]. Он состоит из трех частей, отражающих требования и рекомендации по обеспечению безопасности систем, содержащих программные средства. Первая часть определяет концепцию всего стандарта, а вторая самая большая часть формализует методы обеспечения безопасности. Его третья часть полностью посвящена требованиям и процессам обеспечения **доверия – (качества) компонентов** систем, реализующих функции их безопасности. Положения этой части стандарта трактуются с позиции обеспечения функциональной безопасности, а термин – **доверие** применяется как понятие **качество или уверенность выполнения методических и технологических требований безопасности**. Рекомендуется ряд шагов для предотвращения уязвимостей, возникающих в продуктах и системах, которые могут проявляться **вследствие недостатков**:

- требований – программный продукт или система могут обладать требуемыми от них функциями и свойствами, но содержать методические или алгоритмические уязвимости, которые делают их непригодными или неэффективными в части безопасности применения;
- проектирования – программный продукт или система не отвечают спецификации, и/или уязвимости, являются следствием некачественных процессов проектирования и производства или неправильных проектных решений;
- эксплуатации – программный продукт или система разработаны в полном соответствии с корректными спецификациями требований, но уязвимости возникают как результат неадекватного применения и/или управления при эксплуатации.

Оценка и утверждение *целей функциональной безопасности* требуется для демонстрации заказчику или пользователю, что установленные цели проекта адекватны методам обеспечения его безопасности. Существуют цели и функции безопасности для ПС и цели безопасности для среды. Рекомендуются сопоставлять эти цели безопасности с идентифицированными угрозами, которым они противостоят, и/или с политикой и предположениями, которым они должны соответствовать. Использование стандарта означает, что *требования могут быть четко идентифицированы*, что они автономны, и применение каждого требования возможно и даст значимый результат при оценке качества, основанный на анализе соответствия продукта этому конкретному требованию.

Активное исследование *доверия* – это оценка процесса функционирования системы для определения свойств безопасности. В стандарте применяется *иерархия детализации требований к безопасности* и качеству систем и комплексов программ с использованием специфических терминов: *класс* – наиболее общая характеристика качества, которая структурируется *семейством*. Каждое семейство может иметь несколько *компонентов* – атрибутов, состоящих из *элементов* качества. Семейство доверия (качества) в стандарте может содержать один или несколько компонентов доверия и требований для выполнения.

Класс *руководства по безопасности определяет требования*, направленные на обеспечение понятности, достаточности и законченности эксплуатационной документации, представляемой разработчиком. Руководство администратора – основной документ, имею-

щийся в распоряжении разработчика, для предоставления администраторам объекта, детальной и точной информации о том, как осуществлять администрирование безопасным способом и эффективно использовать доступные процедуры обеспечения безопасности. Требования к руководству пользователя должны обеспечивать возможность эксплуатировать объект безопасным способом.

Класс *поддержка жизненного цикла* определяет требования для реализации всех этапов разработки четко определенной модели, включая политики и процедуры устранения недостатков и дефектов, правильное использование инструментальных средств и методов, а также меры безопасности для защиты среды разработки. Безопасность разработки охватывает физические, процедурные, относящиеся к персоналу и другие меры безопасности, используемые применительно к среде разработки. Определение жизненного цикла ПС устанавливает, что технология разработки, используемая разработчиком для создания объекта, включает в себя положения и действия, указанные в требованиях к процессу разработки и поддержке эксплуатации.

Класс *оценка уязвимости комплекса программ определяет требования*, направленные на идентификацию уязвимостей, которые могут проявиться и быть активизированы. Анализ неправильного применения позволяет выяснить, способен ли администратор или пользователь, используя руководства, определить, что система или ПС конфигурированы или эксплуатируются небезопасным способом. Анализ уязвимостей заключается в идентификации недостатков, которые могли быть внесены на различных этапах разработки. Эти потенциальные уязвимости оцениваются посредством тестирования проникновения, позволяющим сделать заключение, могут ли они в действительности быть использованы для нарушения безопасности системы и программного продукта.

Для *систематического применения классов и семейств требований*, в стандарте ведено понятие *профиль защиты (ПЗ)* – независимая от реализации совокупность требований безопасности для некоторой категории объектов, отвечающая специфическим требованиям проекта и потребителя. Цель разработки и оценки ПЗ – показать, что он является полным, непротиворечивым, технически правильным и поэтому пригоден для изложения конкретных требований к одному или нескольким типам объектов. *Задание по безопасности (ЗБ)* – со-

вокупность требований и спецификаций, предназначенная для использования в качестве основы для оценки конкретного объекта, что оно является полным, непротиворечивым, технически правильным и поэтому пригодно для использования в качестве основы при оценке уровня безопасности соответствующей системы.

Требования к производительности и эффективности динамического использования ресурсов ЭВМ программным продуктом в реальном времени

Для систем реального времени особое значение имеют **требования эффективного использования** программным продуктом ресурсов ЭВМ по производительности. Эффективность в стандарте **ISO 9126** отражена двумя динамическими характеристиками требований – временной эффективностью и используемостью ресурсов ЭВМ, которые рекомендуется описывать в основном количественными атрибутами, учитывающими динамику функционирования комплексов программ. В этих стандартизированных характеристиках отражается только **частная, конструктивная эффективность** использования ресурсов ЭВМ, которую не следует смешивать с требованием **системной эффективности** функциональной пригодности программного продукта при применении в конкретной системе. Основные требования к характеристикам эффективности использования вычислительных ресурсов сосредоточены на наиболее критичных показателях производительности и длительности решения функциональных задач. При этом в контракте, техническом задании и спецификации требований должны быть зафиксированы и утверждены **требуемые значения** этих характеристик и их приоритетов. В стандарте **ISO 9126** эти характеристики качества комплексов программ рекомендуется отражать рядом атрибутов, каждый из которых следует оценивать для средних и наихудших сценариев функционирования комплекса программ.

Временная эффективность: свойства комплекса программ, отражающие требуемое время обработки заданий, **время отклика** из ЭВМ в систему и/или внешнюю среду после получения типового задания и начала решения требуемой функциональной задачи; а также производительность решения задач с учетом количества используемых вычислительных ресурсов (**ISO 14756**). Временная эффективность программного продукта определяется длительностью выполне-

ния заданных функций и ожидания результатов в средних и/или наилучших случаях, с учетом приоритетов задач. **Пропускная способность** решения функциональных задач – производительность, число заданий, которое можно реализовать на данной ЭВМ в заданном интервале времени в зависимости от их содержания и числа действующих пользователей или воздействий из внешней среды.

Используемость ресурсов: степень загрузки доступных вычислительных ресурсов в течение заданного времени при выполнении функций комплекса программ в установленных условиях. Ресурсная экономичность отражается занятостью ресурсов центрального процессора, оперативной, внешней и виртуальной памяти, каналов ввода-вывода, терминалов и каналов сетей связи исполнением программ.

Потребность в производительности ЭВМ в процессе решения функциональных задач может значительно изменяться в зависимости от их свойств, а также от потока, состава и объема исходных данных. Степень использования ограниченной производительности ЭВМ в некоторых пределах не влияет на качество решения функциональных задач комплексом программ. При излишне высокой интенсивности поступления исходных данных может **нарушаться временной баланс** между длительностью решения полной совокупности задач комплексом программ в реальном времени, и производительностью ЭВМ при решении этих задач. Для формирования требований к программному продукту при подготовке технического задания и спецификаций **следует согласовывать с заказчиком динамическую модель и характеристики внешней среды**, в которой будет применяться комплекс программ, а также динамику приема и передачи данных пользователям или системе. Оценивание величины производительности рекомендуется для определения: загрузки операторов-пользователей, пропускной способности по числу задач в единицу времени, временной шкалы событий обработки заданий и данных в системе. Эти результаты предлагается сравнивать с требованиями заказчика и пользователей для оценивания рабочих нагрузок и достаточности производительности программного продукта в конкретной системе и внешней среде.

Эти условия следует детализировать до уровня, позволяющего однозначно формализовать **требования к допустимым значениям интенсивности решения функциональных задач:**

- в среднем, нормальном режиме работы программного продукта с наибольшим качеством функциональной пригодности;
- в режиме предельной загрузки, реализующейся с определенной вероятностью, с допустимым снижением функциональной пригодности и некоторых конструктивных характеристик качества;
- в режиме кратковременной, аварийной перегрузки, способной критически отражаться на функциональной пригодности, надежности и безопасности применения программного продукта.

Наиболее сложным является *задание требований эффективности динамического использования ресурсов производительности ЭВМ в реальном времени*. При этом должна быть определена зависимость качества решения функциональных задач от интенсивности поступающей информации различных типов. Основная задача состоит в определении вероятностей и рисков, с которыми может нарушаться соответствие между потребностями в производительности для решения всей требуемой совокупности задач и реальными возможностями ЭВМ и других компонентов системы. Если эта вероятность невелика, и можно считать допустимым эпизодическое снижение качества за счет получающихся задержек и пропусков в обработке сообщений или заданий, то делается вывод о соответствии производительности ЭВМ функциям данного программного продукта. В зависимости от характеристик потоков заданий и предполагаемых длительностей их реализации, могут распределяться приоритеты на их решения, и тем самым, повышаться эффективность использования ограниченной производительности вычислительной системы для определенного комплекса программ.

Требования к допустимым рискам динамического применения программных продуктов

В предыдущих разделах рассматривались требования к характеристикам качества, которые отражают положительные свойства программных комплексов и их следует улучшать. Однако, не смотря на их реализацию в проектах комплексов программ, могут проявляться негативные свойства, которые проявляются ущербом – риском при разработке и/или применении программного продукта. К таким свойствам – рискам при создании программных продуктов должны формироваться *требования, ограничивающие их до допустимых заданных пределов* (стандарт **ISO 16085**). Для обеспечения требуемых

характеристик качества функционирования программного продукта необходима **организация контрмер** процесса управления рисками под руководством менеджера управления рисками – координатора взаимодействия заказчика и разработчиков, прежде всего, при обеспечении функциональной пригодности и обосновании проекта [14, 16, 18].

Требования к сокращению рисков проекта, связанные с результатами разработки и применения программного продукта, должны реализоваться формализованным процессом, позволяющим **выделять, систематически идентифицировать, оценивать и смягчать факторы, угрозы и величину последствий возможных рисков**. Для этого при управлении проектом следует: идентифицировать угрозы и риски, имеющие как внешние, так и внутренние причины; проводить их количественную оценку; разработку откликов и контрмер для сокращения рисков и контроль реализации откликов. Это может приводить к ухудшению качества программного продукта и к **рискам следующих характеристик**:

- **функциональной пригодности**, назначения, состава и характеристик основных, функциональных задач, решаемых системой и программным продуктом;
- **характеристик** конечного программного продукта и результатов его применения для целей функционирования системы;
- **нарушения ограничений доступных и используемых ресурсов**, к превышению допустимых затрат, нарушению сроков или же к полному срыву возможности реализации проекта комплекса программ и системы в заданных ограниченных условиях и ресурсах.

Эти группы характеристик проектов систем и комплексов программ тесно связаны между собой и определяют соответствующие классы потенциальных рисков. Изменение каждого из них может влиять на другие риски. **Требования обеспечения минимальных допустимых рисков функциональной пригодности имеют доминирующее значение** и изменения двух других классов рисков обычно должны быть, в первую очередь, подчинены сокращению рисков функционирования системы и комплекса программ. Поэтому анализ рисков и возможных угроз целесообразно проводить систематизировано, начиная с установления требований к допустимым рискам функциональной пригодности. Ущерб вследствие ошибок функцио-

нальных требований к проекту программного средства может проявляться двумя видами рисков: недостатками достигнутых характеристик программного продукта, и рисков от нарушения ограниченности доступных и используемых ресурсов в жизненном цикле программного продукта.

В жизненном цикле важнейшим риском является ущерб при невыполнении комплексом программ, назначения и функций главной характеристики качества – *функциональной пригодности*. В зависимости от назначения, функций, критичности требований к системе и программному продукту, может требоваться либо полная ликвидация определенных или всех видов рисков, либо сокращение некоторых из них до допустимых пределов, либо игнорирование некоторых и сохранение на достигнутом уровне ущерба вследствие нарушения требований заказчика к программному продукту.

Для выполнения требуемых функций комплекса программ необходима *адекватная исходная информация от объектов внешней среды*, содержание которой должно полностью обеспечивать реализацию декларированных функций. Полнота формализации номенклатуры, структуры и качества входной информации для выполнения требуемых функций, является одной из важных составляющих при определении функциональной пригодности и сокращения рисков применения комплекса программ в соответствующей внешней среде. Степень покрытия всей выходной информацией: целей, назначения и функций программного продукта для пользователей, следует рассматривать как *основное требование к допустимым рискам функциональной пригодности*. Прослеживание и оценивание адекватности и полноты состава выходной информации снизу вверх к назначению комплекса программ должны завершать выбор базовых характеристик функциональной пригодности, независимо от сферы применения системы.

При начальном формировании требований к функциональной пригодности комплекса программ практически невозможно достоверно предусмотреть сбалансированное выделение каждого вида ресурса для полной реализации каждой требуемой функции и характеристики программного продукта. Кроме того, требования заказчика к функциям всегда субъективны и не стабильны, что также отражается на изменении рисков при разработке и модификациях комплексов программ. При этом некоторые характеристики в реальном проекте

могут приобретать значения более высокие, чем действительно требуются, на что нерационально расходуются ресурсы, а другие – не удовлетворяют требованиям контракта и технического задания. Для разрешения этого противоречия основное значение имеет **деятельность менеджера рисков**, который должен быть способен прогнозировать, проводить поэтапный анализ, контроль, оценивание и мониторинг возможных и реальных отклонений от требуемых характеристик функционирования программного продукта и используемых ресурсов, **управление контрмерами и изменение их для сокращения интегрального риска** всей системы.

Требования к оценке и сокращению рисков вследствие недостаточных характеристик качества комплексов программ должны сопровождать весь их жизненный цикл. Для этого необходимо контролировать области возможного возникновения рисков, оценивать вероятности их проявления, виды и степень влияния угроз, которые следует минимизировать как можно раньше по мере их возникновения и обнаружения в процессах жизненного цикла комплекса программ. Основной эффект по снижению рисков вследствие недостаточных характеристик должен достигаться на начальных этапах разработки, когда возможно предотвращение или сокращение многих из них с минимальными затратами времени и других ресурсов. Для этого в **технологическом процессе разработки** необходимо использовать **методы, которые включают:**

- определение ценности (приоритета) и выделение каждой **требуемой характеристики** для реализации необходимой функциональной пригодности программного продукта и системы;
- определение приоритетов характеристик качества, компонентов и этапов ЖЦ проекта, которые имеют потенциальные технические, стоимостные или плановые риски;
- оценивание вероятности каждого **вида угроз**, потенциальной величины и вероятности их возможного негативного воздействия на характеристику функциональной пригодности системы и программного продукта;
- оценивание **уязвимости** и последствий дефектов каждой характеристики и затрат ресурсов для восстановления требуемой функциональной пригодности системы и программного продукта при проявлении рисков;

- систематизацию, документирование и оценивание эффективности доступных методов, средств и ресурсов **контрмер** для сокращения рисков функциональной пригодности и выделенных характеристик комплекса программ;
- планирование и разработку решений **по контрмерам** для обеспечения допустимого уровня интегрального риска функциональной пригодности и характеристик системы, в том числе, возможно, за счет **изменения требований** к программному продукту, системе и/или доступных ресурсов;
- оценку **вероятности сокращения рисков** характеристик до допустимых пределов, при реализации процессов разработки и всего ЖЦ программного средства с учетом доступных ресурсов.

Улучшение каждой характеристики, требует затрат ресурсов, которые в той или иной степени должны отражаться на основной характеристике комплекса программ – **на функциональной пригодности**. При выборе конкретных допустимых характеристик следует учитывать возможные затраты ресурсов на их достижение и на результирующее повышение функциональной пригодности, желательно, в сопоставимых экономических единицах, в тех же мерах и масштабах. Такое, даже приблизительное, качественное сравнение эффекта и затрат позволяет избегать многих **не рентабельных завышений требований** к отдельным характеристикам, которые не достаточно отражаются на адекватном улучшении функций программного продукта.

Решение этих задач должно быть направлено на обеспечение высокой функциональной пригодности ПС **путем сбалансированного улучшения остальных характеристик в условиях ограниченных ресурсов**. Для этого в процессе системного анализа при подготовке технического задания и требований спецификаций, значения и риски характеристик, должны выбираться с учетом опасности их влияния на функциональную пригодность. Излишне высокие требования к отдельным характеристикам, требующие для реализации больших дополнительных трудовых и вычислительных ресурсов, целесообразно снижать, если они слабо влияют на основные, функциональные характеристики программного продукта.

Риски ограничений доступных и используемых ресурсов в жизненном цикле комплекса программ могут включать:

- **экономические риски** – превышение разработчиком обоснованных, допустимых по контракту размеров стоимости, трудоемкости и эксплуатационных затрат на программные компоненты и продукт в целом, которые могут также отражаться на их функциональной пригодности и других характеристиках качества;
- **плановые риски** – нарушение разработчиком допустимых временных затрат в графиках работ, сроков реализации этапов и проекта в целом, а также распределений задач по подрядчикам, подразделениям и специалистам, что может также увеличивать риски характеристик программных продуктов;
- **кадровые риски** – недостаточная квалификация и число специалистов, отражающаяся на рисках качества разработки, совершенствования и/или применения комплекса программ;
- **технические риски** – недостаточность вычислительных ресурсов, несогласованность ресурсов внутренней и внешней среды для реализации основных функций, что может иметь как самостоятельное значение, так и влияние на риски программных продуктов;
- **технологические риски** – недостаточное качество инструментария для автоматизации всего жизненного цикла и технологических процессов, предназначенных для обеспечения гарантированного сокращения рисков конечного программного продукта.

Прямые риски, обусловленные ошибками заданных требований экономических характеристик, могут вызвать ущерб при **завышении** заказчиком стоимости проекта относительно реально необходимой, или ущерб разработчикам, если стоимость **оценена недостаточной** для его успешной реализации.

Требования по управлению рисками в жизненном цикле программных комплексов регламентированы международным стандартом **ISO 16085**. Он включает особенности обеспечения, мониторинга качества и функциональной безопасности сложных комплексов программ на базе анализа и сокращения рисков процессов, регламентированных стандартом **ISO 12207**. Поэтапное, иерархическое снижение интегрального риска проекта программного комплекса при использовании выбранной стратегии может требовать ее корректировки в зависимости от достигаемого эффекта и требуемых затрат на сокращение определенных рисков, необходимого для повышения качества, надежности и функциональной безопасности. Для решения этих задач **стандартом рекомендуются последовательные процедуры:**

- выявление и идентификация, относящихся к проекту комплекса программ, рисков, как в исходном состоянии и требованиях к проекту, так и на последовательных этапах его выполнения: по характеристикам качества, графикам, трудоемкости, ресурсам и техническим рискам;
- анализ вероятности проявления, причин, взаимозависимости видов и последствий рисков, чтобы сбалансировать и распределить их приоритеты, на основании которых будут отводиться ресурсы на различные контрмеры для снижения этих рисков;
- определение целесообразной стратегии и области управления каждым видом риска или их наборами в проекте, в соответствии с политикой на предприятии, а также со степенью влияния, вероятностью и типами рисков, которые должны выявляться и которыми следует управлять;
- для каждого вида риска (или набора рисков) определение метрики, отражающей изменения в состоянии рисков в зависимости от деятельности по их устранению, которые должны характеризовать изменения в вероятности проявления, последствиях и временных диапазонах проявления рисков;
- реализация разработанной стратегии управления рисками, аттестация результатов и оценка успешности стратегии в контрольных точках жизненного цикла проекта.

Лекция 8

ОРГАНИЗАЦИЯ СЕРТИФИКАЦИОННЫХ ИСПЫТАНИЙ ПРОГРАММНЫХ ПРОДУКТОВ НА СООТВЕТСТВИЕ ТРЕБОВАНИЯМ

Цели, задачи и процессы сертификационных испытаний программных продуктов

Поскольку человеческий мозг способен оперировать ограниченным числом объектов одновременно, то чем сложнее фрагмент программы, тем более вероятно, что специалист допустит или пропустит в нем ошибку. Сложность создания необходимых тестов для испытания функции, обычно такая же, как сложность разработки соответствующей функции. **Покрытие их тестами** для испытаний и устранения дефектов и обеспечения качества – это последовательное испытание модулей, компонентов или комплексов программ, которые могут определяться посредством анализа соответствия **пространства требований** к функциям и характеристикам программы и **пространства применяющихся тестов** для предварительных испытаний и в процессе сертификации.

Обычно испытатели небольших компонентов, особенно в независимых группах специалистов, фокусируют свои усилия на покрытии их функционирования, а испытатели структуры концентрируются на тестовом покрытии структуры модулей или их групп программ. Создано и доступно для **повторного применения** большое количество автономных программных компонентов и модулей высокого качества с унифицированными интерфейсами, которые можно интегрировать в комплексы программ различного назначения. Иногда номенклатура и характеристики готовых компонентов могут не полностью удовлетворять заказчика или разработчиков программного продукта, и приходится дополнительно разрабатывать и тестировать некоторую часть модулей. Кроме того, при комплексировании и испытаниях комплексов программ могут обнаруживаться дефекты и ошибки компонентов, для устранения которых необходима коррек-

тировка и тестирование модулей, с применением технологии, ориентированной на их особенности. Однако специфика разработки и **тестирования небольших компонентов**, особенно с анализом деталей структуры оказывается **не применимой** для тестирования сложных программных продуктов, вследствие их больших размеров, и **рассматривается** далее в лекции 9. Основное внимание в учебнике сосредоточено на **стратегии, методах и процессах сертификационных испытаний целостных программных комплексов** и их функциональных компонентов, **на соответствие заданным и формализованным требованиям** – рис. 8.1.

Цели, задачи и процессы сертификационных испытаний программных продуктов на соответствие требованиям:

- оценка и повышение вероятности соответствия программного продукта установленным и утвержденным заказчиком требованиям и ограничениям ресурсов;
- анализ и формирование требований к тестам и видам организации испытаний;
- оценка основных лимитирующих ресурсов испытаний – трудозатрат специалистов, и ограничений на сроки производства версии программного продукта;
- исходные данные для планирования производства и испытаний программного продукта;
- утверждение графика проведения тестовых процедур при испытаниях;
- распределение ресурсов для испытаний в зависимости от причин, мест и характеристик выявляемых дефектов и реализуемых изменений в программах и данных;
- оценивание в каком объеме испытания достаточны и когда необходимо завершать процесс тестирования.

Рис. 8.1

Основной целью испытаний является повышение вероятности того, что программный продукт при любых обстоятельствах будет функционировать надлежащим образом и ***соответствовать установленным и утвержденным заказчиком требованиям***. Важнейшая задача тестирования, состоит в достижении основной цели проекта системы, в которой применяется программный продукт.

Тестирование предназначено для обнаружения дефектов и ошибок и последующего их устранения. Продукт будет удовлетворять ожидания заказчика и конечных пользователей, если будет выявлено и устранено как можно больше его дефектов. Кроме того, тестирование позволяет определить характеристики качества, при принятии решения о готовности программного продукта для применения. Автоматизация способствует более быстрому, качественному и эффективному тестированию, что ведет к сокращению объема работ и к улучшению процесса тестирования.

В процессе испытаний должно проверяться, что комплекс программ работает в соответствии со спецификацией требований и **удовлетворяет следующим общим базовым критериям** [4]:

- при допустимых входных данных комплекс программ вырабатывает верный результат, соответствующий требованиям;
- при недопустимых входных данных комплекс программ отвергает входные данные и выдает соответствующее диагностическое сообщение;
- независимо от допустимости входных данных программы не зависают и не завершаются аварийно.

Определив задачи, испытатели должны выработать **стратегии** для их решения. В прошлом испытания обычно проводились в конце жизненного цикла системной разработки, усилия сосредотачивались в основном на тестировании готового программного продукта. Относительно недавно производители программных продуктов пришли к пониманию того, что для достижения наилучших результатов **испытания должны сопутствовать всем этапам жизненного цикла** производства системы, и их нужно начинать на самых ранних этапах разработки комплекса программ.

Тщательное **исследование целей и ограничений** проекта должно приводить к выбору соответствующих стратегий тестирования, которые позволяют получить более предсказуемый, высококачественный результат и обеспечивать высокий уровень автоматизации тестирования. К возможным ограничениям обычно относятся сокращенные сроки выпуска продукта на рынок и недостаточное количество специалистов, участвующих в проекте. Другие ограничения могут быть связаны с применением нового процесса разработки или с внедрением нового инструмента тестирования. Чтобы выработать стратегии тестирования для конкретного проекта, испытатели должны сочетать

тщательное исследование ограничений, оказывающих влияние на выбор методов предотвращения дефектов, с использованием методов обнаружения дефектов и ошибок.

Необходимо с самого начала проекта привлекать испытателей к процессу создания комплекса программ. Особенно важно *участие специалистов по испытаниям в разработке требований к программному продукту*. Проблемы управления, связанные с требованиями, составляют более 40% факторов, необходимых для обеспечения успеха при разработке проекта [3]. На этапе определения требований тестирование должно способствовать созданию ясных и непротиворечивых требований. Участие испытателей на этом этапе нужно еще и для того, чтобы обеспечить формулировку системных и всего комплекса требований в пригодных для тестирования терминах. При определенном исходном состоянии системы и множестве входных параметров испытатели должны иметь возможность предсказать состав и содержание выходных эталонных данных программного продукта и системы.

Тестирование системных требований должно быть неотъемлемой частью построения программного продукта и системы в целом. При этом в основе большей части *проблем, лежат неверные, забытые, неясные или неполные требования*. В последние годы достигнуто более полное понимание того, насколько важно разработать качественные требования. Менеджеры проектов и менеджеры по поставке программных продуктов обычно осознают, что, прежде чем приступить к реализации системы, необходимо ознакомиться со стратегией и методологией тестирования требований.

Создание и применение связанных *профилей международных стандартов* способствует предотвращению многих дефектов. Использование стандартных инструкций пользователями по эксплуатации также облегчает обнаружение дефектов и повышает качество комплексов программ. Работы по тестированию являются взаимозависимыми и требуют от участников проекта коллективных усилий. В свою очередь для эффективной работы в команде необходимо соблюдать ряд правил и стандартов, которые регламентируют правила взаимодействия участников крупных проектов. Стандарты проектирования программных комплексов могут потребовать построения структурных диаграмм, обычно такие стандарты способствуют повышению модульности при создании функциональной независимости компо-

нентов. В некоторых предприятиях группа тестирования является основной при проверке того, следуют ли разработчики, стандартам структурного проектирования и тестирования комплексов программ. Тестирующие должны получать от заказчика перечень рекомендуемых стандартов, в составе документов, сопровождающих требования к программному продукту (см. Приложение).

Анализ и формирование **требований к тестам** проводится при определении целей, задач и стратегий испытаний, а также инструментов тестирования, которые возможно применять – рис. 8.2.

Организация и планирование сертификационных испытаний программных продуктов на соответствие требованиям:

- выбор стратегии испытаний и определение требований к тестам программного продукта;
- верификация тестов и требований к компонентам и комплексу программ;
- обеспечение соответствия тестов требованиям к программному продукту;
- планирование испытаний программного продукта;
- анализ матрицы отслеживания требований к тестам на соответствие требованиям к программному продукту;
- разработка графика выполнения тестов для испытаний программного продукта;
- оценка полноты покрытия тестами требований к программному продукту;
- оценка затрат ресурсов на испытания программного продукта.

Рис. 8.2

Для сложных комплексов программ важно иметь пригодные для подготовки тестирования **системные требования и сценарии использования функций** системы и программного продукта. Эти требования необходимо анализировать и определять в терминах требований к тестам, устанавливающим их содержание и корректность. При анализе требований к тестам необходимо решать следующие **проблемы**:

- **что** нужно сделать при определении требований к тестам для проверки корректности конкретной версии программного продукта или его функциональных компонентов;

- *как* преобразовать архитектуру, системные требования к функциям и характеристикам в сценарии и результаты использования программного продукта, а также в конкретные требования к системе тестов;

- *как* провести анализ и формирование документации на комплекс программ для формулировки и документирования требований к тестам и результатам тестирования.

Требования к тестам должны содержать подробный перечень того, что должно быть протестировано. При разработке требований к тестам специалисты по тестированию должны выполнить ряд шагов, чтобы получить представление о потребностях заказчика. Необходимо также изучить системные требования к программному продукту, описание назначения и сценарии использования системы для того, чтобы понять ее цель. Кроме того, следует определить функции, наиболее *значимые для применения системы, и функции повышенного риска*.

Испытания обычно производятся на протяжении всей разработки и сопровождения комплекса программ *на разных уровнях*: над отдельным модулем, функциональным компонентом или комплексом программ в целом. При этом ни один из уровней тестирования не может считаться приоритетным. Важны все уровни тестирования, вне зависимости от используемых моделей и методологий. Тестовые сценарии могут разрабатываться как для проверки *функциональных* требований и характеристик, так и для оценки *нефункциональных* (архитектурных) требований. При этом, существуют такие тесты, когда количественные параметры и результаты тестов могут лишь качественно удовлетворять цели тестирования, например, простота использования, в большинстве случаев, не может быть явно описана количественными характеристиками. Можно выделить следующие, наиболее распространенные *виды организации испытаний* [8]:

- тестирование, базирующееся *на интуиции и опыте специалистов*, наиболее широко используется и основывается на их знаниях, имевшихся ранее аналогий, оно может быть полезно для идентификации тех дополнительных тестов, которые не охватываются более формализованными методами;

- *функциональное тестирование* соответствия требованиям – проверка комплекса программ и/или системы, предъявляемым к ним

требованиям, описанным на уровне спецификации функций и характеристик;

- **системное тестирование** охватывает целиком всю систему, обычно фокусируется на нефункциональных и/или динамических требованиях – безопасности, производительности, точности, надежности; тестируются интерфейсы к внешним приложениям, аппаратному обеспечению, операционной и внешней среде;

- испытания **граничных значений** строятся с ориентацией на использование тех величин, которые определяют предельные характеристики тестируемых компонентов, расширением этого метода являются **тесты оценки живучести** системы, проводимые с величинами, выходящими за рамки специфицированных пределов значений в требованиях;

- **приемо-сдаточное тестирование** – испытания программного продукта, проверяется поведение системы на предмет удовлетворения требований заказчика, как с привлечением разработчиков системы, так и без них;

- **установочное тестирование** проводится с целью проверки процедур инсталляции программного продукта и/или системы в целевом окружении конкретного применения;

- **тестирование удобства и простоты применения** – проверка, насколько легко конечный пользователь может освоить программный продукт, включая не только функциональную составляющую, но и документацию;

- **тестирование, ориентированное на дефекты** – на конкретные, специфические категории ошибок, на обнаружение наиболее вероятных дефектов, предсказываемых, например, в результате анализа рисков.

