

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Санкт-Петербургский
государственный университет аэрокосмического приборостроения

Д. В. Богданов, В. В. Фильчаков

СТАНДАРТИЗАЦИЯ ЖИЗНЕННОГО ЦИКЛА И КАЧЕСТВА ПРОГРАММНЫХ СРЕДСТВ

Учебное пособие

Санкт-Петербург
2000

УДК 681.2
ББК 65.9(2)
Б73

Богданов Д. В., Фильчаков В. В.

Б73 Стандартизация жизненного цикла и качества программных средств: Учеб. пособие / СПбГУАП. СПб., 2000. 210 с.

Учебное пособие посвящено общим вопросам стандартизации в области обеспечения качества разработки программных средств.

В нем описываются стандарты в области обеспечения качества ПС, рассматриваются жизненные циклы программного средства, установленные в зарубежных и отечественных стандартах, обсуждается роль документации при создании качественного ПС. Особое внимание уделено вопросам оценки процесса разработки ПС.

Рассчитано на студентов старших курсов и аспирантов, специализирующихся в области математического и программного обеспечения ЭВМ.

Рецензенты:

кафедра вычислительной техники и математических методов исследования операций Санкт-Петербургского государственного университета
водных коммуникаций;
доктор технических наук профессор *Ф. А. Таубин*

Утверждено

редакционно-издательским советом университета
в качестве учебного пособия

© Санкт-Петербургский
государственный университет
аэрокосмического
приборостроения, 2000

© Д. В. Богданов,
В. В. Фильчаков, 2000

ПРЕДИСЛОВИЕ

Увеличивающаяся в мировом масштабе конкуренция среди организаций – разработчиков программного обеспечения (ПО), повышение требований конечного пользователя к качеству и надежности программных средств привело их разработчиков к пониманию важности вопросов стандартизации в области качества.

Для того чтобы поддерживать конкурентоспособность своей организации, разработчики ПО должны применять более эффективные, рентабельные методы, технологии, инструментальные средства, способствующие постоянному повышению качества и более совершенному удовлетворению потребителей ПО.

Требования потребителей часто включаются в технические условия (ТУ) или неформализованные требования, описанные на некотором вербальном языке. Однако технические условия и неформализованные требования сами по себе не гарантируют их удовлетворения в конечном продукте, так как в настоящее время существует проблема выработки приемлемых требований к программному продукту, а также ряд других проблем, возникающих в процессе разработки конечного продукта. Это соображение привело к разработке стандартов, руководств, руководящих документов, относящихся к системам качества и дополняющих релевантные требования к ПО, установленные в технических требованиях. Международные стандарты серии ИСО 9000 впервые создали общую основу для стандартов на системы качества, которые применимы в различных областях деятельности человека. Основные положения серии стандартов ИСО 9000 рассмотрены в гл. 1 учебного пособия.

Международные стандарты серии ИСО 9000 устанавливают, какие именно элементы должны включаться в систему качества, каким образом конкретная организация должна реализовать эти элементы. Введение единообразных систем качества не является целью этих стандартов. Потребности различных организаций отличаются друг от друга. На проект и реализацию системы качества обязательно оказывают влияние конкретные цели, продукция и процессы, а также специфические методы данной организации.

Международные стандарты серии ИСО 9000 основаны на понимании того факта, что всякая работа выполняется с помощью сети процессов. Каждый процесс имеет входные факторы, а выходом являются результаты процесса — продукция, осязаемая и не осязаемая. Каждая организация существует для того, чтобы выполнять работу, добавляющую стоимость. В процессе получения конечного продукта должны быть выполнены многочисленные операции, включающие в себя организацию, проектирование, управление технологическими процессами, маркетинг, обучение, управление людскими ресурсами, стратегическое планирование, поставку, техническое обслуживание и т. д. Принимая во внимание сложную структуру большинства организаций, важно выделить основные процессы, а также упростить и ранжировать процессы в зависимости от целей административного управления качеством.

Любая организация должна определить и установить свою сеть процессов и интерфейсов, и управлять ею. Организация создает, совершенствует и обеспечивает постоянный уровень качества своей продукции с помощью сети процессов. Это концептуальная основа стандартов серии ИСО 9000. В гл. 2 учебного пособия рассмотрены процессы жизненного цикла ПО и стандарты, их определяющие. Основное внимание в данной главе уделено отечественным стандартам 19- и 34-й систем, проекту международного стандарта ИСО/МЭК 12207, а также документу DO-178B, устанавливающему аспекты сертификации программ для авиационных систем.

В стандарте ИСО 2382-1 дано следующее определение программного обеспечения. ПО — это интеллектуальный продукт, состоящий из программ, процедур, правил и любой другой, связанной с ними документации, относящихся к функционированию системы обработки данных. Таким образом, документация является неотъемлемой частью ПО и ей, а также процессу ее формирования, должно уделяться пристальное внимание. Гл. 3 посвящена вопросам документирования процессов жизненного цикла ПО. В ней рассмотрена роль документации в обеспечении качества ПО, приведены требования стандартов к документации, разрабатываемой в процессе создания ПО, основные типы и виды программной документации.

В гл. 4 учебного пособия рассматривается проект стандарта ИСО/МЭК 15504, являющийся дополнением к другим международным стандартам и моделям для оценки возможности и эффективности организаций и процессов.

ИСО/МЭК 15504 включает намерения серии ИСО 9000 создать уверенность в управлении, обеспечивая пользователей структурой

для независимой оценки, возможности потенциальных поставщиков удовлетворять их потребности. Оценка процесса обеспечивает пользователей способностью оценить возможности процесса на непрерывной шкале простым и сравнимым способом, а не использовать характеристики качества, базирующиеся на ИСО 9001. Кроме того, структура, описанная в проекте стандарта ИСО/МЭК 15504, предоставляет возможность регулировать область оценки для покрытия специфических интересующих процессов, а не всех процессов, используемых организацией.

Стандартизация — наиболее перспективное направление развития передовых информационных технологий в проектировании, производстве и менеджменте, и любые усилия в этом направлении должны всячески приветствоваться.

Кроме того, стандартизация процесса разработки и эксплуатации ПО способствует контролю, оценке и регламентации труда всех участников данного процесса, побуждает к дисциплине.

Так как стандартизация способствует лучшему контролю и регламентации труда занятых в процессе разработки специалистов, побуждает их, прежде всего, к дисциплине, а не к свободному самовыражению в изобретении остроумных трюков и уловок, то введение стандартов наталкивается на определенный саботаж с их стороны. Необходимо отметить, что, во-первых, взятые на вооружение и используемые стандарты намного полезнее, чем хорошие стандарты, записанные на бумаге, а во-вторых, хорошие стандарты получаются не сразу. В процессе систематического применения и совершенствования плохие стандарты можно довести до хороших.

В тексте учебного пособия часто встречаются следующие понятия: программная система, программный продукт, программное средство, программное обеспечение. Авторы признают существование различий между данными понятиями, но в учебном пособии они сознательно рассматриваются как синонимы. Для уяснения различий между данными понятиями читателю следует обратиться к Прил. 1 данного пособия.

Авторы учебного пособия выражают огромную признательность и благодарность доктору технических наук А.Г. Ломако за предоставленный материал по моделям и метрикам оценки качества программного обеспечения, приведенный в подразд. 1.4., а также Risto Nevalainen, Software Technology Transfer, за материал по проекту SPICE.

ГЛАВА 1

СТАНДАРТЫ В ОБЛАСТИ ОБЕСПЕЧЕНИЯ КАЧЕСТВА ПРОГРАММНЫХ СИСТЕМ

1.1. Основные положения стандартов серии ИСО 9000

Во-первых, под стандартами серии ИСО 9000 понимаются все международные стандарты, разработанные Техническим комитетом 176 “Административное управление качеством и обеспечении качества” Международной организации по стандартизации (ИСО). В настоящее время серия содержит все международные стандарты с номерами от 9000 до 9004 (включая все части ИСО 9000 и ИСО 9004), от 10001 до 10020 (включая все части), а также ИСО 8402. Ниже приведены названия основных стандартов, составляющих данную серию.

ИСО 9000-1-94. “Стандарты в области административного управления качеством и обеспечения качества. Часть 1. Руководящие положения по выбору к применению”.

ИСО 9000-2-93. “Стандарты в области административного управления качеством и обеспечения качества. Часть 2. Общие руководящие положения по применению ИСО 9001, ИСО 9002 и ИСО 9003”.

ИСО 9000-3-91. “Стандарты в области административного управления качеством и обеспечения качества. Часть 3. Руководящие положения по применению ИСО 9001 при разработке, поставке и техническом обслуживании ПО”.

ИСО 9000-4-93. “Стандарты в области административного управления качеством и обеспечения качества. Часть 4. Руководящие положения по административному управлению программой общей надежности”.

ИСО 9001-94. “Системы качества. Модель для обеспечения качества при проектировании, разработке, производстве, монтаже и обслуживании”.