Поскольку в сложных программных продуктах подвергнуть испытаниям абсолютно все невозможно, **важно организованно выбирать, что обязательно** нужно протестировать (см. рис. 8.2). Если допустить полный перебор сценариев в тестировании, тестовое покрытие будет избыточным, и для отладки программного продукта потребуется значительное время, что может поставить под угрозу срок сдачи проекта. Если тестирование окажется недостаточным, то увеличится риск пропуска негативного эффекта, устранение которого будет стоить дорого, особенно после сдачи программного продукта в эксплуатацию. Отыскать **нужный баланс** между этими крайностями

помогает опыт и оценки успешности тестирования реализованных проектов.

Испытатели должны придерживаться утвержденного *графика проведения тестовых процедур*. По окончании выполнения теста следует производить анализ выходных данных тестирования и готовить документацию по результатам тестирования. Организация и планы комплексного, системного тестирования и приемо-сдаточных испытаний в совокупности представляют собой этапы, необходимые для тестирования системы в целом. Комплексное тестирование версии программного продукта направлено на проверку его полного функционирования. В процессе комплексного тестирования функциональные компоненты интегрируются и тестируются совместно на основе управляющей логики системы. Поскольку одни компоненты могут состоять из других, то часть комплексного тестирования, может проводиться в ходе иерархического тестирования компонентов. В процессе системного тестирования испытатель проверяет интеграцию отдельных частей, в совокупности составляющих систему в целом. Тестирование на системном уровне обычно проводится отдельной группой испытателей.

На основании *оценки рисков* целесообразно группировать дополнительные требования к тестам, чтобы уменьшить влияние функций повышенного риска при применении системы. Функции, относящиеся к техническим ресурсам и к редким пользователям, а также практически не используемые функции могут ранжироваться как низкоприоритетные. Некоторые требования к модификациям и тестам могут быть ранжированы довольно высоко в списке приоритетов, поскольку их часто применяют, либо конечные пользователи практически не имеют информации и квалификации в этой области использования программного продукта. Некоторые требования к тестам жизненно важны для пользователей. Если при тестировании не повысить внимание к этим требованиям, могут быть нарушены договорные обязательства или предприятие понесет финансовые потери.

При разработке сложных комплексов программ для верификации и тестирования требуются *значительные ресурсы* в течение всего ЖЦ, и наиболее критичным ресурсом является *допустимое время* поэтапного выполнения этих процедур. Выделение и упорядочение компонентов и процедур для тестирования изменений в крупных ПС зависит от их архитектуры и реального состава, готовых к использо-

ванию компонентов. При систематическом **восходящем тестировании**, прежде всего, проверяются программные компоненты нижних иерархических уровней в функциональной группе программ, к которым последовательно подключаются вызывающие их компоненты. Последовательное наращивание компонентов программ **снизу вверх** позволяет проверять работоспособность таких компонентов в их естественном исполнении, без подмены и имитации компонентов нижних уровней. При восходящем тестировании главная задача – обеспечить укрупнение, **интеграцию** и корректное взаимодействие всех компонентов для полного решения функциональных задач комплексом программ.

Если **изменения в программе или в данных невелики**, то испытатели обычно стремятся ограничиться компонентами, непосредственно связанными с выполненной корректировкой. Для этого выделяются для тестирования компоненты, которые связаны по информации или по управлению с теми, которые подверглись даже малым изменениям. Однако следует учитывать, что корректировки программных компонентов в 10 – 30% случаев сами могут содержать ошибки и требуют тщательного тестирования не только тех частей, где внесены изменения.

Тестирование разработанных компонентов комплекса программ необходимо для проверки при корректности внесения изменений в версии программного продукта. При этом возможно **предсказание** мест в программе, где наиболее вероятны при корректировке вновь внесенные дефекты и ошибки, и каких типов они могут быть. Поэтому не всегда необходимо каждую новую версию программного продукта подвергать столь же широкому тестированию, как первую или предшествующую версию. Тестирование необходимо сосредоточивать на компонентах и функциях впервые вводимых или значительно модифицируемых в данной версии. Таким образом, тестирование должно приобретать управляемый, регулируемый характер.

Организация работ по тестированию может руководствоваться различными соображениями и критериями – от управления рисками до специфицированных сценариев контроля функций программных систем. В любом случае, желательно, исходя из ресурсов, количественных оценок и других характеристик, обеспечить использование различных методов тестирования для многосторонней оценки и улучшения качества получаемого продукта. Работы по испытаниям,

ведущиеся на разных уровнях, должны быть организованы в единый (однозначно интерпретируемый) процесс, на основе учета элементов и связанных с ними факторов: людей (в том числе, в контексте организационной структуры предприятия), инструментов, регламентов и количественных оценок (измерений).

Эффективность испытаний может быть достигнута в том случае, если ясно, какие типы дефектов могут быть найдены в программах системы и как изменяется их частота во времени (подразумеваемая историческую перспективу развития качества системы). Эта информация позволяет прогнозировать качество программного продукта и системы, и помогает совершенствовать процесс разработки, в целом. Тестируемая программа может оцениваться на основе подсчета и классификации найденных дефектов. Для каждого класса дефектов можно определить отношение между количеством соответствующих дефектов и размером комплекса программ.

Стратегии и измерения эффективности испытаний должны быть объединены в единый процесс, как **деятельности по обеспечению качества**. Хотя, в большинстве современных методов разработки, тестирование выходит на передний план и является одной из базовых практик, многостороннее тестирование и прогнозирование на основе полученных результатов, часто подменяется отдельными работами в этой области, не позволяющими добиться необходимого качества. Только в том случае, если испытания рассматривать как один из важных процессов всей деятельности по созданию и поддержке программного продукта, можно добиться оценки качества и стоимости соответствующих работ и, в конце концов, соблюсти те ограничения, которые определены для проекта заказчиком.

Очень важным аспектом организации испытаний является решение о том, в каком объеме они достаточны и **когда необходимо завершить процесс тестирования**. Тщательные измерения, такие как достигнутое покрытие тестами или охват функциональности, безусловно, очень полезны. Однако, они не могут определить критериев достаточности тестирования. Принятие решения об окончании тестирования, включает рассмотрение стоимости и рисков, связанных с потенциальными сбоями и нарушениями надежности функционирования тестируемой программной системы. В то же время, стоимость самого тестирования также является одним из ограничений, на основе которых принимается решение о продолжении тех или иных свя-

занных с проектом работ (с частности, тестирования) или об их прекращении.

Соответствие пространств требований и тестов к функциям и характеристикам комплексов программ

Как отмечалось выше, основная цель *верификации комплекса программ и компонентов* состоит в том, чтобы обнаружить, зарегистрировать и устранить дефекты и ошибки, которые внесены во время последовательной разработки или модификации требований к программным компонентам разных уровней. Для обеспечения эффективности затрат ресурсов при ее реализации, верификация должна быть интегрирована как можно раньше с основными процессами проектирования и сопровождения в жизненном цикле комплекса программ. Она должна проводиться, начиная от общих требований, заданных в техническом задании и/или в спецификации на всю систему, до детальных требований на программные компоненты, их взаимодействие и тесты (рис. 8.3).

Цели верификации достигаются посредством последовательного выполнения комбинации из просмотров, анализов, разработки контрольных сценариев и процедур, и последующего их выполнения. Такие сценарии тестов предназначены для поэтапной проверки внутренней непротиворечивости и полноты реализации требований. Однако этот процесс не гарантирует полноту и корректность всех требований, так как обычно эти работы проводятся частично вручную и могут отсутствовать четкие эталоны для их проверки.

Для обеспечения высокого качества программного продукта параллельно с *верификацией требований* и разработкой корректировок, следует разрабатывать и верифицировать *спецификации и сценарии тестов*, отражающие методы и конкретные процедуры проверки реализации этих требований. Тестовые спецификации могут использоваться для проверки согласованности, внутренней непротиворечивости и полноты реализации требований без исполнения программ. Для каждого требования к функциям комплекса программ, его архитектуре, функциональным компонентам должна быть разработана спецификация требований к тестам, обеспечивающая проверку корректности, адекватности и возможности в последующем реализовать тестирование компонента на соответствие этому требованию.

Такая взаимная проверка корректировок функций компонентов, отраженных в требованиях и в спецификациях тестов, обеспечивает повышение их качества, сокращение дефектов, ошибок, неоднозначностей и противоречий.

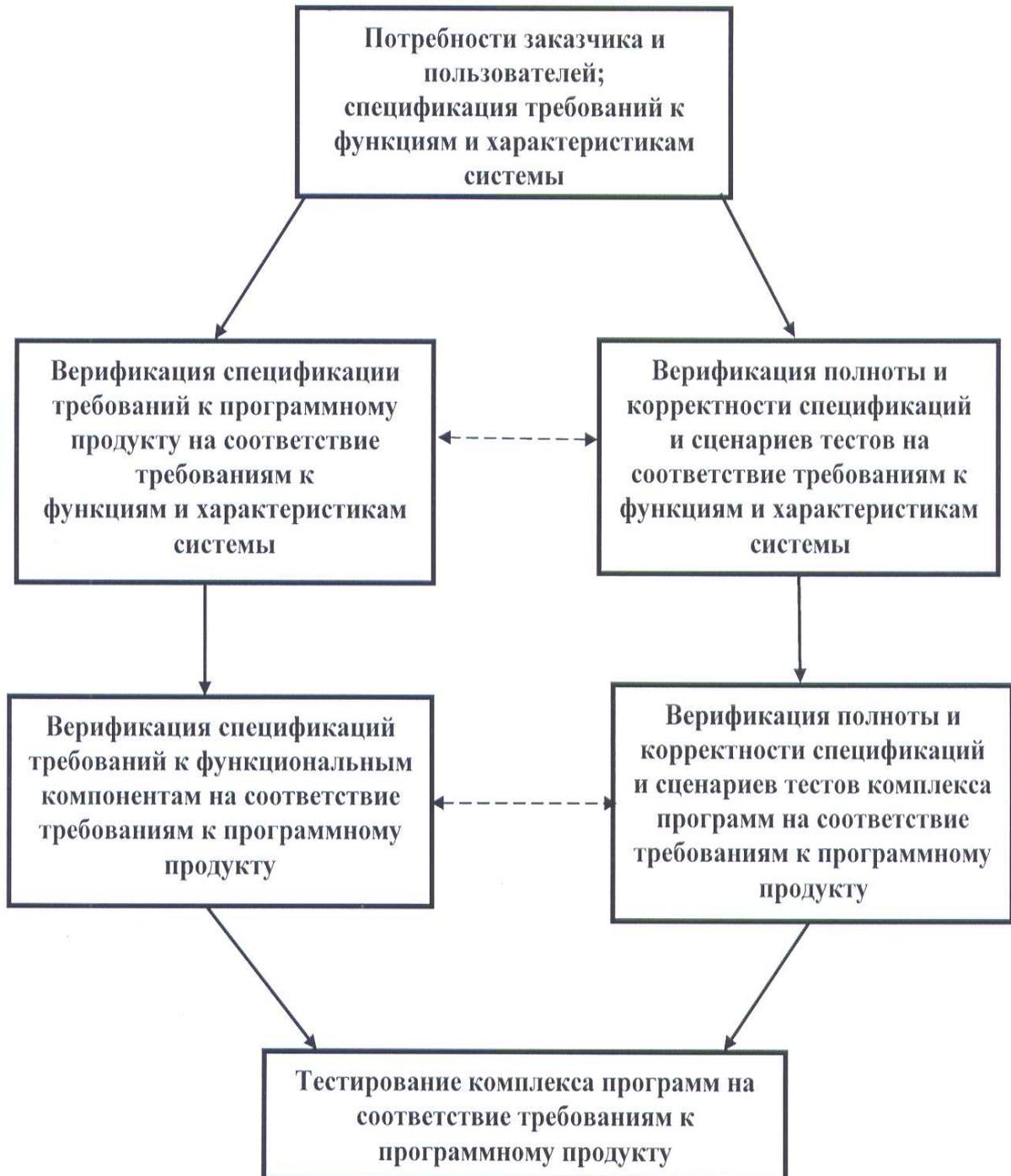


Рис. 8.3

Спецификации тестов должны обеспечивать дополнительный контроль корректности спецификаций требований и верификацию

взаимодействия компонентов на соответствующем уровне описания комплекса программ. Независимая разработка спецификаций тестов на основе спецификаций требований, создает базу для обнаружения, какие требования не тестировались или принципиально не могут быть проверены тестированием. Таким образом, **верификация спецификаций требований тестов** к функциям и характеристикам программных компонентов и комплекса могут использоваться с **двумя целями**:

- для разработки и проверки программ и интерфейсов взаимодействия программных компонентов разных уровней в комплексе программ;
- для создания требований к скоординированному комплексу тестов для проверки совокупности компонентов, обеспечивающих взаимную проверку реализации спецификаций требований комплексом программ.

В результате совокупности **спецификаций требований к тестам** могут применяться **как эталоны и вторая адекватная форма описания содержания комплекса программ** для сквозной верификации спецификаций требований к тестам сверху вниз, а также сами подвергаться верификации на корректность соответствия исходным требованиям к компонентам текстов программ разного уровня (см. рис. 8.3).

Такие **параллельные взаимные проверки** спецификаций требований и текстов программ, и спецификаций тестов способствуют выявлению и исключению множества вторичных дефектов и ошибок комплексов программ. Впоследствии, эти спецификации тестов должны использоваться для непосредственного тестирования исполнения требований к программным компонентам соответствующего уровня. Кроме того, параллельная и независимая разработка, с одной стороны, спецификаций программ и спецификаций тестов, а также их реализации, с другой стороны, позволяет распараллеливать работы, что ведет к сокращению сроков создания компонентов и комплексов программ.

Реализация этих целей верификации и тестирования может производиться разными методами и независимыми специалистами – программистами, интеграторами и испытателями, что позволяет использовать результаты их деятельности для сравнения одних и тех же описаний программ, представленных на языках программирования и

на языках тестов. Особенности описаний и реализации программ, а также *мышления их разработчиков (программистов)* – на основе требований, функций, характеристик структуры и исполнения программ, существенно *отличаются от представлений и методов описаний тех же функций комплекса программ тестирующими* – создателями сценариев и эталонов требуемых результатов тестирования.

Они акцентируют свою деятельность на конкретных процедурах проверки функционирования, возможных результатах и взаимодействии компонентов комплекса программ. Это позволяет выявлять вторичные дефекты, и повышать качество путем *сопоставления двух методов* и результатов описания *одних и тех же* функций и характеристик программ, за счет того, что мала вероятность одинаковых ошибок в сценариях и реализациях тестов и в описаниях требований к функциям и характеристикам программ.

Испытания на базе верифицированных требований к тестам должно обеспечивать качество комплексов программ. Такие сценарии предназначены для поэтапной проверки внутренней непротиворечивости и полноты реализации требований к функциям и характеристикам качества комплексов программ. Требования к тестам должны проверяться на корректность системных взаимосвязей функциональных компонентов *сверху вниз* и на внутреннюю корректность взаимодействия между требованиями для компонентов и их пригодности для последующего тестирования и применения (рис. 8.4). Последовательная разработка и верификация спецификаций требований к программным компонентам должна обеспечивать корректность их логических и функциональных связей и применения. Однако этот процесс *не гарантирует полноту и корректность реализации всех требований*, так как обычно эти работы проводятся частично вручную и могут отсутствовать четкие эталоны для их проверки.

Соотношение между системными требованиями к функциям и характеристикам и требованиями к тестам системного уровня может варьироваться, причем каждому требованию к характеристикам может соответствовать одно или несколько требований к тестам, в зависимости от степени риска и детализации требований.

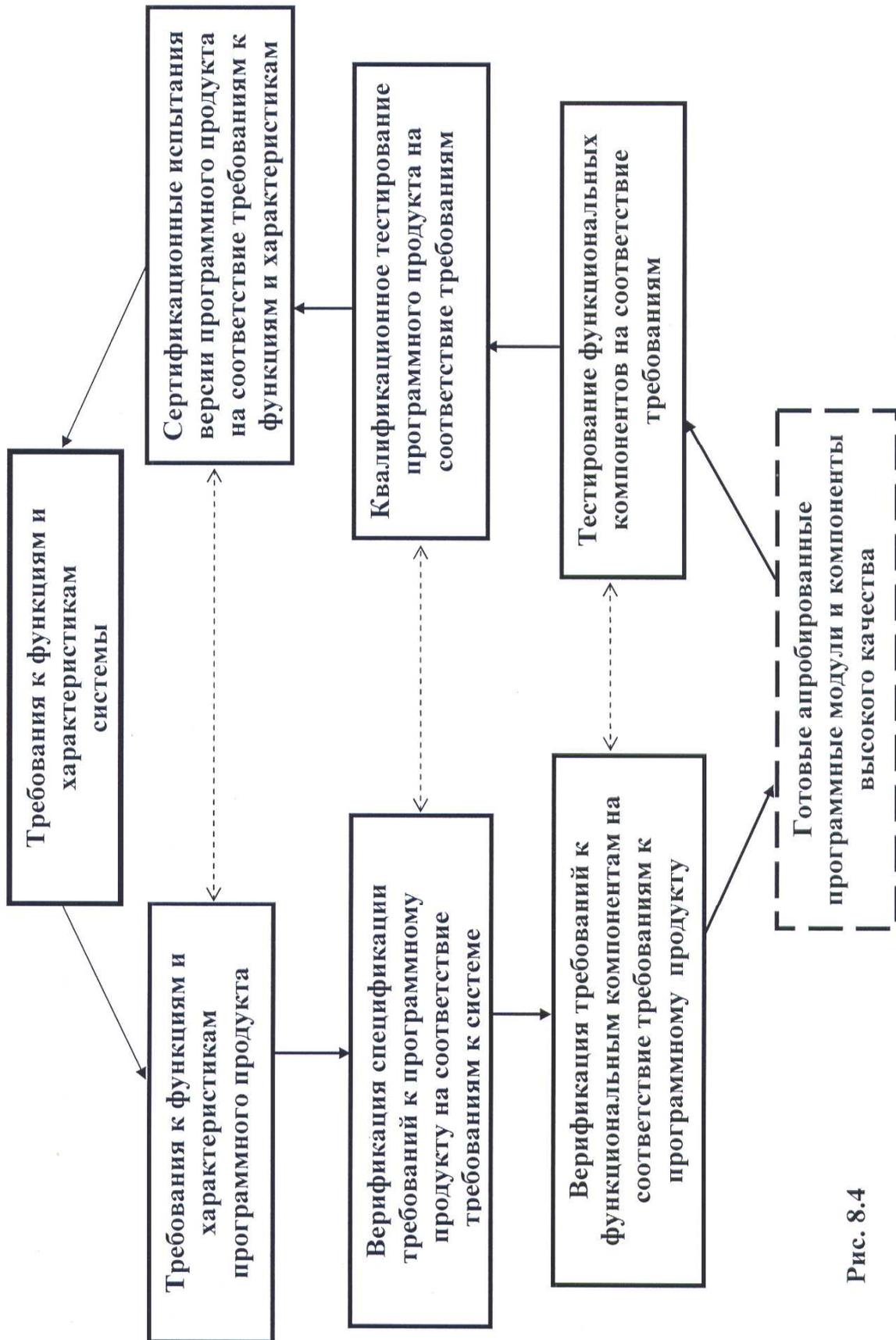


Рис. 8.4

Разрабатывая требования к тестам на основе требований или сценариев использования системы, специалисты по тестированию должны создавать, по крайней мере, **одно требование к тестам на каждое системное требование к характеристикам программного продукта**.

Системные требования к функциям или сценарии использования системы, разбитые на составные части на уровне спецификации функций и характеристик программного продукта или архитектуры, обычно проверяются во время комплексного тестирования или приемо-сдаточных испытаний.

Матрица отслеживания требований тестами позволяет проследить системные требования и сценарии использования системы, а также покрытие требований тестовыми процедурами. Матрица может принимать одну из нескольких форм, основанных на различных видах отображений. Матрица отслеживания требований должна определять каждое требование, которое проверяется испытателями, а также метод его верификации. Матрица отображает тестовые процедуры на системные требования или сценарии использования системы, и помогает убедиться в том, что системные требования или сценарии использования системы, проверенные при тестировании, успешно реализованы.

Тестовое покрытие – это отображение пространства варьирования тестов на некоторую совокупность требований к функциям и характеристикам компонентов или комплекса программ, предназначенных для тестирования. Покрытие можно отражать **с помощью матрицы**, где по одной оси представлены тестовые сценарии, а по другой требуемые функции, характеристики, безопасность или риски компонентов или комплекса программ [5,6]. При этом можно отслеживать степень покрытия требований, поддерживаемых конфигурацией и сценариями использования программного продукта. Такие документы являются статическими графовыми моделями функционирования систем. Подобные модели могут фиксировать потоки и виды данных системы, последовательность состояний, в которых может находиться система и последовательность их обработки. Можно проводить оценку степени покрытия относительно этих моделей, проверяя, что каждому узлу и каждой связи в графе соответствует тестовый сценарий. Оценка покрытия требует учета, в какой степени тесты нацелены на пространство требуемых функций и характеристик ком-

плекса программ. Точность определения степени покрытия ограничена, но выделение **сильных и слабых связей между тестами и пространством требований** к функциям и характеристикам проекта позволяет акцентировать анализ на безопасности, допустимых рисках или некоторых характеристиках, важных для конкретного проекта.

Таким образом, совокупности требований к тестам могут применяться **как эталоны и вторая адекватная форма описания функций и характеристик комплексов программ**. Для обеспечения высокого качества программных продуктов стратегию создания **требований к тестам** целесообразно строить, **основываясь на требованиях к функциям и характеристикам** комплексов программ. Требования к тестам можно дополнять на основе анализа логики архитектуры системы структурным методом. Он может применяться в зависимости от условий договора или особых требований к безопасности или ограниченным рискам системы. Например, покрытие полной структуры решений для некоторых функций может быть необходимо в аэрокосмических и других критических системах с повышенными требованиями безопасности.

Стратегии и планирование испытаний программных продуктов

Стратегия испытаний – это совокупность выбранных методов и решений, вытекающих из целей и задач проекта и его тестирования, общие правила и принципы, способствующие достижению целей разработки программного продукта высокого качества. Ведущий испытатель или тест-менеджер, готовящий план, должен тщательно обдумывать идеи, выводы и решения таким образом, чтобы специалисты, использующие план, четко понимали, что от них требуется: стратегию тестирования, планирование и использование ресурсов, управление конфигурацией проекта и минимизацию рисков (рис. 8.5). При формировании стратегии испытаний целесообразно использовать **следующие рекомендации** [8].

Тестирование в первую очередь следует проводить для требований с наивысшим приоритетом, которые представляют для заказчика наибольшую важность, либо которые могут причинить заказчику наибольшие неприятности в случае проявления дефектов программного продукта. Если запланировано тестирование всех требова-

ний, и ресурсы это позволяют, естественно, проверить выполнение всех требований. В случае недостаточных ресурсов, перед предъявлением продукта заказчику необходимо тщательно протестировать требования с наивысшими приоритетами. Целесообразно получить согласие заказчика на то, что требования, которые были проверены частично или не проверены вообще, не будут использоваться и поддерживаться вплоть до следующей версии продукта.

Стратегии испытаний программных продуктов:

- стратегия испытаний – совокупность методов и решений, способствующих достижению программным продуктом высокого качества;
- испытания в первую очередь следует проводить для требований с наивысшим приоритетом, которые наиболее важны, либо могут причинить наибольшие неприятности;
- сначала испытания следует осуществлять для новых функциональных возможностей, с целью исправления или совершенствования функций и характеристик продукта;
- тестирование полезно начинать с функций и конфигураций, с которыми наиболее часто будет иметь дело конечный пользователь или система;
- приоритет при планировании испытаний целесообразно отдавать:
 - для покрытия критических вариантов использования системы и пользовательских данных;
 - областями повышенного риска наличия дефектов;
 - для покрытия критических пробелов в тестовом покрытии;
 - функциям, наиболее значимым для работы системы, и функциям повышенного риска;
- управление требованиями к тестам должно включать хранение требований, отслеживание связей, оценку рисков требований к тестам.

Рис. 8.5

Тестирование сначала осуществлять для новых функциональных возможностей, которые изменялись с целью исправления или совершенствования функций и характеристик. Если версия является очередным обновлением или эксплуатируемой версией, особое внимание следует уделить новым функциям и характеристикам. Любые изменения, внесенные в программы, могут исказить даже те час-

ти комплекса программ, которые непосредственно не затрагиваются изменениями. В этом случае лучше всего как можно шире следует выполнять регрессионные тесты для всех функций и характеристик комплекса программ, какими бы ни были изменения.

Тестирование следует проводить тех компонентов и участков, в которых наиболее вероятно присутствие дефектов. Если обнаружен какой-то дефект, часто рядом может быть еще несколько аналогичных дефектов. Если есть сведения от разработчиков, что часть компонентов породили проблемы во время модульного тестирования или проверки взаимодействия и функционирования компонентов, такие участки необходимо отметить, чтобы обратить на них внимание при комплексном тестировании или системных испытаниях.

Тестирование желательно начинать с функций и конфигураций, с которыми наиболее часто будет иметь дело конечный пользователь или система. Для этого в спецификации требований полезно включать методику оценки случаев использования некоторых функций и иметь доступ к функциональному разрезу конечного пользователя – математическому ожиданию использования каждой функции. Это дополнительный источник информации для установки приоритетов тестирования. Большая часть времени, отведенного на тестирование, должна затрачиваться на проверку наиболее часто используемых функций, конфигураций и операций.

Стратегия испытаний функций и характеристик комплекса программ должна содержать последовательные процессы:

- разработку автоматизированных и/или ручных тестов для покрытия всех функций, характеристик и рисков качества, которые определены как нуждающиеся в полном или сбалансированном **приоритетном тестировании**, с особым вниманием к факторам функционирования системы, наблюдаемым со стороны пользовательского интерфейса или доступного программного интерфейса системы;
- дополнять тестовые данные или условия тестовых сценариев для покрытия **критических вариантов** использования системы, пользовательских данных или сокращения известных дефектов в программном продукте;
- использовать исследовательское тестирование в областях, которые ранее не были покрыты тестами, и которые, с точки зрения интуиции или результатов тестирования, представляются областями

повышенного риска наличия дефектов;

- насколько позволяют время и ресурсы или вызывает сомнение о возможных не выявленных рисках, использовать методы структурного покрытия для выявления не подвергавшихся тестированию областей и затем дополнять тесты для **покрытия критических пробелов** в тестовом покрытии [5, 6].

При разработке требований к тестам группа тестирования должна выполнить несколько предварительных шагов, в том числе получить четкое **представление о потребностях заказчика**. Необходимо также изучить системные требования, сценарии использования системы и/или описание назначения системы для того, чтобы лучше понять цель ее разработки. Еще один шаг — определение **функций, наиболее значимых** для работы системы, и **функций повышенного риска**. Определяются также инструменты тестирования, которые будут применяться для выполнения проекта. Требования к тестам для комплексов программ можно также получить на основе представлений о логике архитектуры системы. Разрабатывая требования к тестам на основе системных требований или сценариев использования системы, группа тестирования должна создавать, по крайней мере, **одно требование к тестам на каждое системное требование**.

Управление требованиями к тестам включает в себя хранение требований, отслеживание связей, оценку рисков требований к тестам, выстраивание последовательности требований к тестам и определение методов верификации тестов. Отслеживание связей предполагает отображение тестовых процедур на требования к тестам и дефектов на тестовые процедуры. Группа тестирования должна описать способ управления требованиями к тестам в плане тестирования.

Заказчик и разработчики должны анализировать и, в конечном счете, **утверждать план испытаний** комплекса программ, в котором описаны требования к тестированию и представлена **матрица соответствия системных требований и тестовых сценариев** [6]. Матрица соответствия должна содержать информацию о требованиях, а также показывать взаимосвязь между требованиями и другими результатами проекта. Планирование испытаний включает, как определение требований к тестам, так и разработку процессов управления этими требованиями. Оно предполагает, что процессы тестирования, методы, методики, персонал, инструменты, план-график и оборудование организованы и эффективно применяются. Когда план испыта-

ний комплекса программ разработан, обновлен и полностью описывает процессы тестирования, он становится руководящим инструментом для выполнения **Программы испытаний**. Он уточняется посредством корректировок целей, задач и стратегий тестирования, а также изменений требований к тестам, фиксирует параметры Программы испытаний, которые должны быть документированы.

Планирование работ по испытаниям должно учитывать ресурсы и работы, которые необходимо выполнить, чтобы своевременно **подготовить тестовую среду**. Испытатели должны определить требования к аппаратному, программному и сетевому обеспечению с целью создания и поддержки адекватных изменений тестовой среды. Нужно спланировать работы по приобретению, установке и настройке компонентов, моделей или динамических генераторов тестовой среды. Заказчик должен **утвердить стратегию испытаний** и тестовые процедуры, которые должны быть подробно описаны в плане тестирования, и определять какие сценарии и тесты когда будут выполняться.

План испытаний должен определять объем работ по испытаниям. Обычно выстраивают **структуру работ**, в которой на одном уровне определяются категории работ по тестированию, а на другом уровне – подробные описания работ. Структура детализации работ используется в сочетании с хронометражем для определения времени выполнения каждого из этапов испытаний. Кроме того, план тестирования должен отражать **оценки затрат** на тестирование. Оценка затрат может определять число сотрудников группы тестирования в проекте в часах или в количестве людей, если на выполнение определенного объема работ выделяется конкретный срок.

Должны быть документированы **квалификация и навыки сотрудников**, необходимых для проведения испытаний [7]. Состав группы испытателей с требуемыми навыкам может быть обозначен в требованиях к их знаниям. Тест-менеджер должен оценить разницу между требуемой квалификацией и реальной подготовкой персонала, чтобы определить возможные **направления обучения**. Также необходимо определять и документировать в плане роли и **ответственность сотрудников** группы испытаний с учетом особенности продукта.

Любая Программа испытаний должна иметь определенные рамки, отражающие ограничения по сотрудникам, по человеко-часам и

по графику. В плане тестирования также следует отражать допущения, предварительные условия и риски тестирования. Сюда включаются все события, действия или обстоятельства, которые могут помешать выполнению испытаний в срок. С разработкой плана должно быть связано определение функций, разработка которых имеет большое значение для успеха проекта, и функций, разработка которых связана с наибольшим риском. Определение наивысшего риска дает возможность группе испытателей сосредоточить усилия на функциях высокой значимости с точки зрения достоверности результатов тестирования.

Требования к тестам должны заноситься в *базу данных и/или матрицу отслеживания требований*. В базе данных или в матрице каждому требованию к тестам сопоставляется идентификационный номер компонента системной архитектуры программного продукта. Затем компонент архитектуры изучается вплоть до детализированных требований к программным компонентам и до системных требований или сценариев использования системы. После определения требований к тестам группа тестирования должна принять предварительное решение по *методам тестирования*, которые наилучшим образом будут проверять каждое требование.

Необходимо, чтобы испытатели как можно раньше получили информацию о матрице отслеживания требований от конечных пользователей или заказчиков системы. Это способствует достижению согласия по поводу методов верификации, обеспечивающих проверку или контроль каждого требования. Принятие этого решения особенно важно, поскольку методы верификации отличаются сложностью и затратами времени. Раннее получение информации по матрице от заказчика позволяет группе тестирования увеличить время реакции на возможные ее изменения. Поскольку матрица отслеживания требований определяет выполняемые тестовые процедуры, одобрение матрицы заказчиком также означает его удовлетворение степенью покрытия тестами системных требований или сценариев использования системы.

В плане испытаний должны быть определены требования к аппаратному, сетевому и программному продукту, что позволит создать тестовую среду, являющуюся зеркальным отражением среды применения программного продукта, предназначенного для испытаний. Приобретение, установка и настройка различных компонентов тесто-

вой среды должно тщательно планироваться. При составлении плана испытаний определяют требования к тестовым данным и средства для их получения, генерации или разработки. План тестирования должен указывать механизм **управления целостностью тестовой модели**, такой как обновление тестовой базы данных до начального состояния базовой версии для возможности поддержки регрессионного тестирования.

К ключевым элементам планирования испытаний относится **тестирование эксплуатационной и технологической документации**. Группа тестирования должна исчерпывающе документировать планы Программы испытаний, а испытатели обязаны подробно изучить содержание этих планов. Эти специалисты должны получить одобрение плана тестирования у конечного пользователя или заказчика. Заказчик должен утвердить стратегию тестирования и тестовые процедуры, которые подробно описаны в плане тестирования и определяют, какие и когда тесты будут выполняться. Кроме того, предполагается, что заказчик согласен с тем, что план тестирования и связанные с ним тестовые сценарии правильно проверяют удовлетворительное покрытие тестами системных требований или сценариев использования системы.

Разработка тестов включает создание испытателями, сопровождаемых, многократно применяемых, простых и надежных тестовых процедур, что может потребовать не меньше усилий, чем разработка программистами тестируемых текстов программ. Чтобы добиться максимального эффекта от автоматизации тестирования, испытатели должны вести разработку тестов параллельно с созданием программистами текстов программного продукта. Схема структуры комплекса программ является графическим представлением основных работ, которые должны быть выполнены во время разработки тестов.

Анализ полноты покрытия тестами требований при исполнении программного продукта должен определять, какие требования не были протестированы и какие части структуры программного продукта не были исполнены при испытаниях. Тестирование структурного покрытия должно устанавливать, не пропущены ли элементы структуры программы, которые не проверены тестовыми процедурами и покрыли ли тесты всю структуру комплекса программ. Анализ покрытия версии программного продукта тестовыми

данными, основанными на требованиях, должен определить, насколько полно тестирование проверило реализацию всех требований, и показать потребность в дополнительных тестовых сценариях. Поэтому должен выполняться анализ полноты структурного покрытия и проводиться его верификация. При тестировании реализации требований к функциям и характеристикам программных компонентов полнота их покрытия тестами редко достигает 90% и хорошо, если составляет около 80% [5, 6].

При проведении испытаний на основе требований к программному продукту, анализа технического задания и совокупности требований к функциям и характеристикам целесообразно оценивать представление о том, **сколько нужно тестов для полного испытания версии программного продукта**. Если составлять тесты, руководствуясь этим принципом, можно оценивать, какое время потребуется на разработку, исполнение и анализ полного комплекта тестов при создании очередной версии программного продукта.

Испытатели должны составлять **план-график разработки и выполнения тестовых процедур** по шкале времени реализации проекта, как средство определения временных рамок для разработки и выполнения различных тестов (графики Ганта). План-график определяет зависимости между тестами и общие сценарии, которые будут постоянно использоваться при испытаниях. Перед созданием полного набора тестовых процедур для очередной версии программного продукта испытатели должны провести **анализ связей между компонентами**. Результаты этого анализа помогут выявить независимые компоненты, спланировать зависимости между тестами и выделить общие сценарии, которые могут повторно применяться в процессе тестирования последующих версий. Для этого строится матрица связей, которая показывает взаимозависимость различных тестовых сценариев. Графическое представление помогает испытателям определять возможности многократного применения сценариев в различных комбинациях с использованием различных оболочек, что позволяет свести к минимуму объем работ по созданию и сопровождению версий тестовых сценариев. В ходе разработки тестовых процедур группа тестирования должна проводить **конфигурационное управление и контроль** для тестовых сценариев и тестовых данных, а также для каждой отдельной тестовой процедуры.

Испытатели должны определить время и ресурсы на эти работы. **При подготовке графика:**

- назначаются сотрудники, ответственные за выполнение каждой из работ по испытаниям;
- следует учитывать последовательность разработки тестов, их взаимозависимость и связь с процессами и компонентами ЖЦ комплекса программ;
- сокращаются или исключаются возможные конфликты между тестовыми процедурами, которые должны быть документированы;
- тестовые процедуры могут быть объединены в группы в соответствии с функциями программного продукта;
- разработка тестовых процедур и их применение должны быть спланированы таким образом, чтобы исключить дублирование работ;
- в структуре тестовых процедур следует учитывать и документировать их приоритеты и риски.

План-график должен помогать **определить, кто из сотрудников отвечает** за выполнение конкретного вида работ. Составив детальный план-график разработки и выполнения тестовых процедур, группа испытателей сможет избежать дублирования работ. Модель связей компонентов и тестовых процедур занимает важное место в разработке плана-графика. **Необходимо описать установку и настройку тестовых процедур**, чтобы убедиться, что проведение этих работ соответствует стандартам управления конфигурацией. Также необходимо описать **последовательность** выполнения процедур и их **взаимозависимость**, поскольку при испытаниях определенная функция не может быть выполнена, пока предыдущая функция не сформирует нужные данные. График разработки и выполнения тестовых процедур должен включать в себя тестирование, проверяющее различные циклы обработки, относящиеся к комплексу программ. Необходимо спланировать работы по настройке среды, тестированию подготовки обязательных отчетов в график разработки и выполнения тестовых процедур. График разработки и выполнения тестовых процедур должен содержать информацию обо всех тестовых процедурах, которые могут породить конфликты.

При группировании автоматизированных тестовых процедур необходимо исключать их дублирование. Нужно изучать

Программу и план испытаний и тестовые сценарии с целью проверки того, как результаты анализа и проектирования тестов учтены в плане-графике. При составлении графика разработки и выполнения тестовых процедур необходимо учитывать **приоритеты** и **риски**, присвоенные различным тестам. График тестирования должен уделять большее внимание проверке функций, наиболее значимых для программного продукта, и функций повышенного риска. Такие тесты должны выполняться в первую очередь, и график обязан предусматривать достаточно времени для проверки этих функций и, в случае необходимости, для регрессионного тестирования.

Группа испытателей должна отслеживать изменения требований и тестов и соответственно **обновлять график испытаний**. Имея в виду многообразие возможных изменений, требующих уточнения плана-графика, полезно создавать базовую версию плана-графика и версии при каждом последующем его принципиальном изменении. Нужно документально фиксировать каждое изменение плана-графика с помощью систем отслеживания графиков.

Оценки затрат на испытания программных продуктов

Прогнозирование затрат на испытания комплексов программ возможно на основе обобщения статистических данных ряда предшествующих проектов [4, 18]. В данном случае, задача ограничена только ориентировочным перечнем основных составляющих затрат, которые целесообразно учитывать в процессе испытаний. Этот перечень может использоваться как **ориентир** при подготовке Программы испытаний. Обычно наиболее важным для реализации проекта и зависящим от большинства его особенностей и факторов является **трудоемкость, непосредственно определяющая стоимость** испытания создаваемого комплекса программ. Значения длительности разработки и числа специалистов взаимосвязаны и в некоторых пределах могут размениваться. Поэтому оценки этих показателей затрат можно варьировать, и при недостаточном числе специалистов естественно возрастает длительность разработки, хотя трудоемкость может остаться практически неизменной. Многократное применение одних и тех же апробированных компонентов и/или многократная адаптация комплекса к различным условиям применения является одним из **перспективных методов повышения качества и снижения затрат труда специалистов** на испытания комплексов программ.

Оценка трудозатрат на испытания является одной из наиболее сложных компонентов обеспечения соответствия требованиям программных продуктов (рис. 8.6).

Оценки затрат на испытания программных продуктов:

- оценки трудозатрат на испытания и обнаружения дефектов версии программного продукта включают следующие этапы:
 - определение перечня и состава задач испытаний, которые должны быть выполнены;
 - оценка трудозатрат на решение отдельных задач и всего процесса испытаний;
 - определение времени, требуемого для решения каждой задачи и длительности всех сертификационных испытаний;
 - затраты на обеспечение требований надежности и безопасности функционирования комплекса программ;
 - построение подробного расписания и поэтапного графика решения каждой тестовой задачи;
 - оценка рисков невыполнения графика работ и формулировка планов их снижения;
- затраты на сопровождение и корректировки дефектов версий программных продуктов включают:
 - затраты на тестирование для обнаружения и устранения ошибок и дефектов в каждой версии программного продукта;
 - оценка затрат на устранение выявленных ошибок при формировании очередной версии;
 - затраты на доработку и совершенствование программ, формирование и испытание новых модернизированных версий;
 - затраты на конфигурационное управление, тиражирование каждой новой версии и ее внедрение в эксплуатируемых и новых системах;
- затраты на завершающие испытания программного продукта в целом.

Рис. 8.6

Затраты труда и времени, необходимых для квалификационного тестирования могут составлять существенную часть стоимости продукта, при этом жизненно важно для успеха, чтобы тестирование проводило достаточное число специалистов и у них было достаточно

времени на качественное выполнение своих задач. Получение *оценки* трудозатрат на выполнение испытаний проекта крупного комплекса программ включает на следующие *этапы*.

Определение перечня и состава задач испытаний, которые должны быть выполнены. Эта оценка начинается с определения работ, которые необходимо выполнить для того, чтобы тестирование программного продукта считалось состоявшимся. На этом этапе может быть достаточным разбить работу на крупные функциональные задачи, отражающие проверку реализации конкретных требований к функциям и характеристикам комплекса программ. Если используются менее формальные методы, то результатом этого этапа может быть простой список основных задач.

Оценка трудозатрат на решение отдельных задач и всего процесса испытаний. Каждая задача, выявленная на первом этапе, требует для своего решения определенных трудозатрат, представляющих собой объем работ, необходимых для выполнения соответствующей задачи. Оценки этих трудозатрат могут быть представлены в виде произведения количества исполнителей на затраченное ими время и измеряться в таких единицах, как человеко-день или человеко-месяц.

При тестировании комплексов программ основным лимитирующим ресурсом обычно являются допустимые трудозатраты специалистов, а также ограничения на сроки разработки версии программного продукта, на параметры ЭВМ и технологию проектирования. Одним из наиболее важных компонентов планирования тестирования является оценка трудоемкости и времени, необходимых для его выполнения. Затраты на тестирование реализации требований версий сложных программных продуктов могут составлять существенную часть стоимости проекта, при этом жизненно важно для успеха этой операции, чтобы тестирование проводило достаточное число специалистов и у них было достаточно времени на качественное выполнение задач по корректировкам комплекса программ. Ограничения реальных ресурсов на верификацию и тестирование определяют достижимое качество версий программных продуктов.

Определение времени, требуемого для решения каждой задачи и длительности всего квалификационного тестирования. Время, необходимое для решения задачи, измеряется в днях, неделях или месяцах. Время, необходимое для выполнения той или иной задачи, зависит от количества исполнителей, однако эта зависимость не обяза-

тельно линейная. Суммарная продолжительность работ по тестированию зависит от продолжительности решения отдельных функциональных задач, однако это не простое суммирование, поскольку некоторые задачи можно решать параллельно и одновременно с другими.

Построение подробного расписания и поэтапного графика решения каждой тестовой задачи. На основе результатов предыдущих этапов можно построить график выполнения работ, возможно, в виде диаграммы Ганта [5, 17] и вычислить сумму наиболее важных трудовых и временных затрат на испытания комплекса программ.

Оценка рисков невыполнения графика работ и формулировка планов их снижения. Следует оценивать возможные проблемы и риски, которые могут возникнуть при решении задач в запланированные промежутки времени, и предусмотреть средства решения этих проблем. Прежде чем приступать к более подробному анализу перечисленных шагов, **важно учитывать, насколько точными могут быть оценки**. После того, как все требования будут проанализированы и утверждены, оценка стоимости продукта обычно отличается от окончательной, фактической стоимости примерно в два раза. При этом общие затраты на тестирование на всех этапах разработки могут достигать 30 – 40% от полной трудоемкости создания комплекса программ. Из них затраты на испытания критических систем совместно с заказчиком могут составлять до половины этих затрат. Ввиду неточностей, свойственных таким процессам оценки, целесообразен мониторинг и периодическое возвращение к оценке трудозатрат и графика выполнения тестирования, и считать это неотъемлемой частью организации работ по испытаниям.

Затраты на обнаружение и устранение дефектов и ошибок в программе определяются **двумя факторами**: затратами на обнаружение каждой ошибки и затратами на устранение выявленных ошибок при формировании очередной версии. Чем меньше ошибок в программе, тем труднее они обнаруживаются, т.е. тем выше затраты на выявление каждой ошибки. Затраты на устранение ошибок и корректировку программ пропорциональны числу дефектов, выявляемых между очередными версиями программного продукта. Непрерывно требуются затраты для контроля состояния версий комплекса программ и обеспечения их сохранности. По опыту работ, широко тиражируемый комплекс программ объемом $\sim 10^5$ строк, может требовать при испытаниях непрерывных усилий коллектива в составе десятка и

более специалистов для устранения ошибок, корректировок версий и документации [7, 18].

Затраты на завершающие испытания программного продукта в целом обычно могут быть достаточно четко выделены из остальных затрат, так как в этих процессах непосредственно участвуют заказчик и пользователи. Величина этих затрат, без учета ресурсов, необходимых для моделирования динамической внешней среды, может составлять около 10 – 15% от общих затрат труда и 10% времени на разработку программного продукта. При этом практически невозможно разделять затраты на оценивание реализации отдельных требований и характеристик.

Возрастание относительных затрат всех видов **при квалификационном тестировании** программного продукта систем реального времени (СРВ) обусловлено высокой трудоемкостью динамической отладки и тестирования в реальном времени. При этом следует учитывать, что абсолютные значения трудоемкости испытаний программ СРВ приблизительно в 5 раз выше, чем программ административных систем (АС) [3, 18]. Относительная трудоемкость и длительность **при испытаниях** различных классов программных продуктов приблизительно одинакова и составляет 8 – 10% от совокупных затрат на разработку. Однако абсолютная трудоемкость для испытаний АС также приблизительно в 5 раз меньше, чем программ СРВ. При этом следует учитывать, что при изменении трудоемкости полной разработки в 5 раз, абсолютная длительность этого процесса сокращается всего в 1,5 – 1,7 раза. В результате полная длительность отладки и испытаний программ АС также приблизительно в полтора раза меньше, чем программы СРВ того же размера.

Затраты на обеспечение требований надежности и безопасности функционирования комплекса программ определяются требуемым их уровнем и сложностью (размером) комплекса программ. При наличии особенно высоких требований к безопасности критических программных продуктов эти затраты могут в 2 – 3 раза превышать затраты на решение базовых, функциональных задач. Для типовых административных систем трудоемкость создания программных средств защиты обычно составляет 20 – 40% затрат на решение основных, функциональных задач. В более простых случаях, доля таких затрат может снижаться до 5 – 10%. Затраты на обеспечение высокой **надежности** близки к затратам на обеспечение безо-

пасности, и, в составе разработки, они могут достигать 2 – 3 кратного увеличения затрат, при требованиях наработки на отказ в десятки тысяч часов. Для минимального обеспечения автоматического рестарта в ординарных системах, затраты составляют порядка 10 – 20%. Однако практически, в любых системах должен присутствовать минимум программных компонентов, обеспечивающих надежность и защиту от преднамеренных и случайных угроз.

Затраты на сопровождение и корректировки дефектов версий программных продуктов можно считать аддитивными и включающими **составляющие** (см. рис. 8.6):

- затраты на тестирование для обнаружения и устранения ошибок и дефектов в каждой версии программного продукта;
- затраты на доработку и совершенствование программ, формирование и испытание новых модернизированных версий;
- затраты на конфигурационное управление, тиражирование каждой новой версии и ее внедрение в эксплуатируемых и новых системах.

Доля каждой составляющей в общих затратах на сопровождение может значительно изменяться в зависимости от особенностей сферы применения и жизненного цикла конкретного программного продукта. Для долгоживущих (~ 10 лет), многократно тиражируемых (1000 – 100000 экземпляров) комплексов доминирующими обычно являются затраты на модернизацию и доработку версий программного продукта.

Затраты на совершенствование и модернизацию комплексов программ близки по содержанию (но не по величине) к затратам на их первичную разработку. Модернизация обычно производится поэтапно. Для каждой новой версии изменяется (разрабатывается) только некоторая часть от всего объема программного продукта. Экспериментально установлено, что эта часть при вводе очередной версии обычно составляет 10 – 20% от объема всего комплекса [4]. Сложность связей компонентов приводит к тому, что удельные затраты на изменяемые программы, при модернизации каждой версии могут быть в 2 – 3 раза больше, чем затраты на создание программ **такого же объема** при разработке первой версии. Эта величина зависит от того, насколько путем стандартизации архитектуры и интерфейсов, предусматривались перспективы совершенствования комплекса программ. Затраты на модернизацию зависят от тиража косвенно, вслед-

ствие расширения условий применения конкретного комплекса и увеличения потока запросов пользователей на развитие программ. Так же косвенно влияет тираж на запросы для устранения выявленных ошибок. Для выполнения этих работ иногда используется коллектив специалистов, осуществивших первичную разработку проекта. Такая организация наиболее характерна для уникальных, заказных программных продуктов. В этих случаях первичную разработку и модернизацию трудно разделить. Для широко тиражируемых программных продуктов, на сопровождение часто выделяется специальный коллектив, не проводивший первичную разработку.

Финансирование испытаний программного продукта целесообразно определять специальным разделом договора между разработчиком и заказчиком на разработку версии программного продукта. В техническом задании и в контракте следует четко определять порядок квалификации видов и причин изменений в программах при испытаниях, а также распределение ответственности за их инициализацию, реализацию и финансирование. Выявленные ошибки в компонентах и комплексе программ, которые искажают реализацию функций, согласованных с заказчиком в контракте и требованиях спецификаций, а также отраженные в документации на версию, должны устраняться за счет разработчика. Модификацию и расширение функций и характеристик компонентов или создание новых версий программного продукта, ранее не отраженных в требованиях технического задания и контракте с заказчиком, следует квалифицировать как дополнительную работу с соответствующим финансированием заказчиком.

После передачи версии программного продукта в эксплуатацию затраты ресурсов на обнаружение и первичную квалификацию дефектов несут в основном непосредственные пользователи. На разработчиков (поставщиков) программного продукта возлагаются затраты на анализ и локализацию источников и причин дефектов и их устранение. Эти затраты зависят от характеристик выявляемых дефектов, от масштаба комплекса программ, организации и технологии его разработки, инструментальной оснащенности сопровождения, квалификации специалистов, а также от тиража и активности применения данного программного продукта. Априори перечисленные факторы прогнозировать невозможно и оценки этой составляющей затрат целесообразно проводить по результатам начальных этапов сопровождения и модификаций первых версий.

Затраты на тиражирование и адаптацию к параметрам среды пользователей зависят от широты распространения программного продукта и могут оцениваться по типовым прецедентам аналогичных проектов или предшествующих версий.

При сопоставлении результатов оценивания функций и характеристик качества с требованиями технического задания и спецификаций, разработчик или поставщик **обязан удовлетворять требования заказчика только в пределах согласованных параметров** модели внешней среды и системы. Оценивание функционирования комплекса программ за этими пределами должно дополнительно согласовываться испытателями с разработчиком. При этом невыполнение требований может квалифицироваться как их расширение за пределы, ограниченные контрактом, и не учитываться при оценивании заказчиком характеристик качества программного продукта, или как дополнительные работы, подлежащие соответствующему финансированию со стороны заказчика, для доработки комплекса программ с целью удовлетворения этих требований.

Лекция 9

ПОДГОТОВКА СЕРТИФИКАЦИОННЫХ ИСПЫТАНИЙ ПРОГРАММНЫХ ПРОДУКТОВ

Требования к квалификации испытателей сложных программных продуктов

Важнейшим фактором при *оценивании ресурсов, необходимых для испытаний программных продуктов*, являются люди – *специалисты*, с их уровнем профессиональной квалификации, а также с многообразием знаний, опыта, стимулов и потребностей. Быстрый рост сложности, повышение требований и ответственности за качество программных продуктов привели к появлению *новых требований к специалистам*, обеспечивающим этапы испытаний в жизненном цикле ПС. Специалисты в соответствии со своей квалификацией и ролью в проекте вносят в него положительный вклад, а также специфические дефекты и ошибки, что целесообразно учитывать при прогнозировании эффективности процессов испытаний – рис. 9.1.

Непрерывно повышаются размеры и сложность комплексов программ, что вызывает возрастание затрат творческого труда на единицу размера программного продукта. Разработка сложных комплексов, особенно, на начальных и завершающих этапах, характеризуется *высокой долей творческого труда*. Дефекты, трудоемкость и длительность отдельных операций и частных работ существенно зависят от индивидуальных особенностей их исполнителей и характеристик конкретного проекта. В перспективе, несмотря на автоматизацию и повышение инструментальной оснащённости технологии разработки программ, доля творческого труда при испытании версий крупных программных продуктов возрастает. Даже при сокращении суммарных затрат на разработку программных компонентов за счет автоматизации нетворческого труда, все более определяющей для технико-экономических показателей испытаний становится доля затрат на творческий труд и *возрастают требования к творческим способностям* испытателей.

Требования к квалификации испытателей сложных программных продуктов:

- испытатели должны обладать аналитическим складом ума, быть внимательным к деталям, организованы, проявлять творче-

Рис. 9.1

Недостатки или отсутствие достоверного обоснования необходимых ресурсов специалистов для испытаний, могут являться причинами острых конфликтов между заказчиками и разработчиками. Поэтому целесообразно **активно привлекать заказчиков** к управлению испытаниями, чтобы обеспечить своевременность разработки версий программных продуктов в условиях ограниченных ресурсов. Уровень **квалификации заказчика** и корректность требований технического задания на функции и характеристики может весьма сильно влиять на выделяемые ресурсы коллективу специалистов для испытаний комплекса программ. При этом первоначальное техническое задание зачастую оказывается недостаточно квалифицированным и подвергается в дальнейшем многократным изменениям. Этому же может способствовать различие между заказчиком и разработчиком в квалификации, уровне понимания целей продукта и необходимых затрат на реализацию дополнительных требований. Особенно сильно

на достоверность технического задания и возрастание затрат может влиять попытка заказчика форсировать сроки разработки версий программного продукта. Даже квалифицированные заказчики вынуждены иногда корректировать техническое задание на любых этапах управления проектом, что может влиять на увеличение затрат на 10 – 20%. Вследствие этого необходима переработка требований и испытания готовых программ, что отражается большим ущербом вследствие дефектов исходных требований заказчика.

При испытаниях высококачественных комплексов программ, прежде всего, необходима организация и **тесное взаимодействие представителей заказчика и разработчиков** продукта. Взгляды и требования заказчика, в основном, отражаются в функциональных и потребительских характеристиках версий программного продукта. Устремления разработчиков направлены на возможность и способы их реализации с требуемым качеством. Эти различия исходных точек зрения на проект приводят к тому, что некоторые неформализованные представления тех и других имеют **зоны неоднозначности и взаимного непонимания** требований, что может приводить к конфликтам. Организация четкого взаимодействия и сокращение этих зон неоднозначностей требует проведения определенных мероприятий и контактов по обмену знаниями, взаимному повышению квалификации и обучению.

При испытаниях комплексов программ коллективами велика роль **квалификации руководителей** разработки, что непосредственно отражается на успехе проекта. **Лидером испытаний** программного продукта может быть: менеджер продукта, менеджер проектирования, руководитель проекта. **Лидер должен** [4, 7, 16]:

- руководить процессом выявления и формирования требований к функциям программного продукта, подлежащим испытаниям;
- рассматривать конфликтующие пожелания, поступающие от различных участников проекта комплекса программ и находить компромиссы, необходимые для определения приоритетов набора функций, представляющих наибольшую ценность для максимального числа участников проекта, прежде всего, заказчика и пользователей;
- вести переговоры с руководством системы, пользователями и разработчиками и поддерживать равновесие между тем, чего хочет заказчик, и тем, что может предоставить команда разработчиков и испытателей за ресурсы и время, отведенные для их реализации;

- осуществлять проверку спецификаций требований программного комплекса, чтобы удостовериться, что они соответствуют базовой концепции проекта и функций программного продукта;
- осуществлять управление изменением приоритетов испытаний задач и функций, а также добавлением и исключением новых функций комплекса программ.

Коллектив испытателей должен иметь в своем составе квалифицированных, **проблемно-ориентированных аналитиков и системных архитекторов**, способных переводить функциональные требования заказчика в конкретные спецификации и технические требования к испытаниям комплекса программ и его компонентов. Системные аналитики сложных ПС должны иметь, прежде всего, хорошую подготовку по системному анализу алгоритмов и комплексов программ, по методам оценки эффективности, организации и планированию крупных разработок и испытаниям программ. Им необходима высокая квалификация по архитектурному построению, и испытаниям версий программных продуктов определенных классов, умение организовать коллектив для решения общей целевой задачи реализации продукта. Это позволит на ранних этапах исключать или сокращать дефекты, обусловленные различием представления ими целей и задач комплекса программ, а также показателей качества.

Особую роль для обеспечения высокого качества программных продуктов играют испытатели – тестировщики. Они должны обеспечивать проверку реализации требований функциональных спецификаций, пользовательских интерфейсов, разрабатывать стратегию, выполнять и документировать тестирование для каждого компонента и версии программного продукта, должны быть административно независимыми от программистов и спецификаторов требований. Организационная структура испытателей и распределение ответственности за определенные уровни корректировок, имеют важное значение для постоянного совершенствования процесса и качества результатов. Различные виды работ, проводимые по Программе испытаний, обычно включаются в состав декомпозиции плана работ всего проекта. Необходимо определять и фиксировать в плане испытаний, распределение основных ролей и ответственности между участниками процесса тестирования [8, 19]. Испытатели должны обладать **специфической квалификацией** для разработки эффективной

стратегии тестирования, а также для проведения работ по испытаниям в необходимом и достаточном объеме.

Хороший испытатель должен обладать аналитическим складом ума, быть внимателен к деталям, организован, проявлять творческий подход и способности к прогнозированию, быть твердым и гибким при обсуждении спорных вопросов с разработчиками компонентов и требований к программному продукту. Ему целесообразно иметь опыт работы с различными платформами, операционными системами, интерфейсами других продуктов и систем, базами данных и языками разработки комплексов программ. Для эффективного проведения тестирования требований необходимо, чтобы испытатели обладали достаточным опытом осуществления принятой стратегии тестирования и применения выбранных инструментов.

Личные качества испытателей комплексов программ и отношения, установившиеся в коллективе, представляют собой значительный фактор и резерв для повышения эффективности разработки и безопасности программных продуктов. **Человеческий фактор** оказывает гораздо большее влияние на эффективность и качество, чем любой другой отдельно взятый фактор в ЖЦ комплексов программ (см. рис. 9.1). Ни один специалист не может быть носителем всех необходимых высоких качеств, однако **группа испытателей**, будучи единым коллективом, может воплощать максимально **возможное количество требуемых качеств** [7]:

- уметь описывать последовательность событий и конфигурацию системы, которые приводят к возникновению дефекта, это включает способность четко документировать процедуры и результаты тестирования требований, умение передать такую информацию разработчикам, другим испытателям и руководству;
- уметь критиковать и корректно воспринимать критику, так объяснять разработчикам компонентов суть дефектов, чтобы с его слов их можно было устранить;
- уметь противостоять внешнему давлению руководства, так как испытания являются завершающей стадией любого процесса разработки и, как правило, протекает в стрессовых обстоятельствах нехватки ресурсов и времени;
- быть терпеливым и готовым выполнять исполнение тестов столько раз, сколько нужно для того, чтобы выявить и локализовать

дефект, после чего повторно выполнить регрессионные тесты, чтобы убедиться в корректном его устранении;

- обладать гибким мышлением – быть способным быстро переключаться на испытания нового программного компонента или продукта, или даже отказываться от испытания одного продукта в пользу другого, обладающего более высоким приоритетом;
- обладать способностью видеть общую панораму развития проекта и уметь при необходимости сосредоточиваться на деталях, – иметь широкий, и динамичный кругозор.

С учетом сложности тестирования в клиент-серверной или многопользовательской среде испытатель должен обладать большой совокупностью технических знаний и навыков. Он должен иметь опыт работы с различными платформами, операционными системами, приложениями поддержки, интерфейсами других продуктов и систем, базами данных и языками разработки приложений.

Для сокращения и устранения дефектов разработки посредством адекватных контрмер необходима четкая организация коллектива специалистов и автоматизация процессов исправления дефектов, которые позволяют избегать множества вторичных ошибок, обусловленных недостаточной координацией проводимых корректировок и формирования новых версий сложных программных продуктов. Этому должна способствовать утвержденная дисциплина и иерархия принятия решений на координированные изменения компонентов и комплекса в целом, должностными лицами, поддержанная методами и средствами защиты от несанкционированного доступа при выполнении корректировок документов специалистами различной квалификации и *права доступа к модификациям* компонентов на разных уровнях проекта.

Следует установить *полномочия специалистов* или групп для санкционирования и выполнения изменений документов и корректировок на каждом уровне разработчиков изменений (см. таблицу 3.1). В контрмеры входит последовательность работ, которые необходимо выполнять для того, чтобы: запросить разрешение на изменение; обработать запрос на изменение; проследить изменение; распределить изменения в предыдущие версии программного продукта и документации. Изменения, которые воздействуют на программный продукт, уже находящийся на эксплуатации, должны быть предоставлены за-

казчику в соответствии с установленными контрактом формами и процедурами.

На основе проведенного анализа и испытаний персонал должен разрабатывать варианты реализации изменений и документально оформлять: сообщение о каждом дефекте или заявку на внесение изменений; результаты их анализа и варианты реализации изменений, **оценку их влияния на функциональную пригодность**. Следует согласовывать с заказчиком выбранные варианты изменений требований в соответствии с договором. Регистрация и учет истории этого процесса обеспечивает возможность его контроля и пошагового восстановления выполненных изменений (**отката**) документов для **выявления вторичных дефектов**, внесенных в процессе разработки очередной версии программного продукта. Такие дефекты обычно обусловлены одновременным, не скоординированным внесением групп изменений несколькими специалистами или потерей некоторых корректировок в определенной версии.