ИСО 9002-94. “Системы качества. Модель для обеспечения качества при производстве, монтаже и обслуживании”.

ИСО 9003-94. “Системы качества. Модель для обеспечения качества при контроле готовой продукции и заключительных испытаниях”.

ИСО 9004-1-94. “Административное управление качеством и элементы системы качества. Часть 1. Руководящие положения”.

ИСО 9004-2-91. “Административное управление качеством и элементы системы качества. Часть 2. Руководящие положения по услугам”.

ИСО 9004-3-93. “Административное управление качеством и элементы системы качества. Часть 3. Руководящие положения по обработанным материалам”.

ИСО 9004-4-93. “Административное управление качеством и элементы системы качества. Часть 4. Руководящие положения по повышению качества”.

ИСО 10011-1-90. “Системы качества. Руководящие положения по проверкам. Часть 1. Проверки”.

ИСО 10011-2-91. “Системы качества. Руководящие положения по проверкам. Часть 2. Критерии квалификации экспертов-аудиторов систем качества”.

ИСО 10011-3-91. “Системы качества. Руководящие положения по проверкам. Часть 3. Административное управление программами проверок”.

ИСО 10012-1-92. “Обеспечение качества измерительного оборудования. Требования. Часть 1. Системы метрологического обеспечения измерительного оборудования”.

ИСО 10013. “Руководства по качеству. Положения по разработке. (На стадии издания)”.

ИСО 8402-94. “Управление качеством и обеспечение качества. Словарь”.

Увеличившаяся в настоящее время конкуренция между организациями, производителями продукции, в том числе и программных систем, приводит к установлению более жестких требований к качеству этой продукции. Для того чтобы быть конкурентоспособными, организации должны применять эффективные системы, ведущие к повышению качества продукции и более совершенному удовлетворению требований своих заказчиков. Правильно сформулированные и полные требования заказчика, включенные в технические условия, еще не гарантирует того, что эти требования будут полностью удовлетворены, так как в системе поставок и обеспечения организации могут быть недостатки. Это соображение обусловило разработку стандартов, относящихся к системам качества, и дополняющих требова-

ния заказчика к продукции. Международные стандарты серии ИСО 9000 предназначены для создания общей основы стандартов на системы качества. Под *системой качества* понимается, согласно ИСО 8402, совокупность организационной структуры, методик, процессов и ресурсов, необходимых для осуществления общего руководства качеством продукции, производимой организацией.

Система административного управления качеством организации – те аспекты общей функции управления, используемой организацией, которые определяют политику в области качества выпускаемой продукции, цели организации и ее ответственность, а также осуществляют их с помощью средств планирования, управления, обеспечения и улучшения качества в рамках системы качества. Кроме цели организации, на систему административного управления качеством влияют выпускаемая данной организацией продукция и характерные для нее методы производства. В силу того, что методы производства организаций, работающих даже в одной сфере, различны, да и цели, стоящие перед ними, не всегда едины, системы качеств этих организаций не совпадают. Основной задачей системы административного управления качеством является усовершенствование систем и процессов для повышения качества продукции.

Стандарты серии ИСО 9000 устанавливают, какие именно элементы должны быть включены в систему качества, тогда как организация сама должна реализовать их с учетом конкретных целей, продукции и процессов, а также специфических методов, используемых данной организацией.

Кроме того, руководящие положения и требования стандартов серии ИСО 9000 выражены в терминах целей системы качества, которые должны быть достигнуты, и не предписывают способы достижения этих целей, оставляя право выбора этих способов руководству организации. Стандарты данной серии отличают требования к системам качества от требований заказчика к продукции. Требования к системам качества являются дополнительными по отношению к техническим требованиям к продукции. Например, ИСО 12207 устанавливает жизненный цикл разработки программного обеспечения. Процессы и модели качества, соответствующие процессу обеспечения качества (2.3 по ИСО 12207), устанавливаются стандартами серии ИСО 9000.

Стандарт ИСО 9000-1 устанавливает четыре общие категории продукции, охватывающие все виды продукции, поставляемые любой организацией:

- технические средства;

- программное обеспечение;
- обработанные материалы;
- услуги.

Требования к системам качества, установленные в международных стандартах серии ИСО 9000, применимы ко всем четырем общим категориям продукции, но терминология, некоторые положения и аспекты систем административного управления качеством могут быть различными. Это видно из названий стандартов ИСО 9004-2 и ИСО 9004-3. Необходимо отметить, что любая организация предлагает продукцию, как минимум, двух категорий. Например, организация, занимающаяся разработкой программного обеспечения, дополнительно предоставляет своим заказчикам услуги по сопровождению созданного ПО.

Целью руководящих положений и требований международных стандартов серии ИСО 9000 является удовлетворение требований заказчика с позиции четырех аспектов, являющихся ключевыми для качества продукции.

1. Качество благодаря определению потребностей заказчиков в продукции. Первый аспект — это качество благодаря определению и модернизации продукции с целью ее соответствия требованиям и возможностям рынка.

2. Качество благодаря конструкции. Второй аспект — это качество благодаря встраиванию в продукцию характеристик, способствующих тому, чтобы она отвечала требованиям и возможностям рынка. Другими словами, качество благодаря конструкции — это те свойства конструкции, которые влияют на бесперебойность работы изделия в переменных условиях производства и применения.

3. Качество благодаря соответствию конструкции. Третьим аспектом является качество благодаря поддержанию постоянного соответствия конструкции, реализации характеристик, заложенных в проект.

4. Качество благодаря техническому обслуживанию. Четвертый аспект — это качество благодаря техническому обслуживанию продукции в процессе ее эксплуатации по мере необходимости, чтобы сохранить желаемые характеристики.

Серия стандартов ИСО 9000 устанавливает общие руководящие положения, касающиеся административного управления, и требования к внешнему обеспечению качества относительно четырех аспектов.

Международные стандарты серии ИСО 9000 основаны на понимании того факта, что всякая работа выполняется с помощью процес-

сов (рис. 1.1). Каждый процесс имеет входные факторы. Выходом процесса является результат – продукция, осязаемая и не осязаемая. Сам процесс является (или должен являться) преобразованием, добавляющим стоимость. В каждом процессе принимают участие в той или иной мере люди и/или другие ресурсы. Выходом может быть, например, программа, банковская услуга, готовое (или промежуточное) изделие любой основной категории продукции. Существуют возможности сделать измерения на входе, на различных стадиях процесса, а также на выходе.

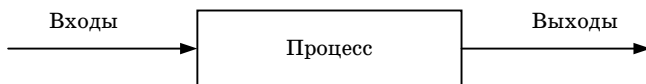


Рис. 1.1. Структура работы

Как показано на рис. 1.2, входы и выходы могут быть нескольких типов: связанные с продукцией (сплошные линии на рис. 1.2) (например, сырье, готовое изделие) и связанные с информацией (пунктирные линии) (например, требования к продукции, информационные характеристики). Данный рисунок представляет процессы поставщика с процессами субпоставщиком и потребителем в сети поставок. В структуре этой сети различные входные и выходные факторы перемещаются в разных направлениях. Термин “продукция” относится здесь ко всем четырем основным категориям продукции.

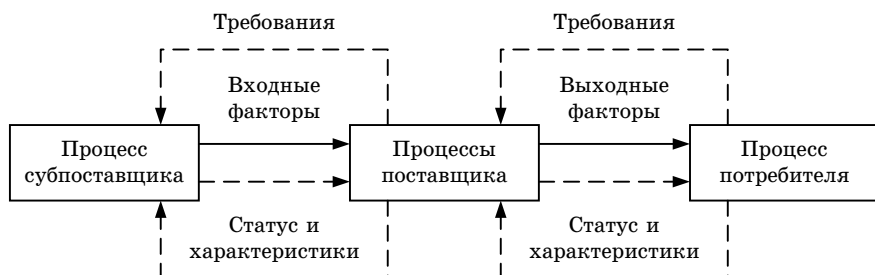


Рис. 1.2. Взаимосвязь процессов в сети поставок при наличии потоков, связанных с продукцией и информацией

Административное управление качеством осуществляется с помощью управления процессами в организации. Управление процессом имеет две стороны:

- управление структурой и функционированием самого процесса, в рамках которого перемещается продукция или информация;
- управление качеством продукции и информации внутри структуры.

Принимая во внимание сложную структуру большинства организаций, важно выделить основные процессы, а также упростить и ранжировать процессы в зависимости от целей административного управления качеством. Примером сложной сети процессов может служить организация, разрабатывающая программную систему согласно ИСО/МЭК 12207 и DO-178.

Любая организация должна определить и установить свою сеть процессов и интерфейсов, и управлять ею. Организация создает, совершенствует и обеспечивает постоянный уровень качества своей продукции с помощью выполнения сети процессов. Это концептуальная основа стандартов серии ИСО 9000. Процессы и их интерфейсы должны быть объектами анализа и постоянного совершенствования в целях обеспечения качества производимой продукции.