В процессах разработки и испытаний программного продукта может участвовать большое число специалистов различных направлений и квалификации, которые при необходимости целесообразно объединять в единый коллектив – **службу сертификационных испытаний на соответствие требованиям и управления конфигурацией программного продукта**. Менеджер проекта является высшим должностным лицом, принимающим важнейшие решения по внесению изменений и корректировке требований и конфигурации сложных комплексов программ. Он взаимодействует с заказчиком и пользователями, определяющими модификации, для согласования изменений требований к системе. Заказчик системы должен оценивать и утверждать наиболее крупные изменения, заметно влияющие на условия контракта, технические требования или стоимость программного продукта. При организации сложного проекта следует идентифицировать инстанцию, уполномоченную **утверждать версии программных продуктов и документов** для поставки заказчику и пользователям, и любые изменения к ним.

Перечисленные выше специализации и квалификации персонала, участвующего в испытаниях программных продуктов, **требуют соответствующего их отбора, подготовки и обучения**. Должны быть разработаны и документированы в плане проекта, требования и цели обучения специалистов, а также разработаны руководства,

включая документы, используемые для обучения (см. **ISO 19759** и **ISO 15504**). Персонал, ответственный за выполнение конкретных задач испытаний программных продуктов *с высоким риском* должен быть *аттестован*, если это необходимо, на основе соответствующего образования, подготовки и/или опыта работы. Может также стать необходимым включение в подготовку специалистов, ознакомление со специфической (проблемно-ориентированной) областью, в которой будет работать данный программный продукт, и повышение квалификации в этой области. Требования к квалификации и обучению персонала должны быть документально оформлены в контракте на проект.

Методы подготовки тестов для испытаний программных продуктов

Ручная пошаговая подготовка сценариев и содержания тестов не может обеспечить достаточно представительный их набор, для тестирования крупных комплексов программ реального времени на соответствие требованиям к функциям и характеристикам качества. Как правило, разработчиками моделей автоматизированной генерации тестов недооценивается сложность их разработки и пренебрегается необходимостью детализации к ним требований (рис. 9.2). *Реализации совокупности тестов должны полностью отражать возможность проверки выполнения требований к комплексу программ* и, соответственно, должны быть принципиально аналогичными по сложности и трудоемкости, разработке программного комплекса.

Эти *два представления комплексов программ* отличаются только формой описания их содержания: функциональным (процессным) или событийным (сценариями и результатами исполнения) (см. лекцию 8). Генерация тестов особенно сложна (также как и требований к программному продукту) для сложных комплексов программ реального времени. Выбор типов моделей генерации тестов зависит от глубины знаний об алгоритмах функционирования программ, характеристиках их качества и обобщенных параметрах результатов работы системы в целом. Кроме того, существенным для выбора типов моделей для генерации тестов может быть длительность расчета имитированных тестовых данных и обеспечение возможности проводить полную обработку результатов тестирования.

Автоматизация испытаний обычно выражается в способности системы тестов формироваться и выполняться без участия или при частичном участии человека [6].

Методы подготовки тестов для испытаний программных продуктов

Стратегия выбора тестов для испытаний программных модулей и компонентов:

- определение критериев выбора маршрутов для тестирования модулей и программных компонентов;
- определение стратегий упорядочения маршрутов для тестирования модулей и программных компонентов.

Требования к генерации динамических тестов внешней среды в реальном времени:

- задачи автоматизированного формирования динамических тестов внешней среды;
- преимущества программной имитации динамических тестов.

Компоненты генераторов динамических тестов внешней среды в реальном времени:

- исходные данные для генерации тестов внешней среды;
- набор средств и компонентов для формирования тестов внешней среды.

Средства обработка результатов динамических испытаний программных продуктов в реальном времени:

- средства оперативной обработки результатов динамических испытаний;
- средства и компоненты определения обобщенных характеристик программного продукта и системы.

Оценки эффективности динамической генерации тестов внешней среды в реальном времени:

- оценки затрат на программную имитацию динамических тестов;
- оценки экономической эффективности программной имитации динамических тестов.

Рис. 9.2

Процессы могут заключаться в автоматизированной работе с системой, предназначенной для тестирования, в загрузке данных для нее и в сравнении полученного результата с ожидаемым. Это может быть сделано с помощью инструментов, как доступных на рынке, так и разработанных на заказ. Автоматизация тестирования должна позволять испытателю спроектировать и разработать полный комплект тестовых сценариев, а затем с небольшими затратами или вовсе без них **повторять тестовые сценарии**.

Даже если автоматизация тестирования действительно имеет смысл, это по-прежнему остается большим и рискованным мероприятием. **Затраты, необходимые для поэтапной автоматизации генерации тестов**, могут быть не равны затратам на ручное написание тестов [5]. Для создания автоматизированных тестов, удобных для развития и сопровождения, должна быть выстроена инфраструктура, позволяющая испытателю определять действия, данные и ожидаемые результаты в удобном формате (см. рис. 9.2). Помимо затрат на создание тестовой инфраструктуры, часто серьезные вложения приходится делать в сами программные инструменты, которые могут стоить сотни тысяч долларов.

Если все-таки решено вложить средства в автоматизацию тестирования, следует проанализировать последствия этого решения в рамках выбранной стратегии испытаний. Поскольку для задач, решаемых в ручном режиме, и для задач, решаемых в автоматическом режиме, необходимо приобретать одни и те же аппаратные средства, соединять их между собой, подключать к компьютерным сетям и вводить их в эксплуатацию, может, имеет смысл сделать **автоматизированную динамическую тестовую среду постоянно действующей конфигурацией**. Подобный подход позволит выполнять автоматизированные тесты без вмешательства со стороны оператора. Если предполагается исполнять автоматизированные тесты в регрессионных испытаниях или для целей технического обслуживания системы, может иметь смысл построить специализированный стационарный **испытательный стенд**, который будет находиться на рабочей площадке испытателей весь период, пока поддерживается программный продукт.

Группа тестирования может при необходимости создавать специальные инструменты – **генераторы динамических тестов**, которые на основе некоторого набора правил автоматически генерируют

тесты в реальном времени. Эти правила можно получить из спецификаций требований к программному продукту, из документации базы данных проекта; испытатели могут разработать их вручную, чтобы привести в соответствие с требованиями к допустимой длительности испытаний. При необходимости генераторы могут динамически создавать тестовые данные, например, для проведения нагрузочного тестирования (см. лекцию 7). Некоторые генераторы тестовых процедур, могут обладать высокой степенью интеграции в процесс анализа и проектирования программных продуктов и позволять разработчикам тестировать функции в соответствии со спецификациями требований системной архитектуры. Другие генераторы тестовых процедур могут извлекать информацию о документированных требованиях из базы данных и создавать тестовые процедуры автоматически.

В результате испытаний программных продуктов необходимо достоверно устанавливать ***степень их соответствия утвержденным требованиям*** к функциям и характеристикам качества. Для этого они должны проходить тщательные динамические испытания в условиях их последующего применения. В реальных системах создание таких условий может быть очень дорого и опасно крупными рисками при проявлении не устраненных при тестировании дефектов и ошибок, что недопустимо. Альтернативой является создание и применение математических моделей на ЭВМ, ***динамически имитирующих реальную внешнюю среду*** для генерации тестов и функционирования тестируемых комплексов программ. Таким образом, формируется возможность выполнять испытания, выявлять и устранять ошибки, а также ***достигать высокого уровня соответствия программного продукта заданным требованиям.***

Стратегии выбора тестов для испытаний программных модулей и компонентов

Относительная простота программных модулей (ПМ) позволяет детально анализировать их внутреннюю структуру и любой маршрут исполнения программы. Это обеспечивает возможность реализации ***двух стратегий тестирования***: от структуры и от данных. Этим двум стратегиям соответствуют два метода тестирования программ: метод анализа потоков управления и анализа потоков данных. Методы дополняют друг друга, и каждый может быть доминирующим на

начальных этапах отладки в зависимости от типа объекта и условий тестирования (Рис. 9.3).

**Стратегии выбора тестов для испытаний
программных модулей и компонентов**

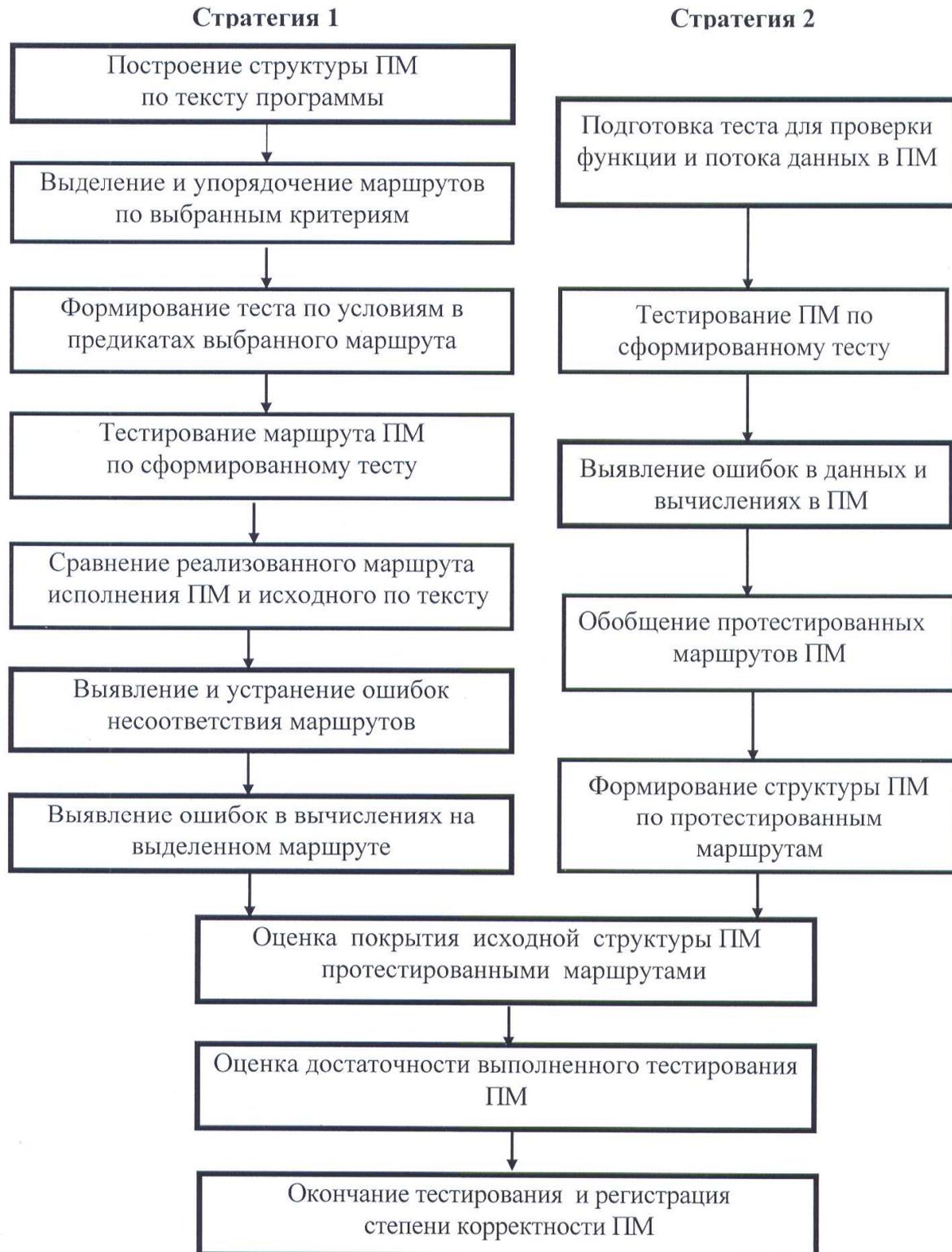


Рис.9.3

При *первой стратегии* за основу принимается структура ПМ, построенная по тексту программы в виде графа. В графе программы по некоторым критериям выделяются и упорядочиваются маршруты исполнения программы и условия-предикаты, при которых они могут быть реализованы. Эти условия используются для подготовки тестовых наборов, каждый из которых должен реализоваться по маршруту, принятому за эталон при подготовке теста. Отклонение исполнения теста от первоначально выбранного маршрута рассматривается как ошибка, причина которой может быть либо в первичной структуре ПМ, либо в реализации конкретного маршрута при данном тесте на входе.

После устранения ошибок и несоответствия, выбранных и реализованных маршрутов, для каждого из них проверяется процесс обработки данных, и выявляются ошибки в результатах их преобразования. Затем оценивается достаточность выполненного тестирования *по степени покрытия исходного графа программы* проверенными маршрутами, которые выделялись по выбранному или заданному критерию. Завершается тестирование при требуемом покрытии графа программы протестированными маршрутами или при использовании ресурсов, выделенных на тестирование. В последнем случае необходима оценка достигнутой корректности программы и регистрация этой величины.

При этой стратегии некоторые выделяемые маршруты могут оказаться принципиально нереализуемыми из-за противоречивых условий в последовательных операторах – вершинах графа программы. Вследствие этого для некоторых модулей могут подготавливаться тесты, которые являются избыточными и не отражают реальное функционирование программы. Тем не менее, данная стратегия имеет преимущества при тестировании логических программ с малой долей вычислений. При этом достаточно эффективно контролируется полнота тестирования при различных критериях выделения маршрутов и методах их упорядочения.

При *второй стратегии* (Рис. 9.3) за основу принимаются требования спецификаций, конкретные тестовые и эталонные значения, которые подготавливаются специалистами путем анализа переменных и предикатов в тексте программы. При каждом тесте программа исполняется по определенному маршруту, который регистрируется. При этом реализованный, в соответствии с анализируемыми требова-

ниями, маршрут и поток данных рассматриваются как эталонные компоненты структуры программы. По мере тестирования отмечаются проверенные операторы, и оценивается полнота **покрытия тестами требований спецификаций** на маршрутах тестирования. Накопление и обобщение реализованных маршрутов позволяет воспроизвести структуру программы и контролировать степень покрытия каждого компонента. Для этого на каждом этапе тестирования выделяются компоненты программы, оставшиеся непроверенными, для которых необходимо подготавливать дополнительные тесты. Однако, при такой стратегии трудно оценить степень покрытия и проверки всех маршрутов исполнения программы без использования структурного графа, построенного по ее исходному тексту.

Данная стратегия имеет преимущества при сравнительно простой структуре программы и при преобладании в ней вычислительных операторов. На каждом маршруте, упорядоченное варьирование переменных акцентирует внимание на вычислительной части программы, что существенно для такого класса ПМ. Однако возрастает риск пропустить сочетания предикатов, определяющих непроверенный маршрут. Поэтому практически всегда целесообразно **совместное применение двух стратегий**, с акцентом на одну из них в зависимости от особенностей тестируемого ПМ или его частей. Программы с преобладанием сложной логической структуры и с относительно малой вычислительной частью целесообразно начинать тестировать по первой стратегии и только для маршрутов с вычислительными операторами использовать анализ потоков данных (вторую стратегию). В модулях, имеющих простую структуру и содержащих значительный объем вычислений после первичной отладки по второй стратегии, целесообразна проверка полноты тестирования структуры и завершение отладки по первой стратегии.

Известно, что исчерпывающее тестирование, гарантирующее полное отсутствие дефектов и ошибок, принципиально не возможно и число дефектов, обнаруживаемых в программах, даже независимыми испытателями имеет пределы. Реальные ограничения доступных ресурсов определяют количество не устраненных дефектов и достижимую корректность компонентов и ПС в целом. На практике учитывать степень покрытия тестами достаточно трудоемко и зачастую желательнее иметь более простой способ регламентирования и оценивания качества тестирования. Возникает проблема, как практически

учесть ограничения ресурсов и **когда прекратить верификацию и тестирование**, а также какая при этом будет достигнута **корректность программ**, и способна ли она удовлетворить заказчика и пользователя при эксплуатации ПС.

В качестве исходной информации **при тестировании структуры программ** используется схема связей между операторами текста программы. По этой схеме выделяется полный набор маршрутов исполнения программы, подлежащих тестированию. Для каждого выделенного маршрута по тексту программы формируется набор условий, определяющих его реализацию и используемый при создании соответствующего теста. Детерминированное тестирование структуры программных модулей имеет целью проверку корректности выделенных маршрутов исполнения программ и обнаружение в основном логических ошибок формирования маршрутов. На практике при отсутствии упорядоченного анализа потоков управления некоторые маршруты в программе (до 50%) оказываются пропущенными при тестировании. Поэтому **первая задача**, которая решается при тестировании структуры программ, – это получение информации о полной совокупности реальных маршрутов исполнения в каждой программе. Такое представление маршрутов позволяет упорядоченно контролировать достигнутую степень проверки маршрутов и в некоторой степени предохраняет от случайного пропуска отдельных пропущенных маршрутов. В результате значительно повышается достигаемое качество программ, и тестирование приобретает планируемый систематический характер.

Анализ критериев тестирования и выделение тестируемых маршрутов удобно проводить, используя графовые модели программ. При планировании тестирования структуры программ возникают, прежде всего, **две задачи**: формирование критериев выделения маршрутов для тестирования и выбор стратегий упорядочения выделенных маршрутов.

Критерии выделения маршрутов для тестирования соответствуют критериям определения структурной сложности программных модулей. В основном используются критерии:

- покрытия графа программы минимальным количеством маршрутов, охватывающих каждую дугу графа хотя бы один раз;
- выделения линейно-независимых маршрутов на базе понятия цикломатического числа;

- выделения маршрутов при всех возможных комбинациях дуг, входящих в маршруты.

Число маршрутов, выделяемых по этим критериям, зависит от структуры программы. В реальных программах часть маршрутов может быть нереализуемой из-за противоречий в условиях, анализируемых на последовательных участках маршрута. Это может приводить к сокращению числа маршрутов по любому критерию. Невыполнение правил структурного построения программ, наоборот, приводит к возрастанию числа маршрутов, выделяемых по третьему критерию. К значительному возрастанию числа маршрутов обычно приводят циклы в программах. Несмотря на перечисленные обстоятельства, приведенные критерии являются достаточно удобными для практических оценок сложности и полноты тестирования структуры программ. Планировать тестирование можно по одному из критериев или, используя последовательно более жестко критерии выделения маршрутов, при которых соответственно возрастает объем и сложность тестирования.

Стратегии упорядочения маршрутов учитывают сложность маршрута и тестов для его проверки (количество операторов, условных переходов и циклов в маршруте, частота его исполнения при рабочем функционировании ПС, сложность получения соответствующих эталонных данных и другие факторы). В первую очередь целесообразно производить проверку основной группы маршрутов с экстремальными значениями показателя в пределах ресурсов, выделенных для тестирования. При имеющихся ограничениях некоторая часть маршрутов может оказаться не проверенной и характеризует достигнутую корректность данной программы по выбранному критерию.

Упорядочение маршрутов при планировании тестирования в ПМ базируется на использовании в основном **трех характеристик программных модулей**:

- числа команд в выделенных маршрутах или расчетной длительности их реализации (стратегия 1);
- числа альтернатив или условных переходов, определяющих образование каждого маршрута (стратегия 2);
- вероятности исполнения маршрутов при реальном функционировании программы (стратегия 3).

Эти стратегии тестирования позволяют сосредоточить внимание разработчика на анализе наиболее важных компонентов программ. При стратегии 1 первичному тестированию подлежат маршруты, наиболее длинные по числу команд и по времени исполнения. Им соответствуют обычно маршруты с наибольшим объемом вычислений и преобразований переменных. Эта стратегия целесообразна при планировании тестирования программ, имеющих вычислительный характер обработки данных при небольшом числе логических условий и маршрутов исполнения программ. Выбранные первичные маршруты не обязательно являются наиболее сложными по логике функционирования.

При стратегии 2 приоритет отдается маршрутам, наиболее сложным по числу анализируемых условий. Такая стратегия предпочтительна при тестировании логических программ с небольшим объемом вычислений. При обеих стратегиях на завершающие этапы тестирования остаются простые по вычислениям или по логике маршруты, для которых необходимы относительно короткие тесты. Это соответствует традиционной стратегии многих разработчиков программ подготавливать вначале тесты с возможно большим охватом компонентов отлаживаемой программы.

В типичной программе на долю 3% операторов приходится до 50% времени исполнения всей программы. В результате изучения профиля программы могут быть выявлены наиболее критические по длительности или самые частые маршруты исполнения программы. Использование профиля программ позволяет сконцентрировать усилия разработчиков на проверке наиболее активно функционирующих компонент и выделить слабо проверенные части программ, которые исполняются редко.

Планирование тестирования структуры программных модулей в значительной степени может быть автоматизировано. Задача автоматизированных систем *планирования тестирования* состоит в выделении маршрутов программ по одному или нескольким критериям с последующим упорядочением по заданной стратегии. В результате разработчик программ автоматизировано информируется о составе маршрутов в программе для проведения упорядоченного тестирования. Кроме того, если фиксировать маршруты, по которым уже проведено тестирование, то их можно автоматически исключать из информирования и выдавать на регистрацию только группу маршрутов,

подлежащих первоочередной проверке. Эти же данные могут использоваться для автоматического расчета полноты проверки и для оценки достигнутой корректности программы по каждому из критериев выбора маршрутов.

Сложность программы определяется числом взаимодействующих компонентов, числом связей между компонентами и сложностью их взаимодействия. При функционировании программы разнообразие ее поведения и разнообразие связей входных и результирующих данных в значительной степени определяются набором путей-маршрутов, по которым исполняется программа. Установлено, что сложность программного модуля зависит не столько от размера программы (числа команд), сколько от числа отдельных путей ее исполнения, существующих в программе. Все маршруты возможной обработки данных должны быть проверены при создании программы и тем самым определяют сложность ее тестирования.

Маршруты исполнения программного модуля можно условно разделить на **два вида**: маршруты исполнения преимущественно вычислительной части программы и преобразования непрерывных переменных, и маршруты принятия логических решений и преобразования логических переменных. **Маршруты первого вида** обычно логически проще и короче, чем второго, и предназначены для преобразования величин, являющихся квантованными результатами измерения некоторых непрерывных физических характеристик. Такие переменные связаны условиями гладкости, т.е. условиями малых изменений производных этих переменных по времени или по другим параметрам.

При оценке сложности вычислительных маршрутов программ необходим учет числа операндов, участвующих в вычислениях. Кроме того, исходные и результирующие данные при тестировании должны принимать несколько значений. Во всем диапазоне исходных переменных следует выбрать несколько характерных точек (предельные значения и несколько промежуточных), при которых проверяется программа. В особых точках значений и сочетаний переменных и в точках разрыва функции необходимо планировать дополнительные проверки. Таким образом, сложность проверки программного модуля будет определяться числом маршрутов исполнения программы и числом обрабатываемых операндов на каждом маршруте, умноженном на число значений для каждой исходной величины на этом маршруте.

Второй вид маршрутов является результатом функционирования схем принятия решений и преобразования логических переменных. Для логических переменных отсутствует сильная корреляционная связь между соседними значениями, и каждое изменение разряда переменной может определять разные области результирующих значений. Такое преобразование переменных обеспечивается алгоритмами со сложной логической структурой, содержащей ряд проверок логических условий, циклов для поиска и селекции переменных, а также логические преобразования переменных. В результате в программе образуется множество маршрутов обработки исходных данных, которые определяют сложность структуры программы.

Программные модули являются наиболее массовой компонентой в ПС и **требуют для тестирования суммарно наибольших затрат**. Затраты на тестирование каждого модуля прямо пропорциональны сложности, которая зависит от его структуры и объема вычислений. При тестировании программного модуля необходимо задать и проанализировать соответствующее число значений параметров. Суммарные затраты на тестирование модуля пропорциональны значению его сложности.

Некоторые модули могут оказаться недостаточно протестированными из-за их высокой сложности и необходимости больших затрат, которые всегда ограничены. Их **не полные испытания определяют качество функционирования групп и комплекса программ в целом**. Ограниченность ресурсов на тестирование программных модулей приводит к целесообразности **выравнивания их структурной сложности** и выделения затрат на проверку в соответствии со сложностью каждого модуля. Особенно сложные модули следует делить на мелкие и простые и так перестраивать их структуру, чтобы сокращалось суммарное число маршрутов в модуле и их длина.

Сложность тестирования групп программ определяется сложностью модулей и межмодульных связей по управлению и по информации. Каждый модуль тестируется автономно до включения в группу программ и частично в составе группы. Затраты на тестирование модулей в составе группы программ должны учитывать относительные суммарные затраты на тестирование всех входящих модулей с коэффициентом <1 , зависящим от степени проверки соответствующего модуля. Если модули автономно не тестировались (например, при нисходящем тестировании), то коэффициент $=1$ и затраты на

тестирование каждого модуля войдут в затраты при тестировании группы программ в полном объеме. При тщательном автономном тестировании модулей можно полагать коэффициент = 0,1 – 0,01, т.е. в ПС затраты на тестирование модулей составляют несколько процентов.

При анализе сложности тестирования межмодульных связей по управлению и по информации следует учитывать, что тестирование каждой информационной связи обычно необходимо при нескольких значениях каждой переменной, что соответственно увеличивает затраты. Оценка относительной сложности тестирования групп программ способствует правильному распределению ресурсов (машинного времени, труда специалистов и т.д.), выделяемых на тестирование групп разной сложности. Кроме того, такие оценки стимулируют более полное тестирование программных модулей, так как обнаружение и локализация каждой ошибки в группе программ на порядок дороже, чем в автономном модуле. Поэтому целесообразно при тестировании групп программ составляющую затрат на тестирование модулей иметь минимальной.

Требования к генерации динамических тестов внешней среды в реальном времени

Для обеспечения высокого качества сложных комплексов программ реального времени необходимы соответствующие *проблемно-ориентированные интегрированные системы автоматизации динамического тестирования*, способные достаточно полно заменить испытания программ с реальными объектами внешней среды (см. рис. 9.2). При этом высокая стоимость и риск испытаний с реальными объектами почти всегда оправдывают значительные затраты на такие интегрированные системы, если предстоят испытания критических программных продуктов с высокими требованиями к качеству функционирования программного продукта, с длительным жизненным циклом и множеством развивающихся версий.

Инструментальные средства автоматизации динамических испытаний комплексов программ реального времени должны обеспечивать [5, 6]:

- определение и формирование динамических тестов – реализацию процесса тестирования разработчиком: ввод тестовых наборов;

генерацию тестовых данных; ввод ожидаемых, эталонных результатов;

- выполнение участка тестируемой программы между контрольными точками, для которого средство тестирования может перехватить операторский ввод (клавиатуры, мыши и т.д.) и для которого вводимые данные могут быть отредактированы и включены в следующие тестовые сценарии;

- управление тестами и участком программы, для которого средство тестирования может автоматически выполнять тестовые наборы;

- анализ и обработку тестовых результатов – возможность средства тестирования автоматически анализировать части тестовые результаты: сравнение ожидаемых и реальных результатов; сравнение файлов; статистическую обработку результатов;

- анализ покрытия тестами исходных требований к программному продукту для обнаружения: функций, которые частично не были выполнены; процедур, которые не были вызваны; данных, к которым не было обращений;

- анализ производительности комплекса программ, когда он выполняется: загрузку центрального процессора; загрузку памяти; обращения к специфицированным элементам данных и/или сегментам кода; временные характеристики функционирования испытываемой программы;

- моделирование внешней среды – поддержку процесса тестирования с помощью модели динамической имитации данных из внешних для программного продукта аппаратных компонентов системы.

Методы *динамических испытаний с исполнением комплекса программ*, в большей или меньшей степени, ориентированы на обнаружение ошибок определенных типов, преимущественно в структуре комплекса программ и реализуемых маршрутах обработки информации. Такая ориентация позволяет упорядочивать последовательность приоритетного применения методов с целью устранения, прежде всего, ошибок в наибольшей степени отражающихся на корректности исполнения программ, а также сосредоточиваться на методах, позволяющих решать частные задачи достижения необходимого их качества и соответствия требованиям при минимальных затратах.

Характеристики динамического функционирования программных продуктов зависят не только от их внутренних свойств, но и *от*

свойств внешней среды, в которой они применяются (см. стандарт **ISO 12119**). Для сокращения неопределенностей и прямых ошибок при оценивании качества программ необходимо до начала испытаний определить основные параметры внешней среды и потоки информации, при которых должен функционировать комплекс программ с требуемыми характеристиками при оценивании его качества и эксплуатации. Для этого заказчик и разработчик совместно должны структурировать, описать и согласовать **модель внешней среды** и ее параметры в среднем, типовом режиме применения, а также в наиболее вероятных и критических режимах, в которых должны обеспечиваться требуемые характеристики качества динамического функционирования программного продукта. Такая **модель должна отражать характеристики**:

- внешних динамических потоков информации, в том числе их распределение по видам источников, характеристикам качества данных и возможным дефектам;
- интенсивность и структуру типовых сообщений от оперативных пользователей и администраторов, и их необходимую квалификацию, отражающуюся на вероятности ошибок и качестве выдаваемой информации;
- возможных негативных и несанкционированных воздействий от внешней среды при применении программного продукта;
- необходимые характеристики вычислительных средств, на которых предназначено функционировать комплексу программ с требуемым качеством.

При создании генераторов динамических тестов внешней среды применяется два принципиально различающихся подхода, которые условно можно назвать интегральным и дифференциальным. При **интегральном** или эмпирико–статистическом подходе основой является формальное описание входной и выходной информации имитируемого динамического объекта, а также функциональной связи между данными на его входе и выходе. При этом структура объекта и процессы, реализующиеся при реальном функционировании его компонентов, не имеют значения и не моделируются. Исходные данные и характеристики для построения таких генераторов тестов получают в натуральных экспериментах или при исследовании более детальных – дифференциальных моделей. **Дифференциальные** или имитационные модели генераторов динамических тестов базируются на описаниях

внутренних процессов функционирования компонентов объекта моделирования, его структуры и взаимодействия составляющих. Результаты функционирования таких моделей определяются адекватностью знаний о компонентах и их характеристиках, а также об их взаимосвязях.

В отличие от натурального эксперимента моделирование внешней среды и динамических тестов на ЭВМ имеет большие возможности контроля, как исходных данных, так и всех промежуточных и выходных результатов функционирования испытываемого программного продукта. В реальных системах ряд компонентов иногда оказывается недоступным для контроля их состояния, так как либо невозможно поместить измерители контролируемых сигналов в реальные подсистемы, подлежащие тестированию, либо это сопряжено с изменением характеристик самого анализируемого объекта. В отличие от этого натурные эксперименты зачастую невозможно остановить на некоторой промежуточной фазе или повторить с абсолютно теми же исходными данными.