Следует заметить, что целью системы качества является выполнение требований к качеству в результате процессов, осуществляемых поставщиком. Но требования к системе качества направлены на процедуры, применяемые в этих процессах. Поэтому в стандартах серии ИСО 9000, устанавливающих модели для обеспечения качества, при формулировке требований пишут: “Поставщик должен установить и выполнять документально оформленные процедуры...”.

При оценке систем качества любой организации стандарт ИСО 9000-1 рекомендует задать три важных вопроса относительно каждого оцениваемого процесса сети.

Определены ли эти процессы и документированы ли их процедуры?

Применяются ли эти процессы в полной мере и выполняются ли они согласно документации?

Эффективны ли эти процессы в достижении ожидаемых результатов?

Результат оценки – совокупность ответов на эти вопросы, связанные соответственно с подходом, применением и результатом. Оценка системы качества может различаться по охватываемой области и включать различные виды деятельности.

Одним из важнейших видов такой деятельности, выполняемой руководством организации систематически согласно стандартам ИСО 9001, 9002, 9003, является оценка статуса и адекватности системы

качества. Выводы, сделанные в процессе оценки системы качества, должны вести к повышению ее эффективности и экономичности. Источником информации для таких выводов являются также результаты внутренних и внешних проверок системы качества.

Внутренние проверки качества, проводимые самой организацией (первая сторона), обеспечивают информацию для эффективного анализа со стороны руководства и корректирующих, предупреждающих и усовершенствующих действий.

Внешние проверки, проводимые заказчиками продукции (второй стороной) и независимыми органами (третьей стороной), обеспечивают, соответственно, доверие заказчика к поставщику и получение сертификата, обеспечив тем самым доверие к целому ряду потенциальных потребителей продукции организации.

Необходимо также обратить внимание на то, в каких ситуациях могут применяться стандарты серии ИСО 9000 и способ использования данной серии поставщиком.

Международные стандарты серии ИСО 9000 предназначены для применения в следующих четырех ситуациях.

1. При административном управлении качеством. Система качества в этой ситуации должна повысить свою собственную эффективность, чтобы выполнить требования к качеству продукции экономичным и оптимальным способом.

2. При заключении контракта между первой и второй стороной. В данной ситуации потребитель требует, чтобы определенные элементы и процессы системы качества стали частью системы качества поставщика, указывая при этом конкретную модель обеспечения качества.

3. При утверждении или регистрации второй стороной. Это та ситуация, в которой система качества оценивается заказчиком. Поставщик может получить официальное признание соответствия его продукции стандарту.

4. При сертификации или регистрации третьей стороной. В этой ситуации систему качества оценивает орган по сертификации, и организация соглашается поддерживать такую систему качества для всех потребителей своей продукции.

Поставщик может выбрать любой из двух способов использования стандартов серии ИСО 9000: “способ, мотивированный руководством” и “способ, мотивированный заинтересованным лицом”. Наиболее распространенным считается второй способ.

При использовании способа, мотивированного заинтересованным лицом, поставщик изначально вводит систему качества как ответ на

непосредственные требования потребителей. Система качества должна соответствовать требованиям стандартов ИСО 9001, 9002, 9003. Руководство организации играет ведущую роль при этом способе, но движущей силой является внешнее заинтересованное лицо (потребители).

При использовании способа, мотивированного руководством, именно руководство организации начинает прилагать усилия по определению будущих потребностей и тенденций рынка. Инструкцией по первоначальному установлению системы качества, повышающей качество продукции, является стандарт ИСО 9004-1 (и другие части ИСО 9004). Далее поставщик может применить стандарты ИСО 9001, 9002 или 9003, как модель обеспечения качества для демонстрации адекватности системы качества с целью получения сертификата. Система качества, реализуемая этим способом, более емкая и плодотворная, чем реализуемая первым способом.

В стандартах серии ИСО 9000 уделяется пристальное внимание подготовке и использованию документации, как виду деятельности, добавляющему стоимость. Соответствующая документация играет значительную роль в следующих видах деятельности по обеспечению качества:

- в достижении требуемого качества продукции;
- в оценке систем качества;
- в повышении качества;
- в сохранении достигнутого уровня качества.

При внутренних и внешних проверках документация на процедуры свидетельствует о том, что процессы определены, процедуры утверждены и находятся под контролем. Только в данных обстоятельствах проверки гарантируют полную оценку адекватности применения и выполнения сети процессов организации.

Кроме того, документация играет немаловажную роль в повышении качества продукции. Если процедуры документированы, применяются и выполняются, то существует возможность определить, как они выполняются.

В целях административного управления качеством организации стандарты серии ИСО 9000 должны использоваться для того, чтобы разработать, внедрить и совершенствовать систему качества как при подходе, мотивированном руководством, так и при подходе, мотивированном заинтересованным лицом. Данная серия содержит два типа стандартов: руководство по применению стандартов в целях обеспечения качества и специальное руководство по административному управлению качеством.

В стандартах ИСО 9001, ИСО 9002, ИСО 9003 элементы системы качества определены и сгруппированы таким образом, что создаются три различные модели, пригодные для демонстрации поставщиками своих возможностей и для оценки этих возможностей второй и третьей стороной.

ИСО 9001 применяется в том случае, когда необходимо продемонстрировать соответствие заданным требованиям на этапах проектирования, разработки, производства, монтажа и технического обслуживания.

ИСО 9002 используется, когда соответствие заданным требованиям должно обеспечиваться поставщиком на этапах разработки, производства, монтажа и технического обслуживания.

Применение ИСО 9003 наиболее целесообразно в случае, когда необходимо продемонстрировать соответствие заданным требованиям на заключительных этапах – контроле готовой продукции и заключительных испытаниях.

В табл. 1.1 приведены требования, которые включены в модели обеспечения качества и перечень перекрестных ссылок на номера пунктов стандартов, содержащих данные требования.

Далее более подробно рассмотрим стандарты ИСО 9004-1, ИСО 9000-3, в которых определены, соответственно, руководящие положения по административному управлению качеством и элементы системы качества по применению стандарта ИСО 9001 при проектировании, разработке, производстве, монтаже и обслуживании программного обеспечения.

1.2. Руководящие положения по административному управлению качеством и элементы системы качества

Любая организация, в том числе и занимающаяся разработкой программных систем, если она намерена разработать и внедрить систему качества, должна обратиться к стандарту ИСО 9004-1. Для решения этой задачи руководство организации должно держать под постоянным контролем технические, административные и человеческие факторы, влияющие на качество производимой продукции. Контроль должен направляться, прежде всего, на предупреждение выпуска несоответствующей продукции, а затем уже на выявление данного несоответствия, если оно имеется, его устранение и сокращение. Разработка и внедрение системы качества должно осуществляться для достижения целей, определенных политикой организации в области качества.

Таблица 1.1

Внешнее обеспечение качества			Руководство по применению ИС О9000-2	Название пункта в ИСО 9001	Руководящие положения по административному управлению качеством ИСО 9004-1	Маршрутная карта ИСО 9000-1
Требования						
ИСО 9001	ИСО 9002	ИСО 9003				
—	—	0	4,1	Ответственность руководства	4	4,1; 4,2; 4,3
—	—	0	4,2	Система качества	5	4,4; 4,5; 4,8
—	—	—	4,3	Анализ контракта	+	8
—	+	+	4,4	Управление проектированием	8	
—	—	—	4,5	Управление документацией и информацией	5,3; 11,5	
—	—	+	4,6	Закупка продукции	9	
—	—	—	4,7	Контроль продукции, поставляемой потребителем	+	
—	—	0	4,8	Идентификация и отслеживание продукции	11,2	5
—	—	+	4,9	Управление технологическим процессом	10;11	4,6; 4,7
—	—	0	4,10	Контроль и испытание	12	
—	—	—	4,11	Проверка оборудования	13	
—	—	—	4,12	Статус контроля и испытаний	11,7	
—	—	0	4,13	Контроль несоответствующей продукции	14	
—	—	—	4,14	Корректирующее и предупреждающее действие	15	
—	—	—	4,15	Погрузочные работы, хранение, доставка	10,4; 16,1; 16,2	
—	—	0	4,16	Управление данными о качестве	5,3; 17,2; 17,3	
—	—	0	4,17	Внутренние проверки качества	5,4	4,9
—	—	0	4,18	Обучение	18,1	5,4
—	—	+	4,19	Техническое обслуживание	16,4	
—	—	0	4,20	Статистические методы	20	
				Экономика качества	6	
				Безопасность продукции	19	
				Маркетинг	7	

Условные знаки: – – исчерпывающие требования; + – требования, менее исчерпывающие, чем содержащиеся в ИСО 9001 и 9002; о – отсутствующий элемент.

Данный стандарт содержит широкий перечень элементов системы качества, относящихся ко всем стадиям и видам деятельности на протяжении всего жизненного цикла (ЖЦ) продукции. Система качества, чтобы быть максимально эффективной, должна соответствовать конкретному виду деятельности и выпускаемой продукции.

Необходимо отметить, стандарт ИСО 9004-1 является документом для внутреннего пользования организации и не должен применяться при заключении контракта или как инструмент регламентации или сертификации. Соответственно, он не является инструкцией по применению ИСО 9001, ИСО 9002, ИСО 9003.

Основной задачей любой организации, занимающейся разработкой программных систем, должно быть обеспечение требуемого качества. В результате успешной деятельности таких организаций должна быть разработана программная система, которая отвечает определенным требованиям, сфере применения и назначения; удовлетворяет заказчиков; соответствует стандартам и техническим условиям; имеет конкурентоспособную цену; экономична с точки зрения затрат на ее разработку.

Система качества организации имеет два взаимосвязанных аспекта. Первый аспект – требования и ожидания заказчика. Заказчик должен быть уверен в том, что организация имеет возможность разработать программную систему требуемого качества и поддерживать этот уровень качества. Второй аспект – требования и интересы организации. Организация должна оптимальным образом использовать имеющиеся у нее ресурсы для достижения и поддержки требуемого уровня качества программной системы. Каждый аспект требует объективных сведений как о системе качества, так и о качестве ПС.

Эффективная система качества должна быть спроектирована так, чтобы удовлетворить ожидания и требования заказчика и, в то же время, защитить интересы организации. Хорошо организованная система качества является надежным средством оптимизации и управления качеством программной системы с точки зрения снижения таких факторов как риск, затраты и увеличения прибыли организации.

Руководство высшего уровня в организации берет на себя обязательства и ответственность за политику в области качества. Как уже отмечалось выше, деятельность по административному управлению качеством охватывает все те операции общей функции управления, которые определяют политику в области качества, задачи и обязанности, и выполняет их с помощью таких средств как плани-

рование, управление, обеспечение и улучшение качества в рамках системы качества.

Политика в области качества согласовывается с другими направлениями деятельности организации, при этом руководство принимает меры, обеспечивающие понимание этой политики, ее осуществление и анализ. Проведение определенной политики и достижение поставленной цели обеспечивается путем разработки, создания и внедрения системы качества. При этом еще раз напомним основной упор системы качества делается на предупреждение проблем, а не на их обнаружение после возникновения.

Воздействие системы качества распространяется на все этапы и виды деятельности жизненного цикла ПС.

К основным видам деятельности, влияющим на качество программной системы, относятся:

- маркетинг и изучение рынка;
- проектирование и разработка;
- процесс планирования;
- проверки;
- техническое обслуживание.

Типичные стадии срока службы продукции, независимо от того, к какой категории она относится, представлены на рис. 1.3.



Рис. 1.3. Типичные виды деятельности, влияющие на качество

В рамках взаимодействия всех видов деятельности организации маркетинг и проектирование имеют особое значение для:

- установления и формулирования требований и ожиданий потребителя и требований к программной системе;
- формирования концепций, обеспечивающих разработку программной системы в соответствии с установленными в документах техническими условиями при оптимальных затратах.

Все виды деятельности, которые прямо или косвенно влияют на качество, должны быть определены и зафиксированы документально. Должны быть предусмотрены следующие меры.

1. Четкое определение общей и конкретной ответственности в области качества.

2. Четкое определение ответственности и полномочий относительно каждого вида деятельности, влияющего на качество. Определение обязанностей и полномочий должно обеспечивать достижение поставленных целей на заданном уровне эффективности.

3. Определение мероприятий, обеспечивающих координацию и управление взаимосвязанными видами деятельности.

4. Особое внимание при организации хорошо структурированной и эффективной системы качества следует уделять выявлению имеющихся или потенциальных проблем качества, проведению предупреждающих и корректирующих действий.

В рамках общей организационной структуры следует четко определить функции, связанные с системой качества, иерархию полномочий и связей. Руководству организации следует определить требования к ресурсам и выделять их в необходимом объеме, обеспечивающем проведение политики в области качества и реализацию поставленных целей. Такие ресурсы могут включать:

- людские ресурсы и специальные знания;
- оборудование, необходимое для проведения разработки системы;
- контрольное, испытательное и проверочное оборудование;
- инструментальные средства и среду разработки.

Особое внимание руководству организации следует уделить определению уровня компетенции, квалификации и подготовки персонала, участвующего в разработке программной системы.

Система качества должна быть организована таким образом, чтобы осуществлялось адекватное и постоянное управление всеми видами деятельности, влияющими на качество. Для осуществления политики и достижения целей в области качества необходимо разработать, опубликовать и поддерживать документированные рабочие

процедуры, координирующие различные виды деятельности, определяющие их цели и характеристики, и обеспечивающие эффективные системы качества.

Все элементы, требования и положения, принятые организацией для своей системы качества, должны быть документированы как политика и процедуры (т. е. инструкциями по проектированию, закупке и обработке) в систематической и понятной форме. Система качества должна включать адекватные положения по необходимой идентификации, распределению, сбору и ведению документации по качеству. Типичной формой основного документа для демонстрации или описания системы качества является руководство по качеству, определенное в ИСО 10013. Основным назначением руководства по качеству является определение общей структуры системы качества и выполнение функций постоянного справочного материала при внедрении и ведении системы.

Руководство организации должно обеспечить разработку и ведение документированных программ качества для системы или процессов. Программы должны быть согласованы со всеми другими требованиями к системе качества и обеспечивать удовлетворение заданных требований к программной системе, а также требований проекта или контракта. Программы качества должны определить:

- цели в области качества (т. е. технические требования, эффективность, затраты и т. д.);
- операции процессов, которые образуют рабочую практику организации;
- конкретное распределение обязанностей, полномочий и ресурсов на различных стадиях жизненного цикла системы;
- применяемые процедуры и инструкции по документированию;
- применяемые программы испытаний, контроля, проверок на соответствующих стадиях жизненного цикла программной системы;
- документированную процедуру изменений и модификаций программы качества по мере выполнения проекта;
- метод определения достигнутых целей в области качества;
- другие действия, необходимые для достижения целей.

Для того чтобы установить, соответствует ли деятельность и связанные с ней результаты функционирования системы качества организации заданным требованиям, и определить эффективность системы качества, следует проводить плановые проверки. Все элементы должны быть предметом регулярной внутренней проверки и оценки с учетом статуса и значения проверяемой деятельности. Для этих целей руководство организации должно разрабатывать и составлять

соответствующую программу проведения проверки, которая включает следующее:

- планы и графики различных видов деятельности, подлежащих проверке;
- назначение персонала, обладающего необходимой квалификацией для проведения проверки;
- документированные процедуры проведения проверок, включая отчетность о результатах проверки качества и о принятых корректирующих действиях по устранению недостатков.

Сфера проверки включает следующие объекты:

- организационную структуру;
- административные и рабочие процедуры, а также процедуры, проводимые в рамках системы качества;
- людские и материальные ресурсы, оборудование;
- рабочие участки, операции и процессы;
- непосредственно программную систему;
- документацию.

Руководство организации должно обеспечить проведение независимого анализа и оценки системы качества через определенные промежутки времени. Анализы политики и целей в области качества проводятся высшим руководством, а анализ прочей деятельности должен выполняться исполнительным руководством или другими лицами с помощью компетентного независимого персонала.

Проведенный анализ с изложением обоснованных и всесторонних оценок должен включать следующие пункты:

- результаты внутренней проверки;
- эффективность удовлетворения требований ИСО 9004-1, изложенных здесь, и поставленных организацией целей и политики в области качества;
- предложения по модернизации системы качества в сложившихся условиях.

Очень важно, чтобы эффективность системы качества измерялась финансовыми показателями. Влияние эффективной системы качества на прибыль и потери организации могут быть весьма значительными, особенно при усовершенствовании операций, приводящих к уменьшению потерь, происходящих из-за ошибок. Подобные меры и отчетность могут предоставить средства для определения неэффективной деятельности и начала внутренних улучшений.

Есть различные подходы к сбору, предоставлению и анализу элементов финансовых данных, зависящих от структуры, деятельности и зрелости систем качества.

1. Подход, определяемый затратами на обеспечение качества. Этот подход оперирует с затратами, которые разделены в широком смысле на затраты, связанные с внутренней деятельностью, и затраты, связанные с внешней деятельностью. Затраты, связанные с внутренней деятельностью, анализируются по модели RAF (предупредительные меры, оценка, отказ). Затраты, связанные с предупредительными мерами и оценкой, считаются инвестицией, тогда как затраты, связанные с отказами, считаются потерей. Эти затраты распределяются следующим образом:

- на предупредительные затраты: усилия по предотвращению отказа;
- на оценку: испытания, контроль и исследования с целью оценки выполнения требований к качеству;
- на устранение внутреннего отказа: расходы, вытекающие из неспособности программной системы выполнять требования к качеству перед сдачей его заказчику;
- на устранение внешних отказов: затраты, вытекающие из неспособности ПО, выполнять требования к качеству после сдачи его заказчику.

2. Подход, определяемый затратами на процесс. При данном подходе анализируются затраты на обеспечение соответствия процесса и затраты на устранения несоответствия процесса.