Программная имитация динамических тестов внешней среды на ЭВМ в реальном времени позволяет:

- проводить длительное непрерывное генерирование имитируемых данных для определения характеристик функционирования комплекса программ в широком диапазоне изменения условий и параметров, что зачастую невозможно при использовании реальных объектов;
- расширять диапазоны характеристик имитируемых объектов за пределы реально существующих или доступных источников данных, а также генерировать динамические потоки информации, отражающие перспективные характеристики создаваемых информационных систем и объектов внешней среды;
- создавать тестовые данные, соответствующие критическим или опасным ситуациям функционирования объектов внешней среды, которые невозможно или рискованно реализовать при натуральных экспериментах;
- обеспечивать высокую повторяемость имитируемых данных при заданных условиях их генерации и возможность прекращения или приостановки имитации на любых фазах моделирования внешней среды.

Компоненты генераторов динамических тестов внешней среды в реальном времени

Одними из наиболее сложных и дорогих имитаторов внешней среды, применяемых для испытаний крупных программных продуктов, являются модели: полета космических аппаратов; диспетчерских пунктов управления воздушным движением; объектов систем противовоздушной обороны; сложных административных систем и другие [5]. Применяемые для этого моделирующие испытательные стенды (МИС) проблемно – ориентированы и размеры программ, моделирующих в них динамическую внешнюю среду, могут даже значительно превышать размеры соответствующих испытываемых программных продуктов. Для их реализации должны выделяться достаточно мощные универсальные *моделирующие ЭВМ*. Кроме того, для автоматизации разработки программных средств могут использоваться отдельные *технологические ЭВМ*, что в совокупности образует инструментальную базу для обеспечения всего ЖЦ сложных комплексов программ реального времени на *объектных реализующих ЭВМ*.

Имитация конкретных динамических тестов с реальными характеристиками, адекватными объектам внешней среды, является основной частью типовых моделирующих стендов (см. рис.9.2). В соответствии с полной номенклатурой и реальными характеристиками таких объектов создаются их интегральные или дифференциальные модели. Выбор типов моделей зависит от глубины знаний об алгоритмах функционирования внешних объектов, характеристиках их компонентов и обобщенных параметрах работы объекта в целом.

Разумное *сочетание части реальных объектов внешней среды и имитаторов на ЭВМ* обеспечивает создание высокоэффективных МИС с комплексными моделями совокупностей объектов внешней среды, необходимых для динамических испытаний качества программных продуктов и систем в реальном времени. Такие стенды позволяют автоматическую генерацию тестов с помощью имитаторов на ЭВМ и аналогов реальной аппаратуры дополнять реальными данными от операторов пользователей, контролирующих и корректирующих функционирование систем обработки информации.

Для каждого эксперимента при динамических испытаниях комплексов программ реального времени следует подготавливать план

сценариев тестирования и *обобщенные исходные данные*. В моделирующей ЭВМ план и обобщенные исходные данные преобразуются в конкретные значения параметров для задания динамического функционирования каждого имитатора или реального объекта внешней среды. Эти данные вводятся и преобразуются на моделирующей ЭВМ вне реального масштаба времени и подготавливают старт сеанса функционирования стенда и испытываемых программ в реальном времени. После этого начинают генерироваться тестовые данные.

Аналоги системы и аппаратуры внешней среды используются преимущественно для генерации динамических тестов, представляющих коррелированные логические переменные, которые трудно описать и смоделировать на ЭВМ. Кроме того, они позволяют проверить и аттестовать некоторые программные имитаторы внешней среды, которые впоследствии играют основную роль при испытаниях. В ряде случаев такие аналоги аппаратуры не могут отразить все особенности объектов внешней среды, и имитаторы на ЭВМ остаются единственными источниками соответствующей части данных для проверки качества программного продукта.

Данные с рабочих мест операторов-пользователей должны отражать реальные характеристики динамических воздействий на тестируемый программный продукт с учетом особенностей и квалификации человека, которому предстоит использовать испытываемые программы в реальной системе обработки информации. На эту часть МИС кроме первичных исходных данных от моделирующей ЭВМ могут вводиться данные обработки ряда тестов испытываемой системой.

Данные натурных экспериментов с объектами внешней среды могут подготавливаться заранее вне сеансов динамических испытаний программного продукта, например, при отладке аппаратной части системы обработки информации. Эти данные отражают характеристики и динамику функционирования объектов, которые трудно или опасно подключать для непосредственного взаимодействия с недостаточно проверенными программами. Кроме того, такие данные могут использоваться для аттестации адекватности имитаторов некоторых объектов внешней среды.

При динамическом тестировании в ряде случаев необходимо иметь *эталонные характеристики данных*, поступающих на испытываемый программный продукт или систему. При работе с реаль-

ными объектами зачастую приходится создавать специальные измерительные комплексы, которые определяют, регистрируют и подготавливают к обработке все необходимые характеристики в процессе реального функционирования этих объектов (например, координаты движения самолетов при испытаниях систем УВД). Такие измерения проводятся при автономном функционировании внешних объектов или при их взаимодействии с комплексом программ в реальном времени.

Повторяемость сеансов испытаний при автоматической имитации динамических тестов обеспечивается фиксированием всех исходных данных и применением программного формирования псевдослучайных чисел. При надежной работе аналогов реальных объектов и моделирующей ЭВМ, в принципе, можно добиться почти абсолютной повторяемости весьма длительных экспериментов и сценариев тестирования. Труднее обеспечивать повторяемость сценариев динамических испытаний, в которых активно участвует оператор-пользователь. В этом случае необходимо регистрировать и сохранять действия оператора в зависимости от времени, а затем повторять их в соответствии с записанным сценарием. При необходимости временная диаграмма может соблюдаться с точностью около 0,5 - 1 с, однако ошибки в действиях оператора и вводимых им параметрах могут отличаться в каждом сценарии тестирования.

После приемки заказчиком или приобретения пользователями, в процессе функционирования и применения программного продукта должно обеспечиваться их регулярное тестирование и оценка текущего качества. Для этого в **составе программного продукта необходимы средства, обеспечивающие:**

- генерацию динамических тестовых сценариев или хранения тестов для контроля работоспособности, сохранности и целостности программного продукта при функционировании и применении;
- оперативный контроль и обнаружение дефектов исполнения программ и обработки данных при использовании программного продукта по прямому назначению;
- реализацию процедур предварительного анализа выявленных дефектов и оперативное восстановление вычислительного процесса, программ и данных (рестарт) после обнаружения аномалий динамического функционирования программного продукта;

- мониторинг, накопление и хранение данных о выявленных дефектах, сбоях и отказах в процессе исполнения комплекса программ и обработки данных.

Средства генерации динамических тестов и имитации внешней среды в составе поставляемого программного продукта предназначены для оперативной подготовки исходных данных при проверке различных режимов функционирования в процессе применения программного продукта и при диагностике проявившихся дефектов. Минимальный состав средств генерации тестов должен передаваться пользователям для контроля использования рабочих версий в реальном времени и входить в комплект поставки каждой пользовательской версии программного продукта.

Важной функцией испытательных стендов является их использование в качестве **тренажеров для операторов-пользователей**. Так как качество функционирования комплексов программ может существенно зависеть от характеристик конкретного человека, участвующего в эксплуатации и обработке информации, то необходимо измерять эти характеристики. Необходимо также иметь возможность их улучшать до уровня, обеспечивающего выполнение **заданных требований к программному продукту**. Поэтому в процесс испытаний органически должен входить процесс динамической тренировки и измерения характеристик реального функционирования и реакции операторов, а также использование МИС для обучения и регулярной подготовки операторов-пользователей в процессе тиражирования и эксплуатации программного продукта.

Для определения качества программных продуктов существенна **адекватность имитаторов динамических тестов**, которая зависит от степени учета второстепенных факторов, характеризующих функционирование реальных объектов или источников информации, при создании их моделей. Точность моделей на ЭВМ, прежде всего, определяется алгоритмами, на которых они базируются, и полнотой учета в них всех особенностей моделируемых объектов. Кроме того, на адекватность имитации влияет уровень дефектов и ошибок в программах имитации. Каждый, не учитываемый в имитаторе элемент или фактор моделируемой системы, необходимо оценивать путем сопоставления частных имитируемых данных с результатами аналитических исследований или с данными, полученными на реальных системах, и определять его возможное влияние на полную требуемую

точность модели и генерируемых динамических тестов с учетом других составляющих, отражающихся на достоверности имитации.

Средства обработки результатов динамических испытаний программных продуктов в реальном времени

Регистрация и обработка характеристик динамических тестовых данных должна обеспечивать их контроль на соответствие заданным требованиям к программному продукту, обобщенным характеристикам каждого объекта внешней среды и исходным данным сеанса испытаний. Так проводится процесс испытаний по конечным результатам функционирования комплекса программ, выдаваемым внешним абонентам и для определения интегральных характеристик качества и их соответствия требованиям к программному продукту.

Селекция результатов динамических испытаний может основываться на стратегии контроля функционирования программного продукта *снизу вверх*, т.е. от анализа исполнения отдельных компонентов программы и далее до стохастических результатов функционирования всего комплекса в динамике реального времени. При этом регистрируется избыточное количество данных, из которых затем отбирается минимум, необходимый для анализа. Может использоваться стратегия *сверху вниз*, т.е. упорядоченное, иерархическое выделение в первую очередь обобщенных результатов функционирования комплекса программ с последующим уточнением регистрируемых и анализируемых результатов вплоть до детального контроля исполнения отмеченных программных компонентов. В этом случае регистрируются только те данные, которые необходимы для анализа в конкретном сеансе динамического тестирования

Вызовы регистрирующих программ должны подчиняться определенной системе контроля динамического функционирования программ при исходной гипотезе, что **некоторые ошибки и дефекты в программах и данных могут проявиться на любой стадии испытаний**. Однако количество вызовов регистрирующих программ и контроль промежуточных результатов, требующих нарушения целостности исполнения функциональных программ, следует ограничивать, учитывая допустимые расходы ресурсов времени на их реализацию.

Так как испытания современных сложных систем обработки информации и/или управления позволяют получать такое большое

количество контрольных данных, что достаточно полный их анализ представляет трудную методическую и техническую задачу, **обработку динамических результатов целесообразно осуществлять иерархически и дифференцировано**. В каждом конкретном случае необходимо стремиться к компромиссу между полнотой регистрации промежуточных данных тестирования и удобством **анализа обобщенных результатов на соответствие требованиям**.

Обработка результатов испытаний программных продуктов реального времени может быть разделена на две, достаточно автономные, части: оперативную и обобщающую. **Оперативная обработка результатов динамического тестирования** должна производиться по упрощенным алгоритмам с большой пропускной способностью, обеспечивающим сохранение реального масштаба времени для всего испытываемого комплекса программ. В оперативную обработку целесообразно включать расчет части интегральных данных динамического тестирования, позволяющих контролировать текущий процесс обработки информации испытываемым комплексом.

Обобщающая обработка накопленных результатов испытаний может производиться вне реального времени после завершения одного или серии испытаний. Основная задача при этом состоит в расчете различных интегральных характеристик качества функционирования программного продукта и **соответствия их заданным требованиям**. Зарегистрированные и обработанные результаты испытаний должны использоваться для установления соответствия полученных характеристик качества **заданным требованиям**. Средства накопления сообщений об отказах, ошибках, предложениях на изменения, выполненных корректировках и оцененных характеристиках качества версий являются основой для конфигурационного управления развитием и совершенствованием комплекса программ.

Оценка эффективности динамической генерации тестов в реальном времени

Затраты на имитацию внешней среды и на генерацию динамических тестов для оценивания качества программных продуктов могут быть одной из существенных составляющих при их создании. В ряде случаев, они соизмеримы с затратами на создание основных функций комплексов программ, что определяется принципиальным соответствием **сложности необходимых наборов тестов** и тестово-

го покрытия программ, и **сложности функций**, реализуемых испытываемым комплексом программ. Создание представительных совокупностей динамических тестов возможно путем использования реальных объектов внешней среды или с помощью программных имитаторов, адекватных этим объектам по результатам функционирования и генерируемой информации. При этом возникает проблема – **какой метод и когда выгодней** по затратам на генерацию тестов и по обеспечению необходимой степени покрытия тестами испытываемых комплексов программ (см. рис. 9.2).

Имитаторы тестов могут быть необходимы не только для оценивания достигнутых характеристик качества комплексов программ, но также для их комплексной отладки, квалификационного тестирования, испытаний и при создании версий. Поэтому затраты на программные имитаторы и их экономическую эффективность целесообразно рассматривать в проекте с учетом всего комплекса задач, которые они способны и должны решать в жизненном цикле программного комплекса. Анализ эффективности программной имитации внешней среды при разработке и определении качества целесообразно разделять **на две части**: оценка факторов, определяющих эффективность средств имитации тестов, и оценка экономического выигрыша при моделировании внешней среды на ЭВМ по сравнению с натурными экспериментами в реальных системах.

Факторы, определяющие эффективность программной имитации внешней среды на ЭВМ при разработке крупных комплексов программ, могут оцениваться в основном по их воздействию на качество создаваемых систем. Это влияние трудно непосредственно измерить, однако качественный анализ показывает, что автоматизированная имитация динамических тестов может значительно изменять не только достигаемые характеристики качества разрабатываемого программного продукта, но также трудоемкость и длительность его создания. Для каждого параметра, отражающего внешнюю среду, отношение диапазона или числа тестов, возможных при программной имитации на ЭВМ по сравнению с натурными экспериментами, может служить оценкой величины, **возрастания достоверности характеристик качества** программных продуктов.

Некоторые значения тестов не только трудно создать при натуральных экспериментах, но они являются маловероятными в реальных условиях. Однако такие, даже маловероятные ситуации и значения

тестов могут быть **критическими и/или особо важными** для функционирования всей системы, для которой разрабатывается программный продукт. Выбор и имитация подобных ситуаций позволяют отрабатывать и оценивать качество в критических маловероятных ситуациях, которые невозможно или опасно создавать на реальных объектах, но без их выполнения некоторые продукты недопустимо эксплуатировать в критических системах управления и обработки информации.

Экономическую эффективность программной имитации внешней среды на ЭВМ по сравнению с натурными экспериментами целесообразно оценивать при одинаковых объемах динамических тестовых данных для испытаний и определения качества. Показателем экономической эффективности имитации может служить **соотношение затрат** на проведение натуральных экспериментов и затрат на программную имитацию той же совокупности тестовых и эталонных данных. Таким образом, затраты на натурные эксперименты для оценивания характеристик комплексов программ определяются использованием всей реальной внешней среды, в которой предстоит в дальнейшем функционировать программ, а также затратами на средства измерения характеристик этой среды и проверяемого программного продукта в процессе разработки, испытаний и определения качества.

Затраты на программную имитацию динамических тестовых данных определяются ресурсами необходимыми на проектирование и эксплуатацию сложных комплексов программ для этих целей. Имитационные стенды практически всегда являются уникальными. В ряде случаев эти комплексы программ могут иметь объем порядка 10^4 – 10^6 строк текста и должны создаваться с применением современных технологических систем.

Даже приближенные оценки при системном анализе соотношения этих затрат в большинстве случаев показывают **высокую рентабельность программных имитаторов внешней среды**, особенно для испытаний и оценивания характеристик качества программных продуктов реального времени. Например, при тестировании системы для управления воздушным движением, применение имитационных стендов, по крайней мере, на порядок снижает затраты по сравнению с натурными экспериментами и использованием реальных объектов (самолетов), а для управления космическими аппаратами или атомными электростанциями это соотношение может быть значительно

больше (~ 10 – 100). При создании и определении качества административных систем с полной загрузкой, имитация способна заменить сложную организацию функционирования по определенной программе большого коллектива операторов банка, налоговой инспекции или таможенного органа.

Примером сложного испытательного стенда и моделей внешней среды для динамического тестирования и проверки *на соответствие требованиям* к функциям и характеристикам комплексов программ может рассматриваться система *управления полетами воздушных судов и диспетчерских систем в центрах управления воздушным движением* [5]. Для комплексной отладки, тестирования, испытаний и сертификации программных продуктов управления воздушным движением (УВД) проводится имитация в реальном времени всей информации, поступающей из внешней среды. Источниками информации для центров УВД являются радиолокационные станции, летный состав на борту воздушных судов, диспетчеры управления воздушным движением и исходные планы полетов. Вследствие этого необходимо динамически имитировать ряд разнородных объектов с учетом интенсивных случайных воздействий от них, а также при наличии управления со стороны диспетчеров центра УВД и летного состава на борту воздушных судов. Некоторые редкие проявления ошибок в программных продуктах могут компенсироваться диспетчерами, контролирующими функционирование центра УВД. Имитировать реакцию и действия диспетчеров автоматически на ЭВМ очень трудно, так как они в значительной степени зависят от квалификации и конкретных психологических особенностей поведения диспетчеров при различных ситуациях воздушной обстановки. Поэтому при комплексной отладке и динамических испытаниях программных продуктов центров УВД обычно участвуют реальные диспетчеры и их средства управления.

Лекция 10

СЕРТИФИКАЦИОННЫЕ ИСПЫТАНИЯ ПРОГРАММНОГО ПРОДУКТА НА СООТВЕТСТВИЕ ТРЕБОВАНИЯМ

Порядок сертификационных испытаний сложного программного продукта

Основной целью сертификации программных продуктов, является контроль и удостоверение качества продукции, гарантирование ее высоких потребительских свойств – рис. 10.1. Задача состоит в повышении эффективности затрат в сфере создания и применения ответственного, конечного программного продукта, а также улучшение объективности оценок его функций, характеристик и конкурентоспособности (см. лекцию 1). Формальной целью сертификации является подготовка и принятие решения о целесообразности выдачи *сертификата соответствия* с учетом следующих факторов:

- полноты, точности и достоверности исходного технического задания и спецификаций требований, представленной в документации на программный продукт;
- достоверности и точности измерения и обобщения результатов сертификационных испытаний, получения адекватных показателей качества конечных продуктов и соответствия требованиям заказчика;
- методологии и качества интерпретации данных об объекте испытаний с учетом достоверности оценок, квалификации и объективности испытателей, заказчиков и пользователей.

В международных стандартах *сертификация соответствия* определена как действие третьей – *независимой стороны*, доказывающее, что обеспечивается необходимая уверенность в том, что должным образом идентифицированная продукция соответствует конкретным стандартам и/или другим нормативным документам.

В понятие нормативные документы включены документы, содержащие правила, общие принципы или характеристики, стандарты, технические требования, инструкции и регламенты по применению конкретной продукции.

Сертификационные испытания программного продукта на соответствие требованиям

Порядок сертификационных испытаний сложного программного продукта:

- требования технического задания и спецификаций требований на программный продукт и систему;
- Программа испытаний программного продукта на соответствие утвержденным требованиям;
- методики и протоколы испытаний по разделам Программы и требований спецификаций на программный продукт;
- функциональные испытания для оценивания характеристик и качества программного продукта;
- комплект адекватной эксплуатационной и технологической документации на программный продукт.

Испытания надежности функционирования программного продукта:

- экспериментальные методы оценивания надежности программного продукта в штатном режиме;
- форсированные испытания для оценивания надежности программного продукта;
- повышение надежности комплекса программ путем оперативного контроля и рестарта.

Испытания функциональной безопасности программного продукта:

- оценивание эффективности испытаний функциональной безопасности программного продукта;
- учет ограниченности ресурсов для обеспечения испытаний функциональной безопасности программного продукта;
- удостоверение достигнутой функциональной безопасности системы с программным продуктом.

Испытания производительности и динамического использования ресурсов ЭВМ программным продуктом:

- оценивание нарушения временного баланса между длительностью решения задач и производительностью ЭВМ;
- оценивание предельной пропускной способности системы с конкретным программным продуктом;
- определение качества динамического функционирования программного продукта при перегрузках.

Рис. 10.1

Специалисты третьей стороны имеют право на расширение условий испытаний в пределах требований на программный продукт и нормативной документации, при которых должно обеспечиваться заданное качество и безопасность результатов его применения. При этом в качестве первой стороны в процессе сертификации выступают разработчики или поставщики комплексов программ и их компонентов, а второй стороной являются заказчики, потребители или пользователи. Одна из этих двух сторон может выступать инициатором – **заявителем на сертификационные испытания.**

Результатом положительных испытаний является **сертификат соответствия** – документ, изданный по правилам Системы сертификации, удостоверяющий, что обеспечивается необходимая уверенность в том, что должным образом идентифицированная продукция соответствует конкретным стандартам и/или другим нормативным документам. Для удостоверения качества программных продуктов и их компонентов, заказчик может потребовать сертифицировать технологические процессы, обеспечивающих их жизненный цикл. Поэтому могут рассматриваться совместно задачи сертификации **конечных объектов – программных продуктов**, а также **технологий и систем качества**, обеспечивающих их создание и совершенствование (см. часть 2).

В исходных нормативных документах и требованиях по сертификации должны быть сосредоточены **все функциональные и эксплуатационные характеристики**, обеспечивающие заказчику и пользователям возможность корректного применения сертифицированного объекта во всем многообразии его функций и характеристик качества. Для особенно важной продукции, например, программных продуктов по государственным заказам для оборонной техники, результаты положительной сертификации технологии производства и системы качества, могут использоваться заказчиком как основание для выдачи лицензии на поставку соответствующей продукции по госзаказу. Такая **лицензия дает преимущество поставщику программных продуктов** при конкурсах на производство определенной продукции и на заключение контракта на ее поставку.

Испытания для сертификации проводятся в проблемно-ориентированных, технически компетентных, испытательных лабораториях, **аккредитованных на право проведения тех испытаний**, которые предусмотрены в ее нормативных документах. Такие про-

верки могут проводиться по графику или вследствие важных изменений технологии и системы качества предприятия, процессов ЖЦ и качества продукции, а также после проведения корректирующих действий требований заказчика к версии программного продукта.

Ресурсы для сертификации программных продуктов должны выделяться в зависимости от характеристик испытываемого продукта. Определяющими ресурсами сертификации обычно являются: возможная трудоемкость и длительность испытаний, совокупная численность и структура коллектива специалистов – сертифицированных, а также их квалификация и подготовленность к коллективной проверке конкретного типа программных продуктов и его компонентов. На основании испытаний оцениваются полученные результаты и обосновываются выводы о соответствии или несоответствии продукции требованиям нормативных документов. Протоколы испытаний представляются в орган по сертификации, а также заявителю по его требованию.

Заявитель для оценивания продукции или процесса, подлежащих сертификации, направляет в орган по сертификации заявку по форме, принятой в Системе сертификации. Орган по сертификации проводит **работу по подготовке и организации сертификации** продукции по заявке. Заявитель должен создать **условия и представить документы для обеспечения процессов испытаний**. Он может представить в орган по сертификации протоколы испытаний, проведенных при разработке и постановке продукции на производство, документы об испытаниях, выполненных сторонними испытательными лабораториями и другие документы, свидетельствующие о соответствии продукции установленным требованиям. На основе анализа представленных с заявкой документально подтвержденных доказательств соответствия его продукции установленным требованиям, орган по сертификации может принять решение о сокращении объема испытаний или о выдаче сертификата.

Процесс сертификации программных продуктов включает:

- анализ и выбор разработчиком или заказчиком (заявителем), компетентных в данной области органа и аттестованной лаборатории для выполнения сертификационных испытаний;
- подачу заявителем заявки на испытания в орган сертификации и принятие сертифицированными решения по заявке, выбор схемы сертификации, заключение договора на сертификацию;

- идентификацию требований к версии программного продукта, подлежащих испытаниям;
- выполнение сертификационных испытаний версии программного продукта сертификационной лабораторией;
- анализ полученных результатов и принятие решения лабораторией и/или органом сертификации о возможности выдачи заявителю сертификата соответствия;
- выдачу органом сертификации заявителю – сертификата и лицензии на применение знака соответствия и на выпуск сертифицированной продукции – версий программного продукта;
- осуществление инспекционного контроля органом сертификации сертифицированной продукции;
- проведение заявителем корректирующих мероприятий при нарушении соответствия продукции установленным требованиям и при неправильном применении знака соответствия.

Удостоверение качества продукта должно быть обеспечено необходимыми ресурсами для выполнения Программы испытаний, методиками планирования и разработки частных процедур проверок. Следует указывать технические и программные средства, используемые во время проведения испытаний, и порядок проведения испытаний, а также ожидаемые результаты проверок. Должны быть разработаны методики контроля за корректировками, действиями по исправлению дефектов, если в службу управления продуктом поступит такой запрос. Служба управления Программами испытаний должна разработать методики сохранения конфиденциальности любой информации об испытаниях, а также данных имеющихся у экспертов.

Протоколы испытаний представляются заявителю и в орган по сертификации. Заявитель может представить в орган по сертификации протоколы испытаний с учетом сроков их действия, проведенных при разработке и постановке продукции на производство, или документы об испытаниях, выполненных отечественными или зарубежными испытательными лабораториями, аккредитованными или признанными в Системе сертификации. На основании протоколов сертификационных испытаний оцениваются полученные результаты и обосновываются сделанные выводы о соответствии или несоответствии продукции требованиям нормативных документов.

Заключение по результатам сертификационных испытаний разрабатывается сертифицированными и содержит обобщенные сведения

о результатах испытаний и обоснование целесообразности выдачи сертификата. Орган по сертификации после анализа протоколов испытаний, анализа документации, указанной в решении по заявке, осуществляет оценку соответствия продукции установленным требованиям, **оформляет сертификат** на основании заключения экспертов и регистрирует его. При внесении изменений в конструкторскую или эксплуатационную документацию, которые могут повлиять на систему качества или программный продукт, удостоверяемые при сертификации, заявитель должен известить об этом орган по сертификации, для принятия решения о необходимости проведения дополнительных испытаний. После регистрации сертификат вступает в силу и направляется предприятию-заявителю. Одновременно с выдачей сертификата предприятию-заявителю может выдаваться **лицензия** на право применения знака соответствия.

За сертифицированными программными продуктами в процессе их эксплуатации в течение всего срока действия сертификата соответствия должен осуществляться **инспекционный контроль**. Инспекционный контроль проводится в форме периодических и внеплановых проверок соблюдения требований к качеству сертифицированной продукции. По результатам инспекционного контроля **орган по сертификации может приостановить или отменить действие сертификата** и аннулировать лицензию на право применения знака соответствия в случае несоответствия продукции требованиям нормативных документов, контролируемых при сертификации, а также в случаях:

- принципиальных изменений модели зрелости, профиля стандартов, нормативных документов на продукцию или метода испытаний;
- изменения конструкции (состава), комплектности продукции;
- изменения организации или технологии разработки и производства;
- невыполнения требований технологии, методов контроля и испытаний, системы качества, если перечисленные изменения могут вызвать несоответствие продукции требованиям, контролируемым при сертификации.

Решение о приостановлении действия сертификата и лицензии на право применения знака соответствия не принимается в том случае, если путем корректирующих мероприятий, согласованных с ор-

ганом по сертификации, его выдавшим, заявитель может устранить обнаруженные причины несоответствия и подтвердить без повторных испытаний в аккредитованной лаборатории, соответствие продукта или процессов нормативным документам. Если этого сделать нельзя, то действие сертификата отменяется, и лицензия на право применения знака соответствия аннулируется. Информация о приостановлении или отмене действия сертификата доводится органом по сертификации, его выдавшим, до сведения заявителя, потребителей и других заинтересованных организаций.

Программа и методики испытаний комплексов программ на соответствие требованиям

Успешная и рентабельная Программа испытаний *требует ясно-го видения целей* и четкого понимания различных параметров и/или этапов использования Программы. Программа сертификационных испытаний может ориентироваться на тестирование только конечного программного продукта с акцентом на его базовые характеристики качества или на последовательные, поэтапные испытания укрупняющихся программных компонентов, усложнение тестов и объектов внешней среды – рис. 10.2. Подробное *изучение системных требований* или сценариев использования системы вместе с тщательным определением параметров Программы испытаний и требований к тестам необходимы для эффективного тестирования продукта. Планирование и реализация испытаний – не единовременный акт, а *процесс*.

Эффективная Программа испытаний, включающая в себя автоматизацию тестирования программного продукта, должна иметь свой *собственный жизненный цикл*, в который входят планирование стратегий и целей, определение требований к тестам, анализ, проектирование и кодирование тестов.

По аналогии с процессом, которому следуют при разработке комплекса программ, необходимо определить жизненный цикл тестов прежде, чем они будут спроектированы. *Ресурсы Программы испытаний ограничены*, а способов тестирования систем много. Представление работ по испытаниям в графической форме, позволяет специалистам оценивать границы и масштаб Программы тестирования.

**Поэтапная автоматизация испытаний
сложных комплексов программ**



Рис. 10.2

Завершив анализ, испытателям следует переходить к созданию моделей реализации и оформлению Программы, которая в графической форме описывает ее масштаб. Структуру Программы испытаний обычно представляют двумя способами.

Один метод организации тестовых процедур, известный как архитектура тестирования *на базе архитектуры системы*, разбивает тестовые процедуры по функциям и компонентам системной архитектуры, по логическому принципу приоритетов в ее иерархии. Второй метод архитектуры тестирования *на базе выбранных методов* связывает тестовые процедуры с различными методами испытаний, представленными в модели Программы тестирования.

Оценивание качества и *соответствия требованиям* программного продукта при испытаниях проводится комиссией заказчика, в которой участвует руководитель (главный менеджер) разработки и некоторые ведущие разработчики, или аттестованной сертификационной лабораторией. *Комиссия при испытаниях должна руководствоваться следующими документами* (см. рис. 10.1):

- утвержденными заказчиком и согласованными с разработчиком контрактом, техническим заданием и спецификациями требований на программный продукт и систему;
- действующими государственными и ведомственными стандартами на жизненный цикл и испытания крупных комплексов программ, на технологическую и эксплуатационную документацию, а также стандартами де-факто, согласованными с заказчиком для использования – профилем стандартов и нормативных документов;
- Программой испытаний по всем требованиям контракта, технического задания и спецификаций;
- методиками испытаний и матрицей тестов, охватывающими каждый раздел требований технического задания, спецификаций и Программы испытаний;
- комплектом адекватной эксплуатационной и технологической документации на программный продукт.

Кроме того, в документе должны быть представлены план-график тестирования и *матрица трассирования тестов к требованиям спецификаций* на программный продукт или на его функциональные компоненты, а также субподрядчики, принимающие участие в тестировании, их роль и ответственность.

Программа испытаний является серией экспериментов и должна разрабатываться с позиции необходимого объема тестирования в процессе проведения испытаний для проверки выполнения всех требований технического задания и соответствия предъявленной документации. Программа испытаний, методики их проведения и оценки результатов, разработанные совместно заказчиком и разработчиком, должны быть согласованы и утверждены. Они должны содержать **уточнения и детализацию требований** технического задания и спецификаций для данного программного продукта, а также гарантировать корректную проверку всех заданных функций и характеристик качества. **Программа испытаний должна содержать следующие четко сформулированные разделы:**

- объект испытаний, его назначение и перечень основных документов, определивших его разработку;
- цель испытаний, с указанием всех требований технического задания, характеристик качества, подлежащих проверке, и ограничений на проведение испытаний;
- собственно Программу испытаний, содержащую проверку комплектности спроектированного программного продукта в соответствие с техническим заданием, план и график проведения тестирования для проверки по всем разделам требований технического задания и дополнительных требований, формализованных отдельными решениями разработчиков и заказчика;
- перечень методик испытаний, однозначно определяющие все требования, понятия проверяемых функций и характеристик качества, условия и сценарии тестирования, инструментальные средства, используемые для испытаний;
- перечень методик обработки и оценки результатов тестирования по каждому разделу Программы испытаний.

Методика испытаний должна содержать: описание организации и матрицу процесса тестирования, тестовые варианты, сценарии и процедуры, которые используются при испытаниях отдельного функционального компонента или комплекса программ в целом. Каждый тест должен иметь уникальный для данного проекта идентификатор; должны быть представлены инструкции для проведения процедур тестирования; описание аппаратуры и инструментальные средства для реализации тестирования, а также инструкции для динамического и регрессионного тестирования. Кроме того, должны быть приведены ссылки на соответствующие проверяемые

требования, указаны условия выполнения (конфигурация аппаратуры и компонентов комплекса программ), входные данные, эталонные и ожидаемые результаты, критерии оценки качества результатов, процедура проведения тестирования для каждого тестового варианта, допущения и ограничения.

Большой объем разнородных данных, получаемых при испытаниях крупных комплексов программ, и разнообразие возможных способов их обработки, интерпретации и оценки приводят к тому, что важнейшими факторами становятся **методики обработки и оценки результатов, а также протоколы проверки по пунктам Программы испытаний**. В соответствии с методиками испытаний, средства автоматизации должны обеспечивать всю полноту проверок характеристик по каждому разделу методик. **Результаты испытаний фиксируются в протоколах** (см. стандарт **ISO 12119** и рис.10.1), которые обычно содержат следующие разделы:

- назначение тестирования и раздел требований технического задания, по которому проводились испытания;
- указания разделов методик в соответствии, с которыми проводились испытания, обработка и оценка результатов;
- условия и сценарии проведения тестирования и характеристики исходных данных при динамическом испытании;
- обобщенные результаты испытаний с оценкой их на соответствие требованиям технического задания и другим руководящим документам, а также технической и эксплуатационной документации;
- описание отличий тестовой и реальной эксплуатационной среды;
- описание обнаруженных дефектов и ошибок и рекомендуемых улучшений в испытываемом комплексе программ;
- выводы о результатах испытаний и о соответствии созданного программного продукта или его функционального компонента определенному разделу требований технического задания и исходных спецификаций.

Протоколы по всей программе испытаний **обобщаются в акте**, в результате чего делается **заключение о соответствии системы требованиям** заказчика и о завершении работы с положительным или отрицательным итогом. При выполнении всех требований технического задания заказчик обязан принять программный продукт, и проект считается завершенным.

Испытания надежности функционирования программного продукта

В составе требований к системам и программным продуктам, функционирующим в реальном времени, особое значение имеют **динамические функции и характеристики**. Для обеспечения их высокого качества недостаточны отдельные сценарии и процедуры тестов, а необходимо создавать **потоки динамических тестов в реальном времени**, адекватные соответствующим данным при функционировании внешней среды систем и/или пользователей (см. рис. 10.1). Эти потоки тестов должны обеспечивать динамическую проверку комплексов программ на соответствие требованиям, выявление дефектов и ошибок в реализации их функций и характеристик в реальном времени. Основная особенность такого тестирования состоит в создании динамической среды функционирования программного продукта, максимально приближенной к реальной, при его практическом применении.

Задача состоит в определении соответствия требованиям, функций и характеристик программного продукта при различной интенсивности потоков тестов, адекватных нормальным условиям применения программного продукта, а также критическим по составу и интенсивности, для выявления предельных условий его работоспособности. Такие **условия тестирования отражаются на интегральных характеристиках**, на снижении надежности и/или безопасности, а также на повышении рисков применения программного продукта. Для комплексов программ реального времени **особое значение могут иметь** причины и методы уменьшения **рисков надежности и производительности**, вследствие дефектов и ошибок, а также при формировании и реализации требований к этим характеристикам программных продуктов. Эти взаимосвязанные характеристики качества программного продукта зависят от одних и тех же свойств воздействий из внешней среды, требуют совместного анализа и методов для выявления и устранения дефектов. Локализация и устранение таких динамических дефектов обычно осуществляется вне реального времени, путем применения детерминированных сценариев и тестовых процедур, а иногда за счет изменения требований заказчика.

Оценивание надежности программных комплексов включает измерение количественных характеристик: завершенности, устойчи-

вости к дефектам, восстанавливаемости и доступности-готовности (см. лекцию 7). При этом предполагается, что в контракте, техническом задании или спецификации требований зафиксированы, и утверждены заказчиком определенные значения этих характеристик и их приоритеты. Измерения проводятся при функционировании готового программного продукта для сопоставления с заданными требованиями и для оценивания рисков соответствия этим спецификациями требований. Тестирование для оценки надежности комплекса программ должно проводиться в тестовом окружении, которое максимально приближено к реальным условиям применения системы. Входные параметры тестов следует задавать на основе вероятностного распределения соответствующих характеристик или их наборов при эксплуатации программного продукта, исходя из частоты возможных сценариев работы пользователей или системы.

Значения надежности коррелированы с характеристикой корректность, однако можно достигать высокую надежность функционирования комплекса программ при относительно невысокой их корректности за счет сокращения времени восстановления при отказах. Кроме того, надежность можно оценивать косвенно в процессе разработки по прогнозируемой плотности обнаружения скрытых дефектов и ошибок, а также по плотности выявляемых и устраняемых ошибок выходных результатов при тестировании динамического функционирования комплекса программ. Степень покрытия тестами структуры функциональных компонентов и комплекса программ в целом может служить *ориентиром для прогнозирования их потенциальной надежности*. Распределение реальных длительностей и эффективности восстановления при ограниченных ресурсах для функционирования программ, может рассматриваться как дополнительная составляющая при оценивании надежности.

Для прямых, количественных измерений надежности необходимы инструментальные средства, встроенные в операционную систему или в соответствующие компоненты комплекса программ. Эти средства должны в динамике функционирования программного продукта, автоматически селектировать и регистрировать аномальные ситуации, дефекты и искажения вычислительного процесса, программ и данных, выявляемые аппаратным, программно-алгоритмическим контролем или пользователями. Накопление и систематизация проявлений дефектов при исполнении программ позволяет оценивать основ-

ные показатели надежности, помогает определять причины сбоев и отказов и подготавливать данные для повышения надежности программных продуктов.

Прямые экспериментальные методы оценивания интегральных характеристик надежности (безопасности и рисков) программных продуктов, в ряде случаев весьма трудно реализовать при нормальных штатных условиях функционирования крупных комплексов программ, из-за больших значений времени наработки на отказ (сотни и тысячи часов), которые необходимо достигать при разработке и фиксировать при испытаниях. Сложность выявления и регистрации редких отказов, а также высокая стоимость экспериментов при длительном многосуточном функционировании крупных комплексов программ приводят к тому, что на испытаниях получаются малые выборки зарегистрированных отказов и низка достоверность оценки надежности. Кроме того, при таких экспериментах трудно гарантировать полную представительность выборки исходных данных, так как проверки определяются конкретными условиями применения данного программного продукта на испытаниях.

При испытаниях надежности в первую очередь обнаруживаются **отказы – потери работоспособности**. Однако в большинстве случаев первоначально остается неизвестной причина происшедшего отказа. Для выявления фактора, вызвавшего отказ (первичной ошибки или дефекта) и устранения его причины, необходимо, прежде всего, определить, каким компонентом системы стимулирован данный отказ. Наиболее крупными источниками отказов являются частичные физические неисправности или сбои аппаратуры ЭВМ, а также дефекты и ошибки программных продуктов. Стабильные неисправности аппаратуры диагностируются достаточно просто, соответствующими аппаратными тестами, после чего должен следовать ремонт или замена определенных блоков. Однако при возникновении случайного отказа, после которого происходит автоматически полное восстановление нормального функционирования, во многих случаях трудно однозначно выявить его первичный источник, особенно при очень редких отказах.

Для диагностики и устранения случайных редких отказов должна быть организована **служба их регистрации** с максимально полным фиксированием характеристик ситуаций, при которых проявился каждый. Сбои в аппаратуре носят более или менее случайный харак-

тер и полное повторение отказовой ситуации маловероятно. Ошибки и дефекты программ содержатся в определенном месте и должны регулярно проявляться при полном повторении внешних ситуаций. На основе таких признаков и, по возможности, детального описания ситуаций возникновения отказа, могут строиться предположения о его причине. Для повышения надежности при высокой наработке на отказ необходима тщательная, систематическая работа специалистов, накапливающих, регистрирующих и анализирующих все отказовые ситуации при функционировании комплекса программ. Эти специалисты должны также регистрировать все проведенные корректировки для прогнозирования причин появления возможных, дополнительных источников отказов, вызванных дефектами корректировок.

Для выявления *тенденции изменения показателей надежности*, их зарегистрированные значения необходимо связывать во времени с моментами корректировки программ. Анализируя корреляцию между значениями надежности и процессом изменения программ, можно выявлять некоторые корректировки, которые содержат ошибки и снижают надежность. Получающиеся при этом показатели позволяют прогнозировать число ошибок, подлежащих исправлению для достижения требуемых значений надежности в зависимости от длительности испытаний. В результате может быть оценена наработка до следующего выявления ошибки или отказа.

При заключительных приемо-сдаточных и сертификационных испытаниях *для достоверного определения надежности* организуются многочасовые и многосуточные прогоны динамического функционирования комплекса программ в реальной и/или имитированной внешней среде в условиях широкого варьирования исходных данных с акцентом на стрессовые ситуации, стимулирующие проявления *угроз надежности*. Такие прогоны позволяют измерять достигнутые характеристики надежности и определять степень их соответствия требованиям технического задания, а также закреплять их в технических условиях и документации на программный продукт.

Если интенсивное тестирование программ в течение достаточно длительного времени не приводит к обнаружению дефектов или ошибок, то у специалистов, ведущих испытания, создается ощущение бесполезности дальнейшего тестирования программного продукта, и он передается на эксплуатацию. Экспериментальное исследование характеристик сложных ПС позволило оценить *темп обнаружения*

дефектов – риски, при котором крупные программные продукты передаются на регулярную эксплуатацию: 0,002–0,005 дефектов в день на человека, т.е. специалисты по испытаниям или все пользователи в совокупности выявляют только около одной ошибки или дефекта каждые два – три месяца использования ПС. Интенсивность обнаружения ошибок ниже 0,001 ошибок в день на человека, т.е. меньше одной ошибки в год на трех-четырех специалистов, непосредственно выполняющих динамическое квалификационное тестирование и/или эксплуатацию комплекса программ, по-видимому, может служить **эталонной высокой надежностью** обработки информации. Если динамическое функционирование программного продукта происходит непрерывно, то эти показатели соответствуют высокой наработке на обнаружение дефекта или отказа порядка 5–10 тысяч часов и коэффициенту готовности выше 0,99. При использовании этого критерия обычно учитывается календарное время испытаний, включающее длительность непосредственного тестирования, как для обнаружения, так и для локализации дефектов, а также длительность корректировки программ и других вспомогательных работ для восстановления нормального функционирования программного продукта.

Форсированные (стрессовые) испытания для оценивания надежности, а также функциональной безопасности и рисков программных продуктов значительно отличаются от традиционных методов испытаний аппаратуры. Основными факторами, влияющими на надежность ПС, являются исходные данные и их взаимодействие с дефектами и ошибками программ или сбоями в аппаратуре ЭВМ. Поэтому форсирование испытаний надежности осуществляется повышением интенсивности искажений исходных данных и расширением варьирования их значений, а также специальным увеличением интенсивности потоков информации и загрузки программ на ЭВМ выше нормальной.

Планирование форсированных динамических испытаний должно предусматривать последующий **пересчет полученных значений надежности** на условия нормального функционирования. Для этого необходимо оценивать надежность испытываемых программ в зависимости от интенсивности искажений данных или от характеристик перегрузки ЭВМ, а также применять способы корректного пересчета получаемых показателей на нормальные условия эксплуатации. При

форсированных испытаниях целесообразно выделять следующие **режимы тестирования**:

- полное искажение, предельные и критические значения ключевых параметров тестов каждого типа внешней информации и воздействий пользователей;
- предельные и критические сочетания значений различных взаимодействующих параметров тестов при эксплуатации программного продукта;
- предельно большие и малые интенсивности суммарного потока и каждого типа внешней информации;
- умышленные нарушения пользователями определенных положений инструкций и рекомендаций эксплуатационной документации на программный продукт.

Как вид форсированных испытаний можно рассматривать тестирование и контроль результатов функционирования одних и тех же ПС при увеличении числа испытываемых экземпляров и нормальных исходных данных – **Бета-тестирование**. На этапе тестирования в соответствии с эксплуатационной документацией, пользователями некоторого предварительного тиража программного продукта, происходит естественное расширение вариантов исходных данных, если они взаимно независимы. Это увеличивает наборы тестов и тем самым дает возможность оценивать наработки на отказ в сотни и тысячи часов. Они позволяют выявлять и устранять значительное число дефектов за относительно **небольшое календарное время** и тем самым доводить надежность до требуемого уровня. Однако следует учитывать, что при этом пропорционально возрастает суммарная трудоемкость таких испытаний.

Особым видом форсированных испытаний является целенаправленное тестирование эффективности средств оперативного контроля и восстановления программ, данных и вычислительного процесса **для оценивания восстанавливаемости**. При таких испытаниях основная задача состоит в оценивании качества динамического функционирования средств автоматического повышения надежности, и в измерении характеристик восстанавливаемости. Для этого имитируются запланированные условия функционирования программ, при которых в наибольшей степени стимулируется срабатывание средств программного рестарта и оперативного, автоматического восстановления работоспособности.

Следует особо отметить трудности достижения и регистрации надежности программ, характеризующейся наработкой на отказ $\gg 100$ ч. При такой надежности резко возрастают сложность обнаружения возникающих отказов и диагностирования их причин. Нарботка на отказ в тысячи часов в ряде случаев достигалась только при эксплуатации и сопровождении сложных программных продуктов в течение нескольких лет. При требовании особо высокой надежности функционирования, суммарные затраты ресурсов на ее достижение и оценивание могут увеличиваться на порядок. Однако требуемое увеличение затрат для получения такой высокой надежности программного продукта в процессе разработки трудно обеспечить практически. Поэтому для ее достижения, активно применяются различные методы программной защиты от сбоев и отказов программ (методы оперативного рестарта). Они позволяют замедлить рост затрат ресурсов на разработку при повышении требований к их надежности.

Оценивание изменения надежности комплексов программ реального времени, путем применения оперативного контроля и рестарта. В любых ситуациях функционирования сложных комплексов программ, прежде всего, должны исключаться катастрофические последствия и длительные отказы или в максимальной степени смягчаться их негативное влияние на результаты, выдаваемые пользователю. Неизбежность дефектов и ошибок в сложных комплексах программ, искажений исходных данных и аппаратурных сбоев приводит к необходимости регулярного контроля процесса исполнения комплекса программ, и сохранности данных. Предвидеть заранее все ситуации исполнения программ и протестировать при них крупные программные продукты оказывается невозможным из-за их огромного количества, поэтому применяются методы, которые направлены на ***оперативное обнаружение последствий*** дефектов и аномального функционирования программ, а также на автоматическое восстановление (рестарт) нормального вычислительного процесса и искаженных текстов программ и данных. Для обеспечения высокой надежности функционирования необходимо максимально быстро обнаруживать аномалии, достаточно точно классифицировать тип уже имеющихся и возможных последствий искажений, а также осуществлять мероприятия, обеспечивающие быстрое восстановление нормального функционирования программного продукта и системы.

Для снижения влияния негативных возмущений различных типов и происхождения на результаты, а также для защиты вычислительного процесса и информации программно-алгоритмическими методами в комплексы программ реального времени должна иметься **избыточность ресурсов**. Избыточность вычислительных ресурсов необходима, прежде всего, для селекции оперативных искажений процесса функционирования программного продукта и для выработки решений по снижению последствий этих аномалий. Основная цель использования избыточности состоит в ограничении или исключении возможности аварийных последствий от динамических возмущений, соответствующих отказу системы. Любые дефекты при исполнении программ необходимо блокировать и по возможным последствиям сводить до уровня сбоя путем оперативного автоматического восстановления. При этом не обязательно сразу устанавливать причины искажения, главная задача сводится к **максимально быстрому восстановлению** нормального функционирования и ограничению его негативных последствий.

Введение средств контроля функционирования и помехозащиты в программы позволяет **скомпенсировать влияние на надежность неполной корректности программных продуктов**, а также снизить негативные воздействия внешних возмущений различных типов. Однако только средствами контроля и обеспечения программной помехозащиты невозможно достигнуть высокой надежности динамического функционирования ПС. Требуемое сокращение вероятности отказа может достигаться путем **повышения затрат ресурсов на тестирование** и увеличения его длительности, что допускает соответствующее снижение затрат на средства обнаружения искажений и восстановление.

Контроль работоспособности комплекса программ, исправление дефектов и восстановление при отказовых ситуациях сокращают ресурсы времени функционирования ЭВМ, доступные для выполнения основных функций, и в общем случае негативно отражаются на производительности системы. Однако если некоторые ситуации контроля и восстановления проводятся достаточно быстро так, что их последствия не фиксируются как отказ и не отражаются на снижении работоспособности, то такие затраты времени полезны для решения функциональных задач и должны быть отнесены к улучшению качества программного продукта.

Испытания функциональной безопасности программного продукта

Функциональная безопасность программных продуктов и систем зависит от отказовых ситуаций, негативно отражающихся на работоспособности и реализации их основных функций, причинами которых могут быть дефекты и аномалии в аппаратуре, комплексах программ, данных или вычислительных процессах (см. рис. 10.1). При этом катастрофически, критически или существенно искажается процесс функционирования программных продуктов и/или систем, что наносит значительный ущерб при их применении. Основными источниками отказовых ситуаций могут быть некорректные исходные требования, сбои и отказы в аппаратуре, дефекты или ошибки в программах и данных функциональных задач, проявляющиеся при их динамическом исполнении в соответствии с назначением. При таких воздействиях, внешняя, функциональная работоспособность систем может разрушаться не полностью, однако невозможно полноценное выполнение заданных функций и **требований к программному продукту**. В реальных сложных системах, связанных с безопасностью, возможны катастрофические последствия и отказы функционирования с большим ущербом, при отсутствии воздействия лиц, заинтересованных в нарушениях работоспособности систем и ПС.

Понятия, методы тестирования и характеристики функциональной безопасности программных продуктов и систем близки к аналогичным для надежности. Поэтому способы оценки и испытаний функциональной безопасности могут базироваться на методах тестирования, определения и обеспечения надежности функционирования комплексов программ. При более или менее одинаковых источниках угроз и их проявлениях эти понятия можно разделить **по величине негативных последствий** и ущерба при возникновении отказовых ситуаций. Чем сложнее системы и чем выше к ним требования безопасности, тем неопределеннее функции и характеристики тестирования требований для обеспечения их безопасности. Неопределенности начинаются с требований заказчиков, которые при формулировке технического задания и спецификаций не полностью формализуют и принципиально не могут обеспечить достоверное содержание всего адекватного набора характеристик и значений требований безопасности, которые должны быть при завершении проекта и

предъявлении конечного программного продукта заказчику. Эти требования итерационно формируются, детализируются и уточняются по согласованию между всеми участниками проекта вследствие естественной ограниченности первичных исходных данных, и изменения их под влиянием объективных и субъективных воздействий со стороны различных процессов на последовательных этапах ЖЦ комплекса программ.

Всегда не полностью, с необходимой детализацией определены и описаны все характеристики, особенности функционирования и безопасности объектов внешней среды. Квалификация и субъективные свойства потребителей и пользователей изменяются по мере освоения функциональных возможностей системы и ее работоспособности, что увеличивает неопределенность ее реальной безопасности. Различия свойств персонала, применяющего систему, дополнительно увеличивают неопределенность значений безопасности и трудности ее прогнозирования при тестировании с учетом множества *субъективных факторов различных специалистов, участвующих в эксплуатации*.

При анализе характеристик функциональной безопасности целесообразно выделять и учитывать особенности *классов систем и их программных продуктов*. Первый класс составляют системы, имеющие встроенные комплексы программ жесткого регламента реального времени, автоматизировано управляющие внешними объектами или процессами. Время необходимой реакции на отказовые ситуации таких систем обычно исчисляется секундами или долями секунды, и процессы восстановления работоспособности должны проходить за это время, в достаточной степени автоматизировано (бортовые системы в авиации, на транспорте, в некоторых средствах вооружения, системы управления атомными электростанциями). Системы второго класса, применяются для управления процессами и обработки деловой информации из внешней среды, в которых активно участвуют специалисты-операторы (банковские, административные, штабные военные системы). Допустимое время реакции на опасные отказы в этих системах может составлять десятки секунд и минуты, и операции по восстановлению работоспособности частично могут быть доверены специалистам-администраторам по обеспечению функциональной безопасности.

Тестирование и обеспечение функциональной безопасности сложных систем должны решаться с учетом одновременного динамического развития всех компонентов внешней среды, и факторов, непрерывно изменяющихся и воздействующих на результаты их решения. Эти факторы влияют на **неопределенность критериев, методов оценивания значений эффективности тестирования и функциональной безопасности** конкретных программных продуктов и систем. Существующие технологии тестирования способствуют повышению функциональной безопасности, снижению потенциального ущерба и рисков, однако практически всегда остается открытым вопрос, насколько применяемые методы оправдывают затраты на реализацию требований заказчика. Испытания, эксплуатация и сертификация способствуют снижению неопределенности оценок эффективности и итерационному приближению к практически приемлемому уровню функциональной безопасности программных продуктов.

Роль негативных воздействий и их разрушительные последствия быстро возрастают в связи с ростом сложности разработки и применения современных систем на базе ЭВМ и ответственности решаемых ими задач. Одновременно возрастает сложность внешней и операционной среды, в которой функционируют комплексы программ и **ответственность функций систем, связанных с безопасностью**. Объективное повышение сложности функций, реализуемых программными продуктами в современных системах, непосредственно приводит к увеличению их объема и трудоемкости создания. Соответственно росту сложности программ возрастает относительное и абсолютное количество выявляемых и остающихся в них дефектов и ошибок, что **отражается на снижении потенциальной безопасности их функционирования**.

Работоспособность программных продуктов может быть обеспечена при исходных данных, которые использовались при их разработке, отладке и испытаниях. Реальные исходные данные могут иметь значения, отличающиеся от заданных техническим заданием и от используемых при эксплуатации программ и баз данных. При таких исходных данных функционирование программного продукта трудно предсказать заранее, и весьма вероятны различные аномалии, завершающиеся отказами, отражающимися на безопасности. Это приводит к практической невозможности достоверных априорных

аналитических оценок функциональной безопасности комплексов программ при ее высоких значениях.

*Достижение требуемой функциональной безопасности систем, содержащих программные продукты реального времени, решается путем использования современных регламентированных технологических процессов динамического тестирования, подобным применяемым при обеспечении надежности. Они должны быть поддержаны группой международных стандартов, определяющих состав и процессы выполнения требований к заданной функциональной безопасности систем и комплексов программ. Для систематической, координированной борьбы с угрозами безопасности программных продуктов необходимы исследования факторов, влияющих на функциональную безопасность со стороны случайных дефектов и ошибок, существующих и потенциально **возможных в конкретных системах и комплексах программ.***

Требования к функциям систем и программных продуктов, а также к безопасности их функционирования должны соответствовать доступным ресурсам для их реализации с учетом допустимого ущерба – рисков вследствие отказов при неполном выполнении требований. *Ограниченности ресурсов* различных видов для тестирования и обеспечения функциональной безопасности значительно влияют на технико-экономические показатели, качество и функциональную безопасность всей системы и ПС. В результате сложность комплексов программ, а также доступные ресурсы для их реализации становятся косвенными критериями или факторами, влияющими на выбор методов разработки, на достигаемую безопасность программных продуктов.

Разработку систем должны завершать *сертификационные испытания и удостоверение достигнутой функциональной безопасности и надежности* систем с программным продуктом, предусматривающие возможность совершенствования их характеристик путем соответствующих корректировок программ. Повышение функциональной безопасности целесообразно также путем анализа выявленных дефектов и *оперативного восстановления вычислительного процесса, программ и данных* (рестарта) после обнаружения аномалий и отказов функционирования программного продукта. Этому может способствовать накопление, мониторинг и хранение данных о

выявленных дефектах, сбоях и отказах в процессе исполнения программ и обработки данных.

Испытания производительности и динамического использования ресурсов ЭВМ программным продуктом

Оценивание ресурсной эффективности состоит в измерении количественных характеристик: временной эффективности и используемости динамических ресурсов ЭВМ комплексом программ (см. рис. 10.1). При этом предполагается, что в контракте, техническом задании и спецификации требований зафиксированы и утверждены требуемые значения этих характеристик и их приоритетов. Оценивание динамических характеристик программ в реальном времени может проводиться при функционировании готового программного продукта или расчетными методами, при разработке для сопоставления с заданными требованиями и оценки *степени соответствия этим требованиям*.

Цель испытаний производительности – продемонстрировать заказчику, что в реальном времени система функционирует в соответствии с требованиями, содержащимися в спецификациях на производительность и касающиеся приемлемого времени отклика при обработке заданного количества транзакций. При тестировании производительности в реальном времени должны применяться промышленные нагрузки, что позволяет предсказать поведение программного продукта и системы при реальной эксплуатации. Средства, обеспечивающее тестирование производительности, должны позволять оценивать влияние перегрузок. *Оценивание перегрузок* – это процесс тестирования работоспособности вычислительных машин при выполнении динамических сценариев большого потока данных *в реальном времени* с целью выяснения того, когда и где программный продукт выйдет из строя, работая под высокой нагрузкой. При тестировании перегрузок система подвергается предельным и максимальным нагрузкам для определения, выйдет ли она из строя и где это произойдет, а также для идентификации того, что выйдет из строя. Эти *допустимые пределы должны быть определены в системных требованиях* к программному продукту, где также должна определяться реакция системы на перегрузки. Данный вид тестирования не-

обходим для систем, работающих с максимальной спроектированной нагрузкой, для проверки того, что они в реальном времени динамически функционируют в соответствии с требованиями.

Адекватное проведение динамического **испытания программного продукта на перегрузки**, при использовании ручных методов подготовки тестов – дорого, трудоемко, неточно и занимает много времени. Необходимы средства автоматизированного тестирования, которые включают имитаторы нагрузки и позволяют испытателям динамически имитировать в реальном времени сотни и тысячи виртуальных пользователей или объектов, одновременно работающих с целевым программным продуктом. Не нужно никому присутствовать лично, чтобы запустить тесты или управлять ими; можно установить таймер, определив, когда следует запустить конкретный тест, и они могут выполняться без участия человека.

Для **измерения характеристик временной эффективности** необходимы инструментальные средства, встроенные в операционную систему или в соответствующий комплекс программ. Эти средства должны в динамике реального функционирования программного продукта регистрировать:

- загрузку вычислительной системы функционирующими программами;
- значения интенсивности потоков данных от конкретных внешних абонентов;
- длительность исполнения заданий при реализации конкретных функций;
- характеристики функционирования устройств ввода/вывода;
- время ожидания результатов (отклика) на функциональные задания пользователей или системы;
- заполнение памяти обмена с внешними абонентами в различных режимах применения программного продукта.

Значения этих характеристик зависят не только от свойств и функций комплекса программ, но также от особенностей архитектуры и операционной системы ЭВМ. Регулярная регистрация и обобщение таких данных позволяет выявлять ситуации, **негативно влияющие** в реальном времени на функциональную пригодность, надежность и другие важные характеристики программного продукта. Существует особый вид тестов для проверки удовлетворения специфических требований, предъявляемых к **параметрам производи-**

тельности программного продукта, когда делается попытка достижения количественных пределов, обусловленных характеристиками самой системы и ее операционного окружения. При этом может производиться динамическое тестирование в реальном времени с целью достижения наибольших **возможностей** по производительности, и выполнения функций программного продукта с повышением нагрузки, вплоть до достижения **запланированных характеристик требований**.

При излишне высокой интенсивности поступления исходных данных может **нарушаться временной баланс** между длительностью решения требуемой совокупности задач программным продуктом в реальном времени, и производительностью ЭВМ при решении этих задач. Также возможно нарушение баланса между имеющейся в ЭВМ памятью и памятью, необходимой для хранения всей поступившей и обрабатываемой информации. Для выявления подобных ситуаций и определения характеристик программного продукта в условиях недостаточности ресурсов ЭВМ проводятся испытания при высокой, но допустимой, в соответствии с требованиями, интенсивности поступления исходных данных.

Наиболее сложным является **оценивание эффективности использования ресурсов производительности ЭВМ в реальном времени**. При этом должна быть определена зависимость качества решения задач от интенсивности поступающей информации различных типов. Основная задача испытаний состоит в определении рисков – вероятностей, с которыми будет нарушаться соответствие между потребностями в производительности для решения всей требуемой совокупности задач и реальными возможностями ЭВМ и других компонентов системы. Если эта вероятность невелика, и можно считать допустимым эпизодическое снижение качества за счет получающихся задержек и пропусков в обработке сообщений или заданий, то делается вывод о соответствии производительности ЭВМ заданным функциям данного программного продукта.

При использовании комплексом программ производительности и памяти реализующей ЭВМ менее чем на 50%, разработчик может практически не учитывать эти ограничения и сопутствующие риски. Закономерным является стремление разработчиков программ применять, особенно для систем реального времени, встроенные объектные **ЭВМ с предельным использованием их технических характери-**

стик. Опыт создания ПС реального времени позволяет утверждать, что практически невозможно использовать производительность объектной ЭВМ более чем на 95%, и почти всегда целесообразно ограничиваться на уровне 80 – 90%, так как иначе, затраты на разработку программного продукта могут значительно увеличиться. Подобная зависимость обусловлена сложностью оптимального распределения в динамике ограниченных ресурсов ЭВМ (особенно производительности) по многим функциональным задачам, необходимостью проектирования комплекса программ с учетом этих ограничений и неоднократными переделками программных компонентов для того, чтобы соблюсти ресурсные ограничения.

Для оценивания характеристик использования производительности при тестировании крупных программных продуктов в реальном времени **должны быть измерены:**

- реальные значения интенсивностей поступающих исходных данных и заданий на вызов функциональных программ, а также распределения вероятностей этих интенсивностей для различных источников и типов заданий;
- длительности автономного решения отдельно каждой функциональной задачи, обрабатывающей исходные данные или включаемой внешними заданиями, а также периодически;
- загрузка ЭВМ в нормальном режиме поступления сообщений и заданий, а также вероятность перегрузки заданиями различных типов и возможные распределения длительностей перегрузки в реальных условиях;
- влияние пропуска в обработке заданий или сообщений каждого типа и снижения темпа решения определенных задач на функциональную пригодность и другие важные характеристики программного продукта.