3. Подход, определяемый потерями качества. При этом подходе внимание фокусируется на внутренних и внешних (материальных и нематериальных) потерях из-за плохого качества. Типичным примером внешних нематериальных потерь является потеря будущих пользователей программной системы вследствие его неудовлетворительного качества. Типичным примером внутренних нематериальных потерь является понижение эффективности производства вследствие переработки продукции, упущенных возможностей и т. д. Материальные потери – это затраты, связанные с внутренними и внешними отказами.

Подходы, перечисленные выше, считаются целесообразными, хотя они не исключают других подходов, адаптаций или их комбинаций. По завершению каждой стадии разработки проекта следует планировать и провести официальный, документированный, системный и критический анализ результатов, достигнутый на этой стадии проекта. Каждый этап анализа проекта должен предусматривать участие представителей всех подразделений, отвечающих за функции, влияющих на качество системы, в зависимости от рассматриваемой стадии. В результате анализа проекта определяются и прогнозируются

ются области возникновения проблем и несоответствий, а также проведение корректирующих действий, обеспечивающих соответствие окончательного проекта и дополнительных данных требований потребителя.

В зависимости от стадии проекта и рассматриваемой продукции должны учитываться следующие элементы.

1. Элементы, относящиеся к требованиям потребителя и их удовлетворению.

2. Элементы, относящиеся к техническим условиям на продукцию.

3. Элементы, относящиеся к техническим условиям на процесс.

Выше была определена лишь незначительная часть вопросов, которая должна быть рассмотрена при административном управлении качеством. Объем данного пособия не позволяет в полной мере рассмотреть основные аспекты административного управления качеством. Для более детального изучения вопросов в области административного управления качеством и систем качества следует обратиться к ИСО 9004.

1.3. Применение ИСО 9001 при разработке программного обеспечения

С прогрессом в области информационных технологий увеличилось количество продукции программного обеспечения и, соответственно, возросла роль управления качеством этой продукции. Одним из путей создания системы управления качеством является разработка руководящих положений по обеспечению качества программного обеспечения.

Требования к общей системе качества при двусторонней контрактной схеме опубликованы в стандарте ИСО 9001. Однако процесс разработки и обслуживания программного обеспечения отличается от такого же процесса для большинства других типов промышленной продукции. Поэтому для такой быстро развивающейся области технологии необходимо разрабатывать дополнительные руководящие положения к системе качества там, где задействована продукция программного обеспечения, принимая во внимание современный уровень развития этой области технологии.

Природа развития программного обеспечения такова, что некоторые виды деятельности связаны лишь с отдельными фазами процесса разработки, тогда как другие могут относиться ко всему процессу. Ниже будут отражены эти различия, а также руководящие положения, содействующие применению стандарта ИСО 9001

организациям, разрабатывающим, поставляющим и обслуживающим продукцию программного обеспечения.

Руководящие положения предназначены для описания предлагаемых средств управления и методов разработки программного обеспечения, отвечающего требованиям покупателя. Это достигается, в первую очередь, предотвращением несоответствия продукции на всех стадиях, начиная от разработки и кончая техническим обслуживанием.

Структура системы качества. Руководство поставщика должно определить и документально оформить свою политику, цели и обязательства в области качества. Поставщик должен обеспечить понимание этой политики, ее осуществление и внедрение на всех уровнях в конкретной организации.

Ответственность, полномочия и взаимодействие всего персонала, который руководит, выполняет и проверяет работу, оказывающую влияние на качество, должны быть четко определены. Особенно это касается персонала, которому необходимы организационная свобода и полномочия для:

- проведения мероприятий, направленных на предупреждение случаев несоответствия продукции;
- выявления и регистрация любых проблем в области качества продукции;
- инициирования, выработки рекомендаций или обеспечения выполнения решений в установленном порядке;
- проверки выполнения решений;
- контроля за дальнейшей обработкой несоответствующей продукции, ее поставкой или монтажом до тех пор, пока выявленные дефекты или неудовлетворительные условия не будут устранены.

Поставщик должен определить требования к внутренней проверке, обеспечить необходимые средства и назначить специально подготовленный персонал для ее проведения. Проверка должна включать контроль, испытание и регулирование процессов проектирования производства, монтажа и обслуживания и/или продукции. Анализ проекта, проверка системы качества процессов и/или продукции должны выполняться персоналом, независимым от тех, кто несет непосредственную ответственность за выполненную работу.

Поставщик должен назначить представителя руководства, который независимо от других обязанностей должен иметь определенные полномочия и нести ответственность за выполнение и соблюдение требований стандарта ИСО 9001.

Система качества, удовлетворяющая требования ИСО 9001, должна периодически анализироваться руководством поставщика с тем, чтобы гарантировать постоянную пригодность и эффективность системы. Следует вести протоколы подобных анализов.

Покупатель должен сотрудничать с поставщиком, чтобы своевременно обеспечить его всей необходимой информацией и разрешить возникающие проблемы.

Покупатель должен назначить представителя, ответственного за связь с поставщиком по вопросам контракта. Этот представитель должен иметь полномочия решать следующие вопросы (но не ограничиваться ими):

- определять требования покупателя к поставщику;
- отвечать на вопросы поставщика;
- принимать предложения поставщика;
- заключать соглашения с поставщиком;
- гарантировать соблюдение организацией, представляющей покупателя, соглашений, заключенных с поставщиком;
- определять критерии приемки продукции;
- принимать решения по тем элементам программного обеспечения, которые признаны непригодными для использования.

Регулярный совместный анализ, проводимый покупателем и поставщиком, должен планироваться с тем, чтобы охватить следующий круг вопросов:

- соответствия программного обеспечения техническому заданию;
- результаты контроля;
- результаты приемочных испытаний.

Результаты такого анализа должны быть согласованы и зарегистрированы.

Поставщик должен разработать и документально оформить систему качества. Система качества должна представлять собой единый процесс, проходящий через весь жизненный цикл продукции, гарантируя тем самым качество, которое формируется в ходе разработки. Упор необходимо делать на предупреждение появления дефектов, а не на исправление их после возникновения.

Поставщик должен гарантировать эффективную реализацию документально оформленной системы качества. Все элементы, требования и положения системы качества должны быть четко представлены в соответствующих документах.

Поставщик обязан подготовить и документально оформить план качества, с тем, чтобы, во-первых, выполнить мероприятия по обеспечению качества для каждой разработки ПО на базе системы каче-

ства и, во-вторых, обеспечить ее понимание и соблюдение заинтересованными организациями.

Поставщик должен разработать законченную систему плановых внутренних проверок качества, чтобы удостовериться в деятельности по обеспечению качества запланированных мероприятий и определить эффективность системы качества. Проверки должны планироваться исходя из статуса и важности различных видов деятельности.

Проверки и последующие действия должны проводиться в соответствии с документально оформленными процедурами.

Результаты проверок должны оформляться документально и доводиться до сведения персонала, ответственного за проверенный участок работы. Руководящий персонал, ответственный за этот участок, должен предпринять своевременные меры по устранению недостатков, выявленных в ходе проверки.

Поставщик должен разрабатывать, документально оформлять и выполнять процедуры, обеспечивающие:

- выявление причин несоответствия продукции и выбор корректирующих воздействий, предупреждающих повторение дефектов;

- анализ всех процессов, рабочих операций, отклонений от требований контрактов, зарегистрированных данных по качеству, отчетов об использовании; рекламации пользователей в целях выявления и устранения потенциальных причин несоответствия продукции;

- проведение профилактических действий для решения проблем на уровне, соответствующем реальному риску;

- осуществление контроля с тем, чтобы удостовериться в действительной реализации и эффективности корректирующих воздействий;

- внедрение изменений в процедуры, вызванные корректирующими воздействиями, и их регистрацию.

Система качества – жизненный цикл. Разработка ПО должна осуществляться в соответствии с моделью жизненного цикла. Действия, связанные с обеспечением качества, должны планироваться и проводиться с учетом особенностей выбранной модели ЖЦ.

Поставщик должен устанавливать и выполнять процедуры, обеспечивающие проведение анализа контракта и координацию этой деятельности.

Каждый контракт должен быть изучен поставщиком для гарантии того, что:

- область действия контракта, а также требования, определены и документально оформлены;
- вероятные случайности или риск идентифицированы;
- информация, являющаяся собственностью фирмы, достаточно защищена;
- любые требования, отличные от тех, которые содержатся в заявке на контракт, нашли необходимое решение;
- поставщик имеет возможности выполнить обязательства, определенные в контракте;
- ответственность поставщика в отношении подрядных работ определена;
- терминология согласована обеими сторонами;
- покупатель имеет возможность выполнить обязательства, определенные в контракте.

Для разработки программного обеспечения поставщик должен иметь полный недвусмысленный набор функциональных требований. Кроме того, эти функциональные требования должны отражать все аспекты, необходимые для удовлетворения потребностей покупателя.

Функциональные требования должны быть сформулированы достаточно точно, с тем, чтобы производить оценку во время приемки продукции.

Эти требования фиксируются в техническом задании (ТЗ). В некоторых случаях этот документ разрабатывается покупателем. В других случаях он разрабатывается поставщиком в тесном сотрудничестве с покупателем; при этом поставщик должен получить согласие покупателя прежде, чем начнется стадия разработки.

Техническое задание, как часть документации на разработку, должно быть объектом контроля со стороны действий (процессов) по документированию и управлению конфигурацией.

Все интерфейсы между разрабатываемым программным обеспечением и существующим ПО и/или аппаратными средствами должны быть полностью определены либо непосредственно, либо путем ссылок в техническом задании.

В процессе разработки технического задания рекомендуется обратить внимание на следующие вопросы:

- назначение лиц (с обеих сторон), ответственных за разработку ТЗ;
- методы согласования требований и утверждения изменений;
- применение общепринятой терминологии в данной предметной области, а также предотвращение ситуаций по неправильному толкованию применяемых терминов;
- запись и изучение результатов дискуссий обеими сторонами.

Процесс разработки необходимо планировать. План разработки должен охватывать следующее:

- описание предметной области (проекта), включая постановку задачи;
- организацию ресурсов под конкретный проект, включая состав команды, обязанности членов команды, привлекаемых субподрядчиков и материальные затраты;
- фазы разработки;
- программу работ над проектом, устанавливающую задачи, которые должны быть решены, ресурсы и время, необходимые для решения каждой задачи;
- согласование следующих планов, разрабатываемых в ходе проекта: план качества; план управления конфигурацией; план комплектации; план проведения испытаний.

План разработки должен корректироваться по мере совершенствования разработки, и каждая фаза должна быть определена до того, как начнутся работы на этой фазе. План должен быть рассмотрен и утвержден до начала его реализации.

План разработки должен устанавливать упорядоченный процесс или методологию преобразования технического задания в продукцию программного обеспечения. Он может включать в себя распределение работ по фазам и определение:

- фаз (этапов, стадий, работ, процессов) разработки, которые должны быть выполнены;
- необходимых затрат для каждой фазы;
- требуемых результатов по каждой фазе;
- процедур проверки, которые необходимо провести на каждой фазе;
- потенциальных проблем, связанных с фазами разработки и с выполнением установленных требований.

План разработки должен определять, как управлять проектом, и включать:

- график разработки продукции, выполнения контракта и связанных с ним поставок;
- организацию контроля за ходом выполнения графика;
- распределение ответственности и ресурсов по работам, включенным в график;
- организацию интерфейсов между различными группами, входящими в состав команды проекта.

План разработки должен устанавливать методы, обеспечивающие правильность выполнения всех работ. Он может включать правила,

практические методы и накопленный опыт по разработке; средства и технические приемы, используемые для разработки; управление конфигурацией.

Анализ хода выполнения работ следует планировать, проводить и документально оформлять, с тем, чтобы обеспечить решение спорных вопросов, касающихся распределения ресурсов, и гарантировать эффективное выполнение планов проекта.

Необходимые затраты по каждой фазе должны быть определены и документально оформлены. Каждое требование должно быть определено таким образом, чтобы его выполнение можно было проверить. Вопрос о неполных, двусмысленных или противоречивых требованиях должны решать лица, ответственные за разработку этих требований.

Результаты, требуемые по каждой фазе разработки, должны быть определены и документально оформлены. Они должны быть проверены на предмет удовлетворения следующих условий:

- отвечать требованиям, установленным для каждой фазы;
- содержать критерии приемки или ссылки на них для перехода к следующей фазе;
- соответствовать принятой практике и накопленному опыту по разработке независимо от того, оговорены ли они во входной информации;
- идентифицировать те характеристики программного обеспечения, которые являются наиболее важными для ее безопасности и эффективного функционирования;
- соответствовать действующим нормативным требованиям.

Поставщик должен составить план проверки всех результатов разработки в конце каждой фазы. Проверка разработки должна установить, что полученные результаты отвечают требованиям, установленным в начале фазы. Эти проверки необходимо проводить, основываясь на выполнении следующих мероприятий по контролю разработки:

- осуществление анализов в ходе фаз разработки через установленные интервалы времени;
- сравнение нового проекта с апробированным аналогичным проектом, если таковой имеется;
- проведение испытаний и демонстрационных показов.

Результаты проверок и последующих действий, необходимых для гарантии того, что установленные требования выполнены, должны быть зафиксированы и проверены после того, как соответствующие действия завершатся.

Поставщик должен подготовить план качества как часть работ по планированию разработки. План качества должен корректироваться в ходе выполнения работ, а пункты, касающиеся каждой фазы, должны быть полностью определены к началу этой фазы.

План качества должен быть официально рассмотрен и согласован со всеми организациями, заинтересованными в его реализации.

Документ, описывающий план качества, может быть самостоятельным документом (озаглавленным “План качества”) или частью другого документа. Кроме того, он может быть составлен из нескольких документов, включая план разработки.

План качества должен определять или давать ссылки на следующие пункты:

- цели качества, выраженные в измеряемых показателях, если это возможно;
- заданные критерии по затратам и результатам для каждой фазы разработки;
- идентификацию видов деятельности, связанную с испытаниями, проверками и оценками, которые должны быть проведены;
- подробное планирование испытаний, проверок и оценок, включая графики, ресурсы и назначенных ответственных;
- конкретное распределение ответственности за мероприятия по обеспечению качества, такие как: анализы и испытания; управление конфигурацией и контроль за изменениями; контроль дефектов и выполнение корректирующих действий.

Проектирование и реализация. Проектирование и реализация — это те виды деятельности, которые трансформируют техническое задание в программное обеспечение. Из-за сложности этой продукции вся деятельность должна осуществляться в строго установленном порядке, с тем, чтобы производить продукцию в соответствии с техническим заданием, а при обеспечении качества не следует чрезмерно полагаться на действия, связанные с испытанием и проверкой.

В дополнение к требованиям, общим для всех фаз разработки, необходимо принять во внимание следующие аспекты, присущие деятельности по проектированию:

- идентификацию конструктивных соображений в дополнение к требованиям, касающихся выходных данных и ожидаемых результатов; следует рассмотреть такие аспекты, как правила проектирования и определения внутреннего интерфейса;

– методологию проектирования, которая должна быть использована при системном проектировании соответствующего вида продукции;

– использование прошлого опыта в проектировании; учитывая уроки, извлеченные из опыта прошлого проектирования, поставщик должен избегать повторений одних и тех же, или аналогичных, проблем;

– последующие процессы; продукция должна быть спроектирована так, чтобы можно было без помех проводить испытания, техническое обслуживание и использование.

В дополнение к требованиям, общим для всех видов деятельности, связанных с разработкой программного обеспечения, необходимо рассмотреть следующие аспекты для каждого вида деятельности по реализации проекта:

– следует установить и соблюдать правила программирования, языка программирования, согласованные правила наименования, кодирования и соответствующего разъяснения;

– поставщик должен использовать соответствующие методы и средства реализации, чтобы выполнить требования покупателя.

Поставщик должен проводить анализ с целью гарантии того, что требования выполняются, и описанные выше методы применяются правильно. Процесс проектирования или реализации должен продолжаться до тех пор, пока последствия всех выявленных недостатков не будут положительно разрешены или пока не будет известна степень риска в случае продолжения работ другими методами. Следует вести протоколы таких анализов.

Проведение испытания может быть необходимо на различных уровнях, начиная от отдельного элемента ПО и кончая готовой продукцией. Существует несколько различных подходов к испытаниям. В некоторых случаях оценка, испытания на месте и приемочные испытания могут быть одним и тем же видом деятельности. Документ, описывающий план испытаний, может быть самостоятельным документом (озаглавленным “План испытаний”) или частью другого документа или может быть составлен из нескольких документов.

Система качества – вспомогательные виды деятельности.

Наиболее важными вспомогательными видами деятельности являются управление конфигурацией и осуществление контроля за документацией.

Управление конфигурацией обеспечивает механизм идентификации, контроля и отслеживания вариантов каждого элемента, входящего в состав программного обеспечения. Во многих случаях более

ранние варианты, которые все еще продолжают использоваться покупателем, должны сопровождаться и находиться под контролем.

Система управления конфигурацией должна:

- однозначно идентифицировать варианты каждого элемента программного обеспечения, совокупность которых образуют конкретный вариант готовой продукции;

- идентифицировать состояние компоновки программного обеспечения, находящегося в разработке или уже поставленного покупателю;

- управлять модернизацией конкретного элемента программного обеспечения, проводимой более, чем одним человеком;

- обеспечить координацию работ по модернизации многочисленной продукции, производимой в одном или более местах, по необходимости;

- идентифицировать и отслеживать все мероприятия и изменения, вызванные изменившейся заявкой, начиная от замысла до выпуска продукции.