Перечисленные задачи могут быть **решены экспериментально** в процессе тестирования в реальном времени законченного разработкой программного продукта, однако при этом **велик риск**, что производительность ЭВМ окажется недостаточной для решения заданной совокупности задач в реальном времени, что отразится на качестве и возможности использования системы. Кроме того, не всегда условия испытаний или опытной эксплуатации системы соответствуют режимам массового ее применения. Поэтому при оценивании требуется принимать специальные меры для создания реальных, а также кон-

тролируемых, наиболее тяжелых по загрузке в реальном времени условий функционирования комплекса программ и внешней среды.

Для корректного **оценивания предельной пропускной способности** системы в реальном времени с данным программным продуктом необходимо измерять следующие характеристики реализации функциональных программ в процессе разработки:

- экстремальные значения длительностей их исполнения и маршруты, на которых эти значения достигаются;
- среднее значение длительности исполнения каждой функциональной группы программ на всем возможном множестве маршрутов и его дисперсию;
- распределение вероятностей и значений длительности исполнения функциональных групп программ.

В общем случае для оценивания длительностей исполнения и определения **качества динамического функционирования программ в зависимости от загрузки**, необходимо оценивать вероятность каждой комбинации тестовых данных и измерять соответствующую ей длительность исполнения компонента программы. После упорядочения значений длительностей можно получить распределение вероятностей в зависимости от длительностей исполнения. Влияние таких ситуаций перегрузки ЭВМ по производительности, может быть ослаблено в реальном времени путем применения **приоритетных дисциплин** оперативной диспетчеризации исполнения заданий на решение функциональных задач.

Достоверность оценивания пропускной способности системы в реальном времени, с конкретным программным продуктом, зависит от корректности моделирования потоков внешних сообщений, а также от используемых распределений длительности исполнения программ. Для оценивания ресурсной эффективности, при подготовке технического задания и спецификаций требований **следует согласовывать с заказчиком модель и динамические характеристики внешней среды**, в которой будет применяться комплекс программ, а также динамику приема и передачи данных (см. лекцию 9).

Для определения использования комплексами программ ресурсов ЭВМ в реальном времени полезно применять рекомендации стандарта **ISO 14756**. Стандарт ориентирован на динамическое оценивание: программных продуктов, операционных систем и вычислительных комплексов, включающих аппаратные и программные сред-

ства. Описание метода измерения производительности начинается с имитации пользователей и потоков данных из внешней среды: их случайных временных характеристик и процессов; функционирования терминалов; установления параметров рабочих нагрузок пользователей и вычислительных средств.

Оценивание величины производительности в реальном времени рекомендуется для определения загрузки операторов-пользователей, пропускной способности программных продуктов по числу задач в единицу времени, временной шкалы событий обработки заданий и данных. Эти результаты предлагается **сравнивать с требованиями** заказчика и пользователей для оценивания допустимых рабочих нагрузок и достаточности производительности в реальном времени в конкретной внешней среде. Детальные процедуры измерений и оценивания в стандарте распределены **по шести разделам**:

- исходные требования;
- процессы измерений;
- результирующие данные;
- проверка корректности результатов;
- расчеты производительности;
- оценивание достоверности измерений производительности.

Если предварительно в процессе проектирования производительность системы в реальном времени не оценивалась или определялась слишком грубо, то **велик риск**, что доработки будут большими или может понадобиться заменить ЭВМ на более быстродействующую. Это обусловлено, как правило, «**оптимизмом**» разработчиков, что приводит к **занижению интуитивных оценок длительностей решения функциональных задач** и возможных предельных интенсивностей потоков внешней информации. Длительная регистрация и накопление значений ресурсной эффективности способствуют выявлению ситуаций, при которых проявляются некоторые динамические дефекты функциональной пригодности.

Лекция 11

УДОСТОВЕРЕНИЕ КАЧЕСТВА И ЗАВЕРШЕНИЕ СЕРТИФИКАЦИОННЫХ ИСПЫТАНИЙ ПРОГРАММНЫХ ПРОДУКТОВ

Испытания для сокращения и ликвидация опасных рисков при применении программных продуктов

При сертификационных испытаниях основные характеристики программного продукта условно можно разделить на две группы: функциональные характеристики, надежность, безопасность и производительность, и обобщенные характеристики качества. В предыдущей лекции представлены испытания функциональных характеристик и основной группы динамических характеристик программных продуктов. Обобщенные их свойства проявляются при испытаниях *рисков* и *эксплуатационных характеристик* продукта, которые рассмотрены ниже в этой лекции. Сертификационные испытания, отражающие качество программного продукта с наиболее общих позиций, состоят в определении рисков при его функционировании и качества эксплуатационной документации для пользователей. В лекции 7 отмечалось, что требования к программным продуктам можно оценивать с двух позиций: с *положительной* – при анализе качества реализации требований к функциям и характеристикам продукта, и с *негативной* – при определении, допустимого ущерба – риска применения программного продукта (рис. 11.1) [1].

Часто заказчики и разработчики первоначально устанавливают требования к каждой функции и характеристике программного продукта без учета относительных затрат на их достижение, а также без детального анализа их совместного влияния на полную *функциональную пригодность и риски у потребителей или для системы*. Это может приводить к несбалансированным значениям требований к отдельным, взаимосвязанным характеристикам, на которые не рационально используются ограниченные ресурсы ЖЦ комплекса программ, или к не адекватно низким их значениям.

В крупных проектах это может угрожать значительным повышением стоимости и/или снижением конкурентоспособности созда-

ваемого программного продукта из-за недостаточного уровня качества отдельных функций и характеристик.

Удостоверение качества и завершение сертификационных испытаний программных продуктов:

- испытания для сокращения и ликвидации опасных рисков при применении программного продукта;
- идентификации уровня допустимого интегрального риска программного продукта;
- изменение требований к функциям и характеристикам программного продукта для снижения риска;
- испытания эксплуатационной документации на соответствие требованиям к программному продукту:
 - испытания документов на соответствие требованиям к функциям и характеристикам программного продукта;
 - испытания эксплуатационной документации на практическую применимость;
- завершение сертификационных испытаний программного продукта;
- документирование процессов и результатов сертификации программного продукта:
 - базовые нормативные документы качества производства программного продукта;
 - исходные документы, характеризующие конкретный программный продукт и его предварительные испытания;
 - отчетные документы испытателей, отражающие результаты сертификационных испытаний программного продукта;
- поставка для применения пользователями сертифицированной версии программ продукта;
- анализ результатов сертификации и усовершенствование процессов испытаний программного продукта.

Рис. 11.1

Характеристики программных комплексов имеют различные меры, вследствие чего они в большинстве своем непосредственно **не сопоставимы между собой**. Они предварительно выбираются и устанавливаются заказчиком при последовательном, почти независимом анализе каждого требования, для использования в контракте и техническом задании. Для обобщенного оценивания качества про-

граммного продукта необходим учет относительного влияния каждой характеристики на функциональную пригодность. При этом не всегда удается учитывать ресурсы для их реализации в конкретном комплексе программ. Это часто приводит к выдвиганию **ряда не рациональных требований**, которые значительно отличаются: либо по степени влияния на функциональную пригодность, либо по величине ресурсов, необходимых для их реализации. **Для целенаправленного эффективного управления сложным проектом** целесообразно иметь механизм объединения разнородных характеристик в некоторый интегральный показатель, отражающий их совокупное влияние на его функциональную пригодность. Таким образом, при разработке проекта выявилась проблема анализа системной эффективности программного продукта и обобщения его свойств **в виде рисков**, а также оценивания совместного влияния конструктивных характеристик на функциональную пригодность с учетом затрат на их реализацию.

Оценивание **опасности рисков программного продукта и выбор эффективных контрмер** для их сокращения включает:

- определение возможных последствий, уровней потенциальных опасностей – угроз, и приоритетов каждого класса и категории рисков проекта комплекса программ;
- выделение и упорядочение ограниченной группы наиболее опасных, высокоприоритетных рисков проекта комплекса программ;
- планирование методов и необходимых ресурсов для реализации эффективных контрмер при сокращении каждой категории опасных, приоритетных рисков проекта;
- анализ, определение стратегии и распределение ресурсов на контрмеры для сбалансированного сокращения интегрального риска проекта комплекса программ с учетом приоритетов опасных рисков;
- распределение ответственности специалистов за появление и/или реализацию сокращения конкретных опасных рисков проекта.

Для учета влияния рисков на функциональную пригодность и другие характеристики программного продукта, а также для выбора рациональных контрмер целесообразно ранжировать допустимые риски **относительными величинами – приоритетами**. Величины и вероятности проявления рисков, а также ресурсы на контрмеры для их сокращения, желательно оценивать соизмеримыми экономическими, стоимостными категориями, или унифицированными относительными количественными величинами – приоритетами. Аналогич-

но, по такой же шкале экспертами целесообразно оценивать относительные затраты ресурсов, которые следует выделять на реализацию сокращения рисков. Для каждого вида рисков отношение коэффициента влияния на функциональную пригодность к относительным затратам на его достижение можно рассматривать, как обобщенный приоритет этому сокращению риска. Подобные рейтинги рисков с оценкой их вероятностей и последствий, особенно необходимы для оценивания, сбалансированного прогнозирования и последовательной минимизации интегральных рисков, а также для **мониторинга различных контрмер** в проектах комплексов программ. Интегральный риск проекта можно оценивать как результат обобщения всех видов рисков с учетом их относительного влияния на функциональную пригодность и другие важнейшие характеристики системы и программного продукта.

Для конкретного программного продукта состав и значения приоритетов следует **поэтапно адаптировать и уточнять** с учетом их назначения и функций. Наивысший приоритет можно интерпретировать как обязательное выполнение разработчиком соответствующего требования к указанной функции или характеристике с отсутствием риска. Такие, экспертные, качественные оценки позволяют прогнозировать и выявлять наиболее крупные и опасные риски, их долю в интегральном риске проекта комплекса программ и системы, а также рентабельность контрмер для их снижения.

Для **выбора критического уровня допустимых рисков** необходимо исследовать особенности проблемно-ориентированной системы, возможную последовательность потенциально опасных событий, любые смягчающие факторы и характеристики, а также природу и частоту возможных негативных последствий идентифицированных угроз в программном продукте и в системе. При этом для сбалансированного снижения интегрального риска, может оказаться эффективным сравнительное, количественное или качественное ранжирование рисков (присвоение им приоритетов) специалистами, хорошо информированными в проблемной области применения соответствующих систем.

Используя **планы управления проектом и рисками**, менеджер должен осуществлять распределение ресурсов тестирования и контрмер, направленных на преодоление негативных случайностей при функционировании. Для обеспечения требуемого качества комплекса

программ необходима **организация контрмер** в процессе управления рисками. Задача менеджера рисков состоит в тестировании для выявления и идентификации источников рисков, противоречий требований характеристик и ресурсов для их реализации и в предложении заказчику и разработчикам рациональных и возможных контрмер, обеспечивающих сокращение рисков до допустимых пределов. Контрмеры для сокращения рисков можно разделить на три типа:

- сокращение или исключение **первичных причин – угроз**, дефектов и ошибок в компонентах и комплексе программ, обусловленных недостатками их требований, тестирования, разработки или модификации, отражающихся на рисках функциональной пригодности или характеристиках программного продукта;
- сокращение или ликвидация **уязвимости** компонентов и комплекса программ при воздействии на них угроз, дефектов и ошибок, путем тестирования и введения средств защиты для блокирования их возможного негативного воздействия на риски функционирования и применения программного продукта;
- непосредственное изменение и сокращение при тестировании последствий **проявления рисков** функциональной пригодности и характеристик комплекса программ, путем их оперативного обнаружения и ликвидации ущерба при сохранении (возможно) вызывающих их первичных источников и причин.

Для выработки плана анализа рисков и применения контрмер при их сокращении должна быть определена и документально установлена **методика применения последовательного анализа угроз, уязвимостей и изменения проявлений рисков**. Если менеджеры проекта имеют достаточно большой опыт работы, а процессы развития проекта регламентированы и ведут себя предсказуемо, количество дефектов и угроз последовательно должно убывать и степень риска в ЖЦ проекта должна уменьшаться. После того как менеджеры проекта идентифицируют возможные риски в жизненном цикле комплекса программ, а также уточнят тактику применения контрмер по сокращению их влияния, возникает необходимость в **идентификации уровня допустимости остаточного интегрального риска программного продукта**.

На этапах проектирования заказчик совместно с разработчиком должны уточнить исходные требования к характеристикам комплекса программ и ограничения для ресурсов, которые могут быть использо-

ваны для его реализации и применения. Эти требования и ограничения могут не полностью выполняться на последующих этапах ЖЦ ПС, что приводит к ущербу у заказчиков и пользователей. Причинами такого ущерба могут быть дефекты и ошибки, а также завышенные заказчиком требования к функциям и характеристикам, которые не могут быть реализованы при выделенных ресурсах, или недостаточное качество технологии и квалификация специалистов – разработчиков, исполняющих проект.

Для сокращения интегрального риска до допустимого значения возможно изменение требований к функциональным характеристикам и/или к используемым ресурсам в технологических процессах ЖЦ комплекса программ. При этом для обеспечения требуемой функциональной пригодности и минимального интегрального риска разработчиками возможно применение **первой стратегии контрмер сокращения рисков**:

- изменение соотношения между отдельными, достигаемыми функциональными характеристиками программного комплекса в пределах согласованных требований, зафиксированных в техническом задании и спецификациях;
- изменение соотношения между используемыми видами ресурсов в пределах, заданных исходными ограничениями на их применение.

Таким образом, при управлении контрмерами рисков по первой стратегии необходимо обеспечить соответствие требованиям, достигнутых функциональных и конструктивных характеристик, которые были первоначально установлены в процессе их разработки. Для этого, могут быть **перераспределены некоторые имеющиеся ресурсы** с целью реализации требуемых отдельных характеристик, и тем самым, в частности, уменьшены риски, с наибольшими значениями или ограничены их последствия. Однако если при первой стратегии не обеспечивается допустимый минимальный интегральный риск и требуемая функциональная пригодность, то **по согласованию с заказчиком, разработчиками** может применяться **вторая стратегия контрмер сокращения рисков**:

- пересмотр и изменение исходных требований к совокупности функциональных и конструктивных характеристик комплекса программ и уменьшение за счет этого результирующих значений рисков;

- пересмотр и изменение некоторых ограничений требуемых и используемых ресурсов и технологии обеспечения жизненного цикла программного комплекса для получения допустимых рисков.

При этой стратегии эффект может быть достигнут **пересмотром и снижением требований к некоторым функциям и характеристикам программного продукта**, определяющих наибольшие риски или увеличением отдельных доступных ресурсов и совершенствованием технологии, отражающихся на необходимом сокращении этих рисков. Если интегральные риски обусловлены недостаточной величиной одного из видов ресурса, то приходится перераспределять доступные ресурсы или искать заказчику способы увеличения некоторого, критического ресурса. В результате изменения характеристик комплекса и ресурсов, выделяемых на этапы их жизненного цикла, должны достигаться сбалансированные значения их рисков, и минимизироваться интегральный риск программного продукта и системы. В соответствие с их значениями следует откорректировать и утвердить обновленные, экономически и функционально оправданные, требования к функциям и характеристикам, используемым ресурсам и технологии проекта. При обеих стратегиях **наиболее стабильными должны быть требования к функциональной пригодности программного продукта**, изменения которых допустимы при исчерпании возможностей сокращения интегральных рисков за счет изменения характеристик качества, используемых ресурсов и других контрмер.

Умение оценивать и обрабатывать риски является основным в деятельности любого менеджера проекта крупного программного продукта. Для **менеджеров таких проектов эти навыки представляются важнейшими**, поскольку упущения в этой сфере могут катастрофически влиять и блокировать усилия, направленные на успешное выполнение проекта всей системы, использующей программный продукт. В некоторых случаях процессы анализа и сокращения рисков могут быть упрощены [14]. Для этого целесообразно выделять и контролировать только отдельные (2 – 3) риски отклонения от требований, наибольшие по величине последствий и по вероятности негативного проявления, и минимизировать возможный в результате ущерб для функциональной пригодности системы (например, риски надежности и производительности в комплексах программ реального времени).

Приоритет результатов выполнения контрмер и каждого предлагаемого изменения рисков комплекса программ, целесообразно оценивать по следующим критериям:

- насколько данное изменение риска может улучшить эксплуатационные характеристики и функциональную пригодность программного продукта в целом;
- каковы затраты на выполнение корректировок комплекса программ, вследствие изменения рисков при создании новой версии, и их распространение пользователям;
- какова срочность извещения пользователя о разработанной корректировке и целесообразно ли ее распространять до подготовки очередной версии программного продукта;
- для какого числа пользователей может быть полезно данное изменение риска;
- как данное изменение риска отразится на эксплуатации пользователями предыдущих версий программного продукта;
- насколько подготовка и внедрение данного изменения риска может отразиться на сроках создания очередной версии программного продукта.

Выше анализировалось и оценивалось преимущественно **изменение функциональной пригодности и снижение риска** при совершенствовании характеристик программного продукта. Однако для заказчика и пользователей доминирующее значение могут иметь номенклатура и особенности реализации некоторых основных функций комплекса программ, которые, как правило, требуют наибольших затрат и определяют основной эффект и риск от применения программного продукта, а также потенциальный уровень спроса на рынке.

Испытания эксплуатационной документации на соответствие требованиям к программному продукту

Эксплуатационная документация должна обеспечивать эффективное применение программного продукта и точно отражать его назначение, функции, характеристики и требования, для использования квалифицированными специалистами – пользователями. Для этого эксплуатационные документы необходимо тестировать на полное соответствие выполнения всей **совокупности требований на программный продукт**, согласованных между разработчиками и заказчиком. В результате эти документы можно использовать при производстве программных

продуктов как отдельный, независимый **третий эталон** (кроме требований и тестов - см. лекцию 7), **для квалификационного тестирования и испытаний** реализации функций и характеристик продукта. Практическое апробирование документов пользователями при применении комплекса программ, является практическим методом тестирования корректности реализации требований к программному продукту, завершающим цикл испытаний и контроля его качества. Разработчики документов должны обеспечивать комфортное и корректное применение комплекса программ пользователями, на основе ясного и непротиворечивого изложения в документах технологических процедур и операций для его штатного функционирования и получения требуемых результатов при эксплуатации.

Организация документирования должна определять стратегию, стандарты, процедуры, распределение ресурсов и планы создания, изменения и применения документов на комплекс программ. Для этого в общем случае должны быть выделены **специалисты**, которые обязаны планировать, утверждать, выпускать, распространять и сопровождать комплекты апробированных и утвержденных эксплуатационных документов. Они должны стимулировать разработчиков программных средств, осуществлять непрерывное, регламентированное документирование процессов и результатов своей деятельности, а также контролировать полноту и качество утвержденных эксплуатационных и отчетных документов.

Состав и содержание комплекта документов конкретного программного продукта, целесообразно адаптировать разработчиками к его особенностям и свойствам **на основе использования стандартов и типовых структур – шаблонов** для двух классов продуктов, которые в наибольшей степени различаются особенностями эксплуатации [15]. **Первый класс** составляют программные продукты автоматизированного управления динамическими объектами и процессами в реальном масштабе времени. В процессе их применения допускается минимальное вмешательство пользователями в процедуры управления применением, и необходим, соответственно, небольшой объем эксплуатационных документов, выделяемых из стандартизированного комплекта. Для продуктов **второго класса** (административные системы) возможно применение пользователями широкого набора процедур управления, которые должны быть регламентированы достаточно полным набором и подробным содержанием документов.

Документация администрирования при эксплуатации системы должна обеспечивать поддержку первичной инсталляции, безопасного функционирования и восстановления программ и данных после сбоев. Администратор системы и программного продукта должен быть информирован о всех изменениях функционирования устройств системы и внешней среды, могущих привести к сбою или возникновению аварийной ситуации, и предпринимать соответствующие профилактические действия. Для этого требуется полная информация о требованиях программному продукту, к компонентам системы и внешней среды, которые имеют свои особенности в управлении с помощью специальных программных средств, поддерживающих администрирование и управление системой. Каждый из документов административного управления должен не противоречить международным стандартам на коммуникацию, интерфейсы с пользователями и базами данных, на защиту и обеспечение безопасности информации.

Документация для оперативных пользователей программных продуктов в системе, их функционирования, обработки и анализа результатов должна обеспечивать взаимодействие пользователей с различными аппаратно-программными реализациями терминалов. Для этого необходима унификация концепции, архитектуры, функций и методов визуализации пользовательского интерфейса. Типовые *формы-шаблоны документов* и процедуры работы с ними, рассматриваемые как *объекты стандартизации*, относятся в основном к функциональному, оперативному уровню взаимодействия пользователей с системами и программными продуктами [15].

Стандарты **ISO 15910** и **ISO 18019** являются наиболее современными нормативными документами, регламентирующими процессы *создания эксплуатационной документации для пользователей крупных программных продуктов*. В них изложены детальные структуры планов документирования, ориентированных на разработчиков документов, соответствующих требованиям на программный продукт, и процедуры контроля реализации плана. Эксплуатационная документация должна проходить *испытания на достоверность*, которые должны быть спланированы и реализованы на базе требований к функциям и характеристикам, а также к применению эксплуатационных процедур реального программного продукта.

Стандарты представляют разработчикам документации методы определения и применения процесса документирования при создании кон-

кретного программного продукта. Основной работой является создание **комплексного плана разработки документации**, реализация которого обеспечивает достоверное документирование программного продукта. Стандарты также определяют виды информации, представляемой заказчиком разработчику документации для тестирования, проверки и распространения документации. Стандарты определяют реализацию процесса документирования, описанного в **ISO 12207**, и могут быть **адаптированы к условиям и требованиям конкретного проекта**. Минимальный состав документации определяется заказчиком (например, с использованием **ISO 12207** и/или **ISO 6592**), что должно быть учтено при разработке плана документирования.

Описание требований эксплуатационной концепции для системы управления и программного продукта должно содержать действия пользователя, необходимые для работы с системой, ее связи с существующими системами и процедурами. Его используют при создании соглашения между поставщиком, разработчиком и пользователями. Данный документ фиксирует текущее состояние системы и программного продукта, назначение, требования, возможности и ограничения в зависимости от режима или конкретного состояния эксплуатации и включает **описания**:

- конкретной эксплуатационной среды и ее характеристики;
- основных компонентов и функций программного продукта, системы и связей между ними;
- внешних интерфейсов программного продукта и системы;
- возможностей, функций и характеристик программного продукта и системы;
- состава эксплуатационного персонала, его организационной структуры, уровня технической подготовки, обязанностей, взаимодействия;
- форм регистрации обнаруженных дефектов и ошибок;
- соглашений о внесении изменений, возникающих в процессе сопровождения версии программного продукта;
- концепцию поставки новой или модифицированной версии программного продукта, эксплуатационный сценарий;
- информацию о взаимодействии пользователей, поставщика, разработчика и предприятия, осуществляющего поддержку программного продукта, во время эксплуатационного периода.

В обязанности документаторов входит обеспечение плодотворных **контактов заказчика с разработчиками** программного продукта, гарантирующих понимание сути и требований к выпускаемой продукции и соответствующих ей аудиторий. Обязанностью разработчика является обеспечение полноты, корректности и актуальности всех документов, предъявляемых разработчиком на момент их поставки. В ряде случаев требуется сохранить конфиденциальность и секретность предоставленных материалов. В договоре должны быть установлены уровни (грифы) конфиденциальности или секретности материалов, представляемых заказчиком документатору.

План документирования должен включать **определение аудитории пользователей документации**, уровня образования, квалификации, способностей, подготовки, опыта пользователей и другие характеристики, связанные с содержанием, структурой и использованием документации. Обычно имеется ряд различных групп пользователей, преследующих различные цели при пользовании конкретной документацией. Каждый тип пользователей, включая их характеристики и задачи, решаемые ими при помощи документации, должен определяться отдельно.

План документирования должен быть официально согласован и утвержден, что подтверждает полный учет и **тестирование в документах всех требований заказчика к программному продукту**. Должны быть определены процессы тестирования и проверки, выполняемые при разработке документации. План документирования должен быть подготовлен и утвержден до начала разработки документации, чтобы гарантировать согласование всеми сторонами содержания поставленных задач и используемых в проекте методов. После утверждения плана он должен быть доведен до всех заинтересованных сторон, включая разработчиков документации, а также заказчика и субподрядчиков [15].

Проверка результатов выполнения плана должна проводиться заказчиком с привлечением документаторов. Целью проверки является гарантирование полноты и достоверности, представленных материалов и удовлетворения ими возможности выполнения **требований заказчика к функциям и характеристикам программного продукта** в соответствии с условиями договора и плана документирования. Проверка должна проводиться персоналом заказчика, обладающим соответствующей квалификацией и полномочиями на изменения в доку-

ментах и их утверждению. Заказчик должен хранить копии всех вносимых изменений для сравнения их с последующими проектами документов. Заказчики должны уделять особое внимание структуре, полноте и практичности документации в соответствии с планом-проспектом ее содержания. План документирования должен быть проверен и утвержден до начала работ над первым проектом документации.

Документатор должен обеспечить разработку документации так, чтобы допустить возможность внесения в нее корректировок всех типов. Для этого необходимо, чтобы разработчики документации получали учтенные копии *изменений требований к программному продукту*, подтверждающие внесение соответствующих изменений в данную версию после конкретной даты ее пересмотра.

Испытания эксплуатационной документации на практичность следует рассматривать как дополнение к проводимым проверкам и анализам документации. При необходимости в плане документирования следует определить внешнюю среду тестирования, которая должна полностью соответствовать *эксплуатационной среде конечного пользователя и требованиям к программному продукту*. При проведении тестирования документации на практичность, необходимо проверить соответствие и актуальность документов конкретному программному продукту.

При сертификационных испытаниях может быть использовано для оценки практичности определения характеристик качества в стандартах **ISO 9126** и **ISO 25000**. *Практичность – применимость*: свойства программного продукта и документации, отражающие сложность их понимания, изучения и использования для квалифицированных пользователей при применении в указанных условиях. Требования к практичности и ее характеристикам – понятности и простоте использования, зависят от назначения и функций продукта и могут формализоваться заказчиками набором свойств, необходимых для обеспечения удобной и комфортной эксплуатации. Для обеспечения полноценного изучения процессов применения программного продукта специалистами необходима эксплуатационная документация, объем которой существенно зависит от назначения и функций, и может быть задан на основе анализа прецедентов подобных успешных проектов. Для некоторых проектов, подлежащих широкому тиражированию, могут быть желательны адекватные по содержанию электронные учебники, требования к объему и функциям которых целе-

сообразно оценивать также по прецедентам. Следует учитывать, что малый объем эксплуатационной документации может снизить качество и полноту использования функций сложного продукта, а очень большой объем – также может ухудшить эксплуатацию из-за трудности выделения из множества второстепенных деталей и освоения, наиболее существенных свойств и особенностей его применения.

В практичности следует учитывать все разнообразие характеристик внешней среды пользователей, на которые может влиять продукт, включая требующуюся подготовку к использованию и оценке результатов функционирования комплекса программ. **Применимость (практичность) использования программного продукта и документации** – понятие достаточно субъективное и трудно формализуемое, однако в итоге зачастую значительно определяющее функциональную пригодность и эффективность применения продукта. В эту группу показателей входят атрибуты с различных сторон отражающие функциональную понятность, удобство освоения или простоту использования. Некоторые характеристики можно оценивать экономическими показателями – затратами труда и времени специалистов на реализацию соответствующих функций.

Понятность: свойства, обеспечивающие пользователю понимание документации, является ли программный продукт пригодным для его целей, и как его можно использовать для конкретных задач и условий применения. Понятность зависит от качества документации и субъективных впечатлений от функций и характеристик программного продукта. Ее можно описать качественно четкостью функциональной концепции, широтой демонстрационных возможностей, полнотой, комплектностью и наглядностью представления в эксплуатационной документации возможных функций и особенностей их реализации.

Простота использования: возможность пользователю удобно и комфортно эксплуатировать и управлять программным продуктом. Аспекты изменяемости, адаптируемости и легкости инсталляции могут быть предпосылками для простоты использования и выбора конкретного продукта. Она соответствует управляемости, устойчивости к ошибкам и согласованности с ожиданиями и навыками пользователей. Эта характеристика учитывает физические и психологические особенности пользователей и отражает уровень комфортности условий эксплуатации, возможность предотвращения ошибок пользовате-

лей. Кроме того, удобство использования характеризуется рядом динамических параметров: временем ввода и отклика на задание, длительностью решения типовых задач, временем на регистрацию результатов, которые перекрываются с динамическими характеристиками временной эффективности. Простоту использования комплексов программ административных систем, в значительной степени, характеризует корректность и адекватность описаний интерактивных директив управления, объем и время ввода заданий, и время ожидания пользователями результатов при их исполнении. Однако некоторые атрибуты этой характеристики доступны для более полной количественной оценки путем измерения трудоемкости и длительности соответствующих процессов подготовки и обучения квалифицированных пользователей к полноценной и эффективной эксплуатации программного продукта.

Изучаемость: свойства программного продукта и документации, обеспечивающие удобное освоение его применения достаточно квалифицированными пользователями. Она может определяться трудоемкостью и длительностью подготовки пользователя к полноценной эксплуатации продукта. Качество изучаемости зависит от внутренних свойств и сложности комплекса программ и документов, а также от субъективных характеристик квалификации конкретных пользователей. На значения изучаемости существенно влияют демонстрационные возможности справочных средств обучения, качество и объем эксплуатационной документации.

Приобретение, поставка, разработка, функционирование и сопровождение программных продуктов, в большой степени, зависит от квалификации персонала. Поэтому в проекте программного продукта обязательно должно быть запланировано и осуществлено **обучение** с целью подготовки работы персонала при его приобретении, поставке, разработке или сопровождении. Для достижения высокого качества программной документации она должна рассматриваться как **неотъемлемая часть процесса создания и испытаний программного продукта**.

Затраты на создание достаточно полного комплекта эксплуатационной документации практически пропорциональны размеру комплекса программ. Удельные затраты на документацию зависят от класса, назначения и широты применения программного продукта, что трудно учесть в достаточно общем виде. Эти затраты сопутст-

вуют в некоторой степени всем этапам разработки, однако, оформление эксплуатационных документов обычно локализуется в специальном этапе работ. Затраты на разработку комплекта *эксплуатационной документации* для сложных программных продуктов, подлежащих длительному сопровождению, обычно определяются затратами на объем текста документов *ориентировочно* 5 – 10 страниц на тысячу строк программы.