Поставщик должен разработать и реализовать план управления конфигурацией, который включает в себя следующее:

- организации, занятые в управлении конфигурацией, и ответственность, возложенную на каждую из них;

- виды деятельности по управлению конфигурацией, которые должны быть осуществлены;

- технические средства, технологии и методологические принципы, которые должны быть применены в управлении конфигурацией;

- стадии, на которых элементы программного обеспечения должны быть подвергнуты управлению конфигурацией.

К видам деятельности, связанной с управлением конфигурацией, относятся идентификация конфигурации, контроль изменений, установление отчета о статусе конфигурации.

Поставщик должен установить и осуществлять процедуры по идентификации элементов программного обеспечения на всех фазах проекта. Каждый отдельный элемент ПО должен иметь свою собственную и отличную от других идентификацию.

Процедуры должны применяться для гарантии того, что для каждого варианта элемента ПО могут быть идентифицированы:

- функциональные и технические требования;

- все технические средства, используемые при разработке, которые влияют на функциональные и технические требования;

- все интерфейсы с другими элементами ПО и с аппаратными средствами;

— все документы и компьютерные файлы, имеющие отношение к конкретному элементу ПО.

Для выпущенной продукции необходимо установить процедуры, облегчающие отслеживаемость элемента или продукции ПО.

Поставщик должен установить и выполнять процедуры по идентификации, документальному оформлению, анализу и санкционированию любых изменений в элементах ПО в рамках управления конфигурацией. Все изменения в элементах ПО должны проводиться в соответствии с этими процедурами.

Поставщик должен установить и обеспечить регистрацию, управление и предоставление отчетов о статусе ПО, заявок на изменение и реализацию утвержденных изменений.

Поставщик должен установить и обеспечить процедуры по контролю всех документов. К данным процедурам относятся:

— определение тех документов, которые должны быть объектом контроля;

— утверждение и опубликование;

— изменение, включая отмену и, если необходимо, выпуск.

Процедуры по контролю за документацией должны применяться к соответствующим документам, включая следующие:

— процедурные документы, описывающие систему качества, которая должна применяться на протяжении всего ЖЦ;

— документы по планированию, описывающие планирование и развитие всех видов деятельности поставщика, а также взаимодействие с покупателем;

— документы на программное обеспечение, включая: информацию на входе фазы разработки; ожидаемые результаты в конце фазы разработки; планы и результаты проверок и оценок; документацию для покупателя и пользователя; эксплуатационную документацию.

Все документы должны быть изучены и утверждены уполномоченными должностными лицами до их опубликования. Действующие процедуры по документированию должны гарантировать то, что соответствующие документы имеются в наличии на соответствующих участках там, где выполняются операции, важные для эффективного функционирования системы качества, а также то, что устаревшие документы быстро извлекаются из употребления.

Там, где используются компьютерные файлы, особое внимание следует обратить на соответствующие процедуры утверждения, доступа, распределения и архивного хранения.

К другим вспомогательным видам деятельности относятся: измерение продукции и процесса; правила, практические методы и на-

копленный опыт; средства и технические приемы; закупка; поставка программного обеспечения третьей стороной; подготовка кадров.

1.4. Показатели качества программных средств в ГОСТ 28195-89 и ГОСТ Р ИСО/МЭК 9126-93

Показатели качества программных средств (ПС) устанавливают ГОСТ 28195-89 “Оценка качества программных средств. Общие положения” и ГОСТ Р ИСО/МЭК 9126-93 “Информационная технология. Оценка программной продукции. Характеристика качества и руководства по их применению”. Одновременное существование двух действующих стандартов, нормирующих одни и те же показатели, ставит вопрос об их гармонизации. Ниже кратко рассмотрим каждый из перечисленных стандартов.

ГОСТ Р ИСО/МЭК 9126-93 устанавливает шесть характеристик качества программных средств. Под характеристикой качества, согласно этому стандарту, понимается “набор свойств (атрибутов) программной продукции, по которым ее качество оценивается или описывается”. Определения качества и характеристики, установленные в этом стандарте, отражают представление пользователя о качестве программной системы.

Функциональные возможности. Данная характеристика описывает свойства программного средства в части полноты удовлетворения требований пользователя и в этом смысле является определяющей для потребительских свойств средства, в то время как остальные характеристики носят более “технический” характер, что не уменьшает их значения. Кроме того, эти характеристики (такие как надежность, эффективность и др.) могут входить в число требований пользователя.

Требования пользователя четко обусловлены при наличии контракта и, соответственно, технического задания (технических требований). В других случаях речь идет о предполагаемых потребностях, которые должны быть установлены и формально определены какими-либо нормативными документами (стандартами, техническими условиями и пр.). Оценка качества программного средства должна начинаться с точного и формального установления предъявляемых требований, которые могут различаться (и различаются) для различных систем.

Функциональные возможности – набор атрибутов, относящихся к сути набора функций и их конкретным свойствам. Функциями являются те, которые реализуют установленные или предполагаемые потребности. Данный набор атрибутов характеризует то, что

программная система выполняет для удовлетворения потребностей, тогда как другие наборы, главным образом, характеризуют, когда и как это выполняется.

Надежность. Специфика программного средства заключается в том, что оно не подвержено старению и износу, а отказы проявляются из-за ошибок в требованиях, проекте, реализации.

Надежность — набор атрибутов, относящихся к способности программного средства сохранять свой уровень качества функционирования в установленных условиях за определенный период времени.

Практичность. При оценке этой характеристики следует исходить из требований пользователя, так как пользователи разного уровня подготовленности предъявляют разные (часто взаимоисключающие) требования.

Практичность — набор атрибутов, относящихся к объему работ, требуемых для исполнения и индивидуальной оценки такого исполнения определенным или предполагаемым кругом пользователей.

Эффективность. Оценка данной характеристики также критически зависит от требований пользователя. Программное средство может выглядеть неэффективным не в силу плохого кодирования, а в силу противоречивости и нереальности исходных требований. Например, требования к программному средству выполнять функции на технических средствах минимальной (по объему оперативной и дисковой памяти, тактовой частоте и пр.) конфигурации компьютера противоречат требованиям к высокому быстродействию. Вообще говоря, и теория, и практика свидетельствуют, что быстродействие и объем используемой памяти являются взаимодополняющими характеристиками в том смысле, что увеличение одного приводит к увеличению другого при прочих равных условиях.

Эффективность — набор атрибутов, относящихся к соотношению между уровнем качества функционирования программного средства и объемом используемых ресурсов при установленных условиях.

Сопровождаемость. Мобильность. Для этих двух характеристик следует учитывать, что в специфических российских условиях им часто не уделялось ранее и не уделяется сейчас достаточного внимания со стороны пользователя. Эти характеристики связаны с долгосрочным планированием развития программного средства (и эксплуатирующей его организации). Улучшение сопровождаемости и мобильности может, вообще говоря, повысить стоимость средства в настоящий момент времени, что (многократно) окупается лишь через несколько лет.

Сопровождаемость — набор атрибутов, относящихся к объему работ, требуемых для проведения конкретных изменений (модификаций).

Мобильность — набор атрибутов, относящихся к способности программного средства быть перенесенной из одного окружения в другое.

Все приведенные характеристики являются наборами атрибутов и, следовательно, должны уточняться на множестве соответствующих подхарактеристик. ГОСТ Р ИСО/МЭК 9126-93 не устанавливает соответствующих показателей, но в рекомендуемом приложении “А” к этому стандарту дается пример (качественная модель) таких подхарактеристик, называемых комплексными показателями.

В качестве примера приведем комплексные показатели для некоторых характеристик.

Характеристика “Функциональные возможности”:

- пригодность — атрибуты ПС, относящиеся к наличию и соответствию набора функций конкретным задачам;
- правильность — атрибуты ПС, относящиеся к обеспечению правильности или соответствию результатов или эффектов;
- способность к взаимодействию — атрибуты ПС, относящиеся к его способности взаимодействовать с конкретными системами;
- согласованность — атрибуты ПС, которые заставляют программиста придерживаться соответствующих стандартов или соглашений, или положений законов, или подобных рекомендаций;
- защищенность — атрибуты ПС, относящиеся к его способности предотвращать несанкционированный доступ, случайный или преднамеренный, к программам и данным.

Характеристика “Эффективность”:

- характер изменения во времени — атрибуты ПС, относящиеся к временам отклика и обработки и к скорости выполнения его функций;
- характер изменения ресурсов — атрибуты ПС, относящиеся к объему используемых ресурсов и продолжительности такого использования при выполнении функций.

Приведенные комплексные показатели в свою очередь формируются на основе показателей нижележащего уровня. ГОСТ Р ИСО/МЭК 9126-93 не устанавливает этих показателей, так как современное состояние соответствующих моделей, терминов, определений не позволяет включить их в рассматриваемый международный стандарт.

В СССР действовал и продолжает действовать в РФ ГОСТ 28195-89. Этот стандарт устанавливает четырехуровневую модель оценки качества ПС. Характеристики верхних двух уровней (называемые фактор и критерий) устанавливаются в основном тексте документа.