Завершение сертификационных испытаний программных продуктов

Предъявляемый заявителем на сертификацию программный продукт должен представляться в комплекте с соответствующей документацией. Перечень и содержание группы этих документов ниже *ориентированы на общий случай удостоверения качества версий сложных программных продуктов реального времени*. Комплект документов может сокращаться и адаптироваться по согласованию между заявителем, сертифицикаторами и руководством производства в соответствии с характеристиками программных продуктов. Некоторые документы могут объединяться в интегрированные отчеты с четкой ответственностью определенных специалистов за их выполнение [15].

Ориентировочный комплект основных документов при сертификации программного продукта состоит *из трех групп*:

- *базовые нормативные документы* характеристик качества программных продуктов и их производства, а также подготовленные разработчиком на их основе Программа, Руководство и инструкции, предъявляемые испытателями сертифицирующей организации для испытаний версии программного продукта – рис. 11.2;

- *исходные документы*, отражающие результаты предварительных испытаний и характеристики сертифицируемого программного продукта, подготавливаемые руководством производства продукта (заявителем) для сертификации его качества – рис. 11.3;

- *результатирующие документы* сертификационных испытаний версии программного продукта, представляемые органу сертификации, заявителю и руководству производства для утверждения сертификата и знака качества – рис. 11.4.

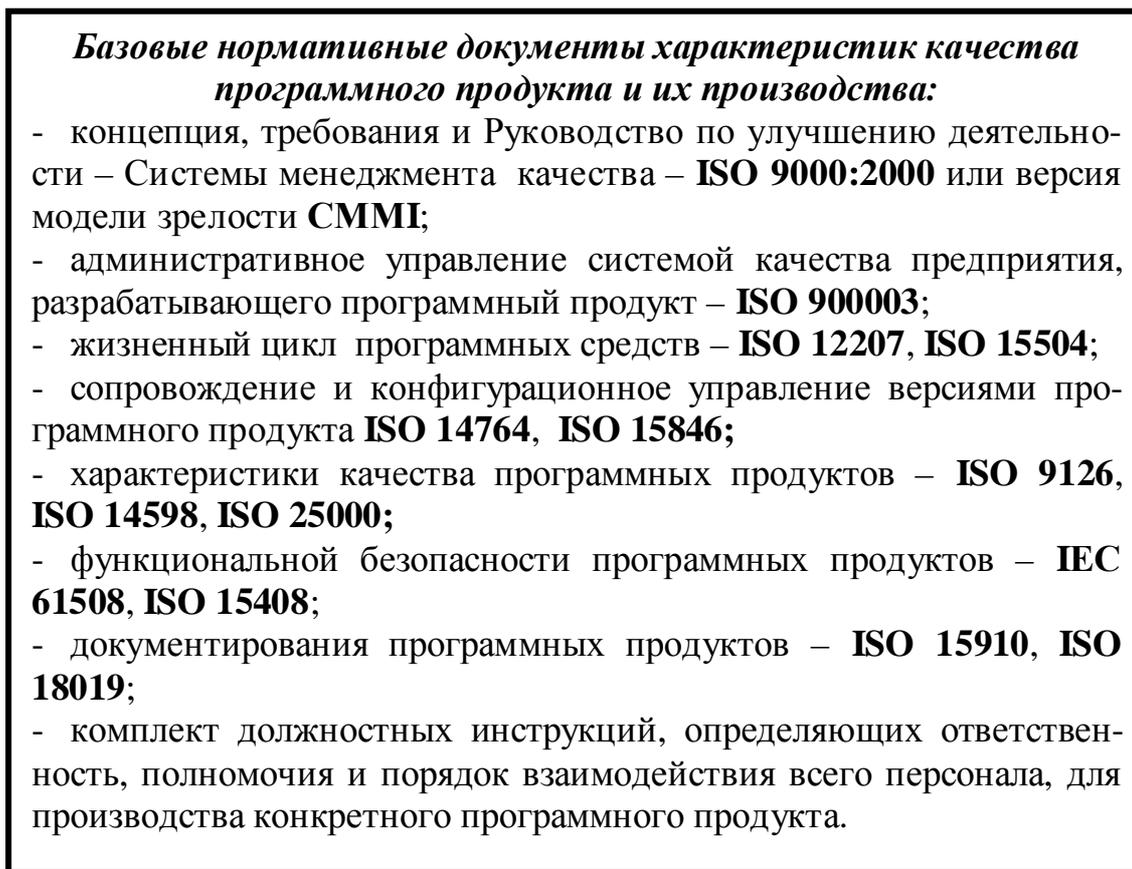


Рис. 11.2

Состав и содержание базовых нормативных документов для сертификации программных продуктов зависят от требований к их качеству, а также от характеристик проектирования, разработки и модификации программных средств и особенностей технологической среды. Поэтому необходимый комплект документов для каждого предприятия или проекта следует выбирать и адаптировать – **создавать профиль стандартов**, применительно к этим характеристикам.

Заявитель должен подготовить и предъявить испытательной лаборатории согласованный между заказчиком и разработчиком и утвержденный комплект исходных документов для проверки их достоверности, достаточности состава и качества изготовления в соответствии с нормативными документами (см. рис. 11.2).

При подготовке для сертификационных испытаний программного продукта в составе системы руководителями производства следует зафиксировать **результаты** [3, 11]:

- завершена разработка всех функций комплекса программ и исправление всех выявленных ошибок;
- все функциональные компоненты размещены под формаль-

ный автоматизированный контроль системы управления конфигурацией компонентов проекта;

Исходные документы, отражающие результаты предварительных испытаний и характеристики сертифицируемого программного продукта:

- Техническое задание – требования к функциям, характеристикам, качеству программного продукта, системы и внешней среды;
- Договор заказчика с производителем программного продукта;
- описание целей, требований и обязательств производителя в области критериев качества процессов производства и программного продукта;
- полные тексты программ, содержание базы данных и технологической документации версии программного продукта;
- комплект эксплуатационных документов, поставляемых заказчику и пользователям для применения версии программного продукта;
- план, Программа и методики испытаний, применения и оценки качества программного продукта;
- методики сопровождения, анализа и утверждения версий программного продукта;
- методика конфигурационного управления, утверждения, хранения, защиты, копирования версий программного продукта и сопровождающих документов в архиве предприятия;
- технические условия на версию программного продукта, базу данных управления конфигурацией и эксплуатационную документацию для тиражирования и серийного производства;
- руководство по генерации и инсталляции пользовательских версий и загрузке базы данных в соответствии с условиями и характеристиками внешней среды;
- акт о завершении предварительных испытаний и готовности к поставке и/или предъявлению для сертификационных испытаний версии программного продукта.

Рис. 11.3

- завершено компонентное и комплексное тестирование всех функций и исправление всех выявленных дефектов в версии программного продукта;
- подготовлена полная версия программного продукта с контролируемыми изменениями после предварительных испытаний;
- версия программного продукта сопровождается технологиче-

ской и эксплуатационной документацией, перечнем изменений, содержит список отчетов о дефектах, которые исправлены в версии;

Результирующие документы сертификационных испытаний программного продукта:

- Договор заявителя с сертифицирующей организацией на проведение испытаний версии программного продукта;
- отчет заявителя о наличии, актуальности и систематичности тестирования и оформления программного продукта и документации на протяжении жизненного цикла программного продукта;
- отчет сертифицированных Программы и методик проведения сертификационных испытаний в соответствии с требованиями Договора на сертификацию с заявителем;
- результаты аттестации имитаторов внешней среды и генераторов динамических тестов для сертификационных испытаний версии программного продукта;
- результаты выполнения планов и методик сертификационных испытаний, протоколы соответствия испытаний предъявляемым требованиям, утвержденные сертифицированными и согласованные с заявителями;
- протоколы достигнутых при сертификационных испытаниях характеристиках качества версии программного продукта;
- акт результатов испытаний реальных характеристик версии программного продукта, выводы о их соответствии требованиям к характеристикам программного продукта;
- сертификат версии программного продукта и обеспечения его жизненного цикла, лицензия на применение знаков соответствия.

Рис. 11.4

- предоставлена контролируемая полная версия программного продукта, которая установлена в тестовой внешней среде, стабильно функционирует и позволяет получать поддающиеся интерпретации результаты испытаний;
- проведены совещания, посвященные управлению незавершенными работами по устранению дефектов и оценки времени для исправления дефектов;
- в предшествующий, согласованный с заказчиком интервал времени, не произошло сбоев, остановки, неожиданного прекращения

процесса или другой аномалии функционирования комплекса программ на объектной ЭВМ в системе;

- анализ функционирования продукта показал, что комплекс программ достиг приемлемого уровня качества, стабильности, надежности и безопасности;
- оценки покрытия требований и допустимых рисков показали, что все риски сокращены до допустимых пределов;
- группа управления продуктом и системой установила, что программный продукт, как это определено в ходе заключительного цикла предварительных испытаний, удовлетворяет требованиям заказчика и пользователей;
- проведено совещание разработчиков с заказчиком и установлено, что **критерии завершения предварительных испытаний программного продукта выполнены, и его можно предъявить для сертификационных испытаний.**

Завершаются предварительные испытания предъявлением заказчику на утверждение **комплекта документов**, содержащих **результаты необходимые для сертификационных испытаний** версии программного продукта (см. рис. 11.3).

Кроме того, целесообразно подготавливать факультативно, по согласованию, для заказчика и пользователей следующие документы:

- документацию средств автоматизации проектирования, производства, модификации, контроля и предварительных испытаний программного продукта;
- тесты, сценарии и генераторы динамических тестовых данных, использованные для предварительных испытаний программных компонентов и версии продукта в целом;
- результаты и протоколы предварительных испытаний, функциональные и конструктивные характеристики программного продукта в реальной внешней среде;
- отчет о подтверждении заданного качества, полные характеристики достигнутого качества функционирования, а также степени покрытия тестами спецификации требований к программному продукту;
- план, методики и средства автоматизации обучения заказчика и пользователей применению испытанной версии программного продукта;

- технические условия на версию программного продукта, базу данных управления конфигурацией и эксплуатационную документацию для тиражирования и серийного производства;
- руководство по генерации и инсталляции пользовательских версий и загрузке базы данных в соответствии с условиями и характеристиками внешней среды;
- отчет о технико-экономических показателях производства завершенной версии программного продукта, выполнении планов и использованных ресурсах;
- акт о завершении предварительных испытаний и готовности к поставке и/или предъявлению для сертификационных испытаний версии программного продукта.

На завершающих этапах предварительных испытаний, когда, как правило, на группу тестирования оказывают давление руководители для ускорения и окончания работ, разработчикам полезно иметь в своем распоряжении *документ*, в котором оговариваются все требования и дополнения заказчика, которые планируется испытать. Специалисты склонны забывать, какие соглашения и требования были достигнуты, а также планы проведения предварительных испытаний. Если проблема возникнет в области, которая не подвергалась тестированию в соответствии с планом и Программой, она должна быть устранена и проверена после выпуска очередной версии программного продукта [3, 11].

Отчетный доклад о результатах предварительных испытаний должен содержать перечень *всех не устраненных дефектов*, с соглашением и планом того, будут ли они исправлены в более поздних версиях или же их исправление откладывается на неопределенное время. Необходимо вести наблюдение за такими дефектами, чтобы не тратить дополнительные усилия на их выявление в будущих версиях продукта. Принимая решение прекратить работы по предварительным испытаниям, руководителям производства следует учитывать несколько факторов, которые играют важную роль при *оценке неготовности программного продукта* к поставкам:

- количество катастрофических дефектов, обнаруженных в процессе тестирования, которые остаются неисправленными;
- общее количество возможных предсказуемых дефектов, которые не были исправлены;
- процентное отношение количества тестов, которые завершили-

лись успешным исходом и устранением дефектов, к числу всех запланированных тестов.

Критерий выхода из предварительных испытаний и готовности версии программного продукта к сертификации включают:

- время, отведенное на испытания, истекло, хотя неизвестно достигнутое качество программного продукта, однако любой другой метод прекращения испытаний может давать большую уверенность в качестве сертифицируемого или поставляемого программного продукта;
- завершены все запланированные циклы тестирования, и план проведения испытаний выполнен, тестовое покрытие лучше, чем в случае, когда просто не хватило времени для доведения тестирования до логического конца;
- профиль устраненных дефектов соответствует критерию выхода из испытаний и возможное количество не устраненных ошибок на тысячу строк программного кода достигнуто.

С целью выработки **оценки готовности версии** программного продукта к сертификации следует проводить специальные **совещания разработчиков и заказчика**. В некоторых предприятиях оценку готовности определяет группа тестирования; в других организациях совещание проводит руководитель проекта системы, либо руководитель производства и заказчик программного продукта. В любом случае в задачу группы тестирования и испытаний входит представление результатов и выработка рекомендаций относительно готовности продукта к поставкам для сертификации.

С самого начала проекта необходимо определить, что следует считать успешным прохождением стадии предварительных испытаний, и **соответствия исходным требованиям** заказчика, когда можно прекратить испытания продукта и начать его поставку или сертификацию. Результатом сертификационных испытаний должен быть комплект отчетных документов (см. рис. 11.4) и подтверждение результатов, зафиксированных при предварительных испытаниях – рис. 11.2 и 11.3. При сертификационных испытаниях программного продукта целесообразно выборочно или полностью использовать результаты предварительных испытаний с учетом их полноты и достоверности. Утверждение этого комплекта документов для конкретного программного продукта дает право на присвоение ему **сертификата и знака качества**.

Поставка пользователям сертифицированной версии программ продукта для применения

После завершения испытаний новой версий программного продукта, обычно осуществляется процесс ее внедрение для применения (см. рис. 11.1). Это производится, как правило, *в два этапа*; силами разработчиков версии в целях обкатки, проверки и выявления ошибок в изменениях на этапе опытной эксплуатации, и/или посредством использования специализированных коллективов сопровождения, для тиражирования и распространения версий программного продукта. Основные обязанности специалистов сводятся к передаче физических носителей с текстами программ и комплектом эксплуатационной документации, а также к проведению консультаций для выделенной группы специалистов заказчика и пользователей. Разработчики и испытатели в этом случае получают возможность непосредственно *контролировать работу пользователей* с системой и документацией, что обеспечивает высокую оперативность отработки замечаний и рекламаций, формирование квалифицированных предложений для изменений, оценку эффективности применения версии программного продукта. Кроме того, должны разрабатываться учебно-методический план, подготавливаться учебные пособия, необходимые для *обучения пользователей*, а также проводиться обучение выделенной группы специалистов, ответственных за последующее обучение коллективов пользователей.

В процессе жизненного цикла большое значение имеет *история эксплуатации программного продукта*, развития его версий и соответствующая документация. Еще на стадии проектирования первой версии могут возникать *идеи совершенствования комплекса программ*, которые в то время невозможно реализовать из-за высокой стоимости, ограниченных сроков проектирования или по иным причинам. Идеи изменения могут быть направлены на коренное улучшение функциональных возможностей программ или некоторые «косметические» улучшения реализуемых функций. Идеи небольших корректировок программ целесообразно накапливать отдельно от предложений по существенному совершенствованию программного продукта. Таким образом, должен создаваться документ – *исходные данные для изменения требований* и планирования доработок в процессе сопровождения, содержащий разделы:

- выявленные дефекты и ошибки в программном продукте;
- предложения по совершенствованию функций и улучшению качества эксплуатируемых версий программного продукта;
- идеи и предполагаемая экономическая эффективность коренной модернизации, расширения функций и улучшения характеристик версии программного продукта.

После внесения изменений в требования специалисты разрабатывают и испытывают конкретные корректировки программного продукта. Специалисты должны провести проверки каждого внесенного изменения совместно с заказчиком, утвердившим изменение в целях подтверждения целостности и работоспособности измененного программного продукта. Получить подтверждения того, что внесенные **изменения удовлетворяют требованиям заказчика**, установленным в договоре, посредством вспомогательного процесса испытаний качества, проведения аудита функциональной и физической конфигурации. Результатами данной работы являются: **новая версия** программного продукта, включающая в себя принятые изменения требований; отклоненные изменения; отчет о приемке версии; отчеты о проверках и аудитах; отчет об испытаниях откорректированного программного продукта.

Снятие сертифицированной версии программного продукта с эксплуатации и развития версий должны быть подготовлены анализом, обосновывающим это решение. В анализе следует определить и экономически обосновать: возможность сохранения устаревшей версии комплекса программ, а также необходимость создания и применения новой версии программного продукта. При снятии программного продукта с сопровождения следует определить необходимые для этого действия, а затем разработать и документально оформить этапы работ, обеспечивающие их эффективное выполнение. Должны быть предусмотрены возможности **доступа к полным архивным документам** снятого с сопровождения программного продукта.

Анализ результатов сертификации и усовершенствование процессов испытаний программного продукта

По окончании разработки версии программного продукта специалистам полезно **изучить ход выполнения Программы сертифи-**

кационных испытаний, чтобы определить, каким образом можно ее усовершенствовать, и в следующем проекте получить преимущества. Испытатели должны постоянно изучать данные, получаемые специалистами контроля качества, и следовать их рекомендациям по исправлению дефектов. Кроме того, на протяжении всего жизненного цикла комплекса программ целесообразно **документировать полученные уроки** и оценивать их на каждом промежуточном рубеже испытаний. Измерения, сделанные в процессе тестирования и, особенно, на этапе исполнения тестов, помогут акцентировать внимание на определенных видах требований и конкретных проблемах, требующих решения. С целью улучшения результатов, испытатели должны периодически диагностировать качество проделанной работы, изменяя при необходимости привычные способы их ведения. Зафиксированные на этом этапе уроки могут оказаться полезными при производстве и проведении испытаний в дальнейшем.

Полученные уроки, оценки измерений характеристик программного продукта, любые связанные с этим действия по устранению недостатков или по совершенствованию работ, должны документироваться на протяжении всего процесса испытаний в централизованной базе данных **конфигурационного управления требованиями и тестами**. Наличие отвечающей современным требованиям базы данных, содержащей спецификации требований, применявшиеся тесты, важные проблемы и полученные уроки, позволяет специалистам, занятым в данном проекте, следить за ходом работ и состоянием проблем вплоть до его полного завершения. Для каждого урока желательно указывать измерение, **демонстрирующее потенциальный выигрыш** (экономленные часы труда), если бы было применено определенное действие по совершенствованию процесса. После завершения тестирования специалисты должны сопоставить фактическую реализацию Программы сертификационных испытаний с запланированными критериями.

Усовершенствование процессов сертификационных испытаний целесообразно выполнять итеративно. Испытателям следует выяснять, повторялись ли те или иные дефекты и ошибки, и подтверждать, были ли упущены из виду какие-либо предложения по совершенствованию испытаний. При изучении реализации Программы испытаний следует концентрировать внимание на оценке того, удовлетворяет ли комплекс программ критериям завершения тестирова-

ния, и готов ли он к поставкам. Испытатели должны освоить непрерывный итерационный процесс **изучения полученных уроков как часть развития культуры тестирования**. После сбора полученных уроков и других измерений, и определения корректирующих действий испытатели должны оценивать эффективность Программы испытаний.

Отказ комплекса программ при выполнении тестовой процедуры не всегда вызван его дефектами. Проблема может представлять собой **ложный отрицательный результат**, означающий, что тест завершился неудачей, хотя в тестируемом программном продукте не было ошибки. Случаи ложных отрицательных результатов **могут возникнуть из-за** изменений в программах, ошибок в настройке теста, ошибок тестовых процедур, ошибок испытателя, логических ошибок автоматизированного динамического теста или ошибок настройки внешней среды. Если ряд ложных отрицательных результатов указывает на дефект, допущенный при разработке тестовых процедур, неизбежно усовершенствование процесса их создания.

Даже в случае совпадения полученных результатов с ожидаемыми эталонами, испытателям необходимо убедиться **в отсутствии эффекта ложных положительных результатов**, когда кажется, что тестовая процедура выполнена успешно, но на самом деле существует проблема, связанная с комплексом программ. Испытатели должны внимательно следить за появлением ложных положительных результатов, к которым может приводить использование инструментов автоматизированного тестирования, недостаточно чувствительных к нюансам конкретного комплекса программ. Наряду с критическим просмотром или экспертной оценкой тестовых процедур, группа тестирования должна оценивать и осуществлять выборочную проверку результатов тестирования даже в том случае, если тест завершился успешно.

Группа тестирования должна выявлять те компоненты и функции, по которым было **составлено самое большое число отчетов о проблемах**. Такой анализ может указать на необходимость проведения дополнительного тестирования для проверки этих компонентов. Если разработчики заявляют, что дефекты в определенной функциональной области ликвидированы, а регрессионное тестирование обнаруживает там дефекты, испытатель должен выяснить, проблемы заключаются во внешней среде или неудачно реализованы исправле-

ния в программном продукте. Кроме того, анализ поможет определить функциональность, с которой связано наибольшее количество обнаруженных дефектов, на чем необходимо сосредоточиться при дальнейшем тестировании и исправлении ошибок.

При необходимости испытатели должны **помочь разработчикам понять и воспроизвести дефекты системы и/или программного продукта**. Каждый дефект обычно ранжируется в зависимости от приоритета. После того как разработчики устранят выявленные дефекты, испытатели должны провести регрессионное тестирование продукта с целью проверки, можно ли считать тот или иной отчет о проблеме закрытым. После этого отправляется уведомление заинтересованным членам группы управления конфигурацией и группы разработчиков программного продукта, в котором сообщается о появлении нового отчета о проблеме.

Приложение

**МЕЖДУНАРОДНЫЕ И ГОСУДАРСТВЕННЫЕ
СТАНДАРТЫ, РЕГЛАМЕНТИРУЮЩИЕ
ТРЕБОВАНИЯ, ЖИЗНЕННЫЙ ЦИКЛ, ИСПЫТАНИЯ И
СЕРТИФИКАЦИЮ КОМПЛЕКСОВ ПРОГРАММ**

1. **CMMI – Capability Maturity Model Integration for Product and Process Development** – Интегрированная модель оценивания зрелости продуктов и процессов разработки программных средств.
2. **ISO 15288:2002**. Системная инженерия. Процессы жизненного цикла систем.
3. **ISO 19760:2003**. – Системная инженерия. Руководство по применению стандарта ISO 15288.
4. **ISO 12207:2008**. ИТ. Процессы жизненного цикла программных средств.
5. **ISO 15271:1998**. (ГОСТ Р – 2002). ИТ. Руководство по применению ISO 12207.
6. **ISO 16326:1999**. (ГОСТ Р – 2002). ИТ. Руководство по применению ISO 12207 при административном управлении проектами.
7. **ISO 15504:1-5:2003-2006**. ИТ. Процесс аттестации. Ч.1. Концепция и словарь. Ч. 2. Выполнение аттестации. Ч. 3. Руководство по производству аттестации. Ч. 4. Руководство пользователей для процессов усовершенствования и определения зрелости процессов. Ч. 5. Образец модели процессов аттестации.
8. **ГОСТ Р 51904 – 2002**. Программное обеспечение встроенных систем. Общие требования к разработке и документированию.
9. **ISO 9000:2000**. (ГОСТ Р – 2001). Система менеджмента (административного управления) качества. Основы и словарь.
10. **ISO 9001:2000**. (ГОСТ Р – 2001). Система менеджмента (административного управления) качества. Требования.
11. **ISO 9004:2000**. (ГОСТ Р – 2001). Система менеджмента (административного управления) качества. Руководство по улучшению деятельности.
12. **ISO 90003:2004** – Руководство по применению стандарта **ISO 9001:2000** к программным средствам.

13. **ГОСТ Р 40.003:2005** – Порядок сертификации систем менеджмента качества на соответствие **ГОСТ Р ИСО 9001 – 2001**.
14. **Система сертификации ГОСТ Р. Временный порядок сертификации производств с учетом требований ГОСТ Р ИСО 9001 – 2001.**
15. **Руководящие указания по аудиту систем менеджмента качества ГОСТ Р ИСО 19011:2003.**
16. **ISO 10005: 2005.** (ГОСТ Р – 2007). Менеджмент организации (Административное управление качеством). Руководящие указания по планированию качества.
17. **ISO 10006: 1997.** (ГОСТ). Руководство по качеству при управлении проектом.
18. **ISO 10007: 2007.** (ГОСТ Р – 2007). Административное управление качеством. Руководящие указания при управлении конфигурацией.
19. **ISO 10013: 2001.** (ГОСТ Р – 2007). Менеджмент организации (Административное управление качеством). Руководство по документированию систем менеджмента качества.
20. **ISO 12182:1998.** (ГОСТ Р– 2002). ИТ. Классификация программных средств.
21. **ISO 9126:1991.** (ГОСТ – 1993). ИТ. Оценка программного продукта. Характеристики качества и руководство по их применению.
22. **ISO 9126-1-4: 2002.** ИТ. ТО. Качество программных средств: Ч.1. Модель качества. Ч.2. Внешние метрики. Ч. 3. Внутренние метрики. Ч. 4. Метрики качества в использовании.
23. **ISO 14598-1-6:1998-2000.** Оценивание программного продукта. Ч.1. Общий обзор. Ч.2. Планирование и управление. Ч. 3. Процессы для разработчиков. Ч.4. Процессы для покупателей. Ч.5. Процессы для оценщиков. Ч.6. Документирование и оценивание модулей.
24. **ISO 25000:2005** ТО. – Руководство для применения новой серии стандартов по качеству программных средств на базе обобщения стандартов ISO 9126:1-4: 2002 и ISO 14598:1-6:1998-2000.
25. **ISO 15939: 2002** – Процесс измерения программных средств.
26. **IEC 61508:1-6: 1998-2000.** Функциональная безопасность электрических / электронных и программируемых электронных систем. Часть 3. Требования к программному обеспечению. Часть 6.

Руководство по применению стандартов **IEC 61508-2** и **IEC 61508-3**.

27. **ISO 15408 -1-3. 1999.** (ГОСТ Р – 2002). Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Ч.1. Введение и общая модель. Ч. 2. Защита функциональных требований. Ч. 3. Защита требований к качеству.
28. **ISO 13335 - 1-5. 1996-1998.** ИТ. ТО. Руководство по управлению безопасностью. Ч.1. Концепция и модели обеспечения безопасности информационных технологий. Ч.2. Планирование и управление безопасностью информационных технологий. Ч.3. Техника управления безопасностью ИТ. Ч.4. Селекция (выбор) средств обеспечения безопасности. Ч.5. Безопасность внешних связей.
29. **ISO 10181: 1-7. ВОО. 1996-1998.** Структура работ по безопасности в открытых системах. Ч.1. Обзор. Ч.2. Структура работ по аутентификации. Ч.3. Структура работ по управлению доступом. Ч.4. Структура работ по безотказности. Ч.5. Структура работ по конфиденциальности. Ч.6. Структура работ по обеспечению целостности. Ч.7. Структура работ по проведению аудита на безопасность.
30. **ISO 14252:1996** (IEEE 1003.0). Руководство по функциональной среде открытых систем POSIX.
31. **ISO 9945:1-4:2003** – ИТ. Интерфейсы переносимых операционных систем. Ч.1. Базовые определения. Ч.2. Системные интерфейсы. Ч.3. Команды управления и сервисные программы. Ч. 4. Обоснование.
32. **ISO 13210:1994.** ИТ. Методы тестирования для измерения соответствия стандартам POSIX.
33. **ISO 14756: 1999.** ИТ. Измерение и оценивание производительности программных средств компьютерных вычислительных систем.
34. **ISO 12119:1994.** (ГОСТ Р – 2000 г). ИТ. Требования к качеству и тестирование.
35. **ISO 14764: 1999.** (ГОСТ Р – 2002). ИТ. Сопровождение программных средств.
36. **ISO 15846:1998.** ТО. Процессы жизненного цикла программных средств. Конфигурационное управление программными средствами.

37. **ISO 16085: 2004** – Характеристики процессов управление рисками при разработке, применении и сопровождении программных средств.
38. **ISO 6592:2000** ОИ. – Руководство по документации для вычислительных систем.
39. **ISO 9294:1990** (ГОСТ–1993). ТО. ИТ. Руководство по управлению документированием программного обеспечения.
40. **ISO 9127:1990** (ГОСТ–1993). ТО. ИТ. Руководство по управлению документированием программного обеспечения.
41. **ISO 15910:1999** (ГОСТ Р – 2002) ИТ. Пользовательская документация программных средств.
42. **ISO 18019:2004** ИТ. Руководство по разработке пользовательской документации на прикладные программные средства для офисов, бизнеса и профессиональных применений.
43. **ГОСТ Р 51901-2002**. Управление надежностью. Анализ риска технологических систем.
44. **DO-178 В -1995**. Соглашение по сертификации бортовых систем и оборудования в части программного обеспечения.

Основная литература

1. Липаев В.В. Программная инженерия. Методологические основы. Учебник. – М.: ТЕИС. 2006.
2. Крайер Э. Успешная сертификация на соответствие нормам ИСО серии 9000. Пер. с нем. – М.: ИЗДАТ. 1999.
3. Липаев В.В. Методы обеспечения качества крупномасштабных программных средств. – М.: СИНТЕГ. 2003.
4. Фатрелл Р. Т., Шафер Д. Ф., Шафер Л. И. Управление программными проектами: достижение оптимального качества при минимальных затратах. Пер. с англ. – М.: Вильямс. 2003.
5. Липаев В.В. Тестирование крупных комплексов программ на соответствие требованиям. Учебник. – М.: Глобус. 2008.
6. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. Пер. с англ. – М. ЛОРИ. 2003.
7. Липаев В.В. Человеческие факторы в программной инженерии: рекомендации и требования к профессиональной квалификации специалистов. Учебник. – М.: СИНТЕГ. 2009.

Дополнительная литература

8. Блэк Р. Ключевые процессы тестирования. Пер. с англ. – М: ЛОРИ. 2006.
9. Вигерс К.И. Разработка требований к программному обеспечению. Пер. с англ. – М.: Русская редакция. 2004.
10. Гецци К., Джазайери М., Мандриоли Д. Основы инженерии программного обеспечения. Пер. с англ. – СПб.: БХВ-Петербург. 2005.
11. Костогрызов А.И., Липаев В.В. Сертификация качества функционирования автоматизированных информационных систем. М.: Изд. Вооружение. Политика. Конверсия.1996.
12. Коберн А. Современные методы описания требований к системам. Пер. с англ. – М.: Лори.2002.
13. Липаев В.В. Функциональная безопасность программных средств. – М.: СИНТЕГ. 2004.
14. Липаев В.В. Анализ и сокращение рисков проектов сложных

программных средств. – М.: СИНТЕГ. 2004.

15. Оценка и аттестация зрелости процессов создания и сопровождения программных средств и информационных систем (ISO/IEC TR 15504 – СММ). – М.: Книга и бизнес. 2001.

16. Соммервилл И. Инженерия программного обеспечения. 6-е издание. Пер. с англ. – М.: Вильямс. 2002.

17. Трубачев А.П., Долинин М.Ю., Кобзарь М.Т. и др. Оценка безопасности информационных технологий. Общие критерии. Пер. с англ. Под ред. В.А. Галатенко. – М.: СИП РИА, 2001.

18. Тэллес М., Хсих Ю. Наука отладки. Пер. с англ. – М.: Кудиц-образ. 2003.