В табл. 1.2. показаны факторы и критерии качества ПС согласно ГОСТ 28195-89.

Таблица 1.2

Наименование факторов и критериев качества ПО и их обозначение	Характеризуемое свойство
1. Надежность ПО (Н)	Характеризует способность ПС в конкретных областях применения выполнять заданные функции в соответствии с программными документами в условиях возникновения отклонений в среде функционирования, вызванных сбоями технических средств, ошибками во входных данных, ошибками обслуживания и другими дестабилизирующими воздействиями
1.1. Устойчивость функционирования (Н1)	Способность обеспечивать продолжение работы ПС после возникновения отклонений, вызванных сбоями технических средств, ошибками во входных данных и ошибками обслуживания
1.2. Работоспособность (Н2)	Способность ПС функционировать в заданных режимах и объемах обрабатываемой информации в соответствии с программными документами при отсутствии сбоев технических средств
2. Сопровождаемость (С)	Характеризует технологические аспекты, обеспечивающие простоту устранения ошибок в ПС и программных документах и поддержания ПС в актуальном состоянии
2.1. Структурность (С1)	Организация всех взаимосвязанных частей ПС в единое целое с использованием логических структур "последовательность", "выбор", "повторение"
2.2. Простота конструкции (С2)	Построение модульной структуры ПС наиболее рациональным образом с точки зрения восприятия и понимания
2.3. Наглядность (С3)	Наличие и представление в наиболее легко воспринимаемом виде исходных модулей ПС, полное их описание в соответствующих программных документах
2.4. Повторяемость (С4)	Степень использования типовых проектных решений или компонентов, входящих в ПС
3. Удобство применения (У)	Характеризует свойства ПС, способствующие быстрому освоению, применению и эксплуатации ПС с минимальными трудозатратами с учетом характера решаемых задач и требований к квалификации обслуживающего персонала

1	2
3.1. Легкость освоения (У1)	Представление программных документов и ПС в виде, способствующем пониманию логики функционирования ПС в целом и его частей
3.2. Доступность эксплуатационных программных документов (У2)	Понятность, наглядность и полнота описания взаимодействия пользователя с ПС в эксплуатационных программных документах
3.3. Удобство эксплуатации и обслуживания (У3)	Соответствие процесса обработки данных и форм представления результатов характеру решаемых задач
4. Эффективность (Э)	Характеризует степень удовлетворения потребности пользователя в обработке данных с учетом экономических, вычислительных и людских ресурсов
4.1. Уровень автоматизации (Э1)	Уровень автоматизации функций процесса обработки данных с учетом рациональности функциональной структуры ПС с точки зрения взаимодействия с ним пользователя и использования вычислительных ресурсов
4.2. Временная эффективность (Э2)	Способность ПС выполнять заданные действия в интервал времени, отвечающий заданным требованиям
4.3. Ресурсоемкость (Э3)	Минимально необходимые вычислительные ресурсы и число обслуживающего персонала для эксплуатации ПС
5. Универсальность (Г)	Характеризует адаптируемость ПС к новым функциональным требованиям, возникающим вследствие изменения области применения или других условий функционирования
5.1. Гибкость (Г1)	Возможность использования ПС в различных областях применения
5.2. Мобильность (Г2)	Возможность применения ПС без существенных дополнительных трудозатрат на ЭВМ аналогичного класса
5.3. Модифицируемость (Г3)	Обеспечение простоты внесения необходимых изменений и доработок в ПС в процессе эксплуатации

1	2
6. Корректность (К)	Характеризует степень соответствия ПС требованиям, установленным в техническом задании, требованиям к обработке данных и общесистемным требованиям
6.1. Полнота реализации (К1)	Полнота реализации заданных функций ПС и достаточность их описания в программной документации
6.2. Согласованность (К2)	Однозначное, непротиворечивое описание и использование тождественных объектов, функций, терминов, определений, идентификаторов и т.д. в различных частях программных документов и текста программы
6.3. Логическая корректность (К3)	Функциональное и программное соответствие процесса обработки данных при выполнении задания общесистемным требованиям
6.4. Проверенность (К4)	Полнота проверки возможных маршрутов выполнения программы в процессе тестирования

Характеристики двух нижних уровней (называемых метрика и оценочный элемент) устанавливаются в справочном Прил. 2 к рассматриваемому стандарту. В том же приложении установлены методы проведения контроля за качеством ПС. Методы определения качества ПС различаются:

- по способу получения информации о ПС – измерительный, регистрационный, органолептический, расчетный;
- по источникам получения информации – традиционный, экспертный, социологический.

Программно-аппаратные средства проведения контроля зависят от вида конкретного ПС и должны разрабатываться отдельно.

В качестве примера приведем некоторые метрики следующих критериев фактора “Надежность”:

- устойчивость функционирования:
 - а) средства восстановления при ошибках на входе;
 - б) средства восстановления при сбоях оборудования;
 - в) реализация управления средствами восстановления;
- работоспособность:
 - а) функционирование в заданных режимах;
 - б) обеспечение обработки заданного объема информации.

В табл. 1.3. приведены некоторые оценочные элементы факторов “Сопровождаемость” и “Корректность”.

Таблица 1.3

Код элемента	Наименование	Метод оценки	Оценка
C0803 ¹	Наличие комментариев в точках входа и выхода программы	Экспертный	0–1
C0302	Оценка простоты программы по числу точек входа и выхода	Расчетный	$W = 1 / ((D+1)(F+1))^2$, где D – общее число точек входа в программу; F – общее число точек выхода из программы
C1002	Оценка простоты программы по числу переходов по условию	То же	$U = (1 - A/B)$, где A – общее число переходов по условию; B – общее число исполняемых операторов
C0303	Осуществляется ли передача результатов работы модуля через вызывающий его модуль	Экспертный	0–1
C0604	Оценка программы по числу циклов	То же	0–1
C0901	Соответствие комментариев принятым соглашениям	– // –	0–1
C1001	Используется ли язык высокого уровня	– // –	0–1
K0101	Наличие всех необходимых документов для понимания и использования ПС	– // –	0–1
K0103	Наличие описания основных функций	– // –	0–1
K0201	Реализация всех основных функций	– // –	0–1
K0701	Комплектность документации в соответствии со стандартами	– // –	0–1
K1003	Отношение числа модулей, отработавших в процессе тестирования и отладки Q_T^M к общему числу модулей Q_O^M	Расчетный	$(Q_T^M) / (Q_O^M)$

¹ Коды оценочных элементов составлены из пяти символов следующим образом: 1-й символ – буква русского алфавита указывает на принадлежность элемента к тому или иному фактору; 2- и 3-й символы – номер метрики, которой принадлежит оценочный элемент; 4- и 5-й – порядковый номер данного оценочного элемента в метрике.

Оценка качества ПС производится в определенной последовательности. На начальных этапах разработки ПС производится выбор показателей и их базовых значений. Для показателей качества на всех четырех уровнях принимается единая шкала оценки от 0 до 1. Показатели качества на вышестоящем уровне (кроме уровня оценочных элементов) определяются показателями качества нижестоящего уровня. В процессе оценки качества ПС на каждом уровне (кроме уровня оценочных элементов) проводятся вычисления показателей качества ПС, т. е. определение количественных значений абсолютных показателей (P_{ij} , где j – порядковый номер показателя для i -го показателя вышестоящего уровня) и относительных показателей K_{ij} , являющихся функцией показателя P_{ij} и базового значения $P_{ij}^{\text{баз}}$.

Критерий и метрика характеризуются двумя числовыми параметрами – количественным значением и весовыми коэффициентами V_{ij} . Сумма весовых коэффициентов показателей уровня l , относящихся к i -му показателю вышестоящего уровня $l-1$, есть величина постоянная. Сумма весовых коэффициентов V_{ij} принимается равной единице.

Общая оценка качества ПС в целом формируется экспертами по набору полученных значений оценок факторов качества. Для оценки качества ПС различного назначения методом экспертного опроса составляется таблица базовых показателей качества ПС.

При сопоставлении характеристик стандартов ГОСТ Р ИСО/МЭК 9126-93 и ГОСТ 28195-89 следует учесть несоответствие используемой терминологии. Например, подхарактеристика 1.4 ГОСТ Р ИСО/МЭК 9126-93 называется, как критерий 6.2 ГОСТ 28195-89 – “согласованность”. Однако в первом документе имеется в виду согласованность ПС со стандартами и другими нормативными документами, а во втором – “однозначное, непротиворечивое описание и использование тождественных объектов, функций, терминов, определений, идентификаторов и т. д. в различных частях программных документов и текста программы”. Точное сопоставление характеристик и подхарактеристик первого из обсуждаемых стандартов с факторами и критериями второго оказывается, как правило, невозможным.

Таким образом, на сегодняшний день действуют два нормативных документа по оценке качества ПС – ГОСТ 28195-89 и ГОСТ Р ИСО/МЭК 9126-93. Первый содержит рекомендуемый (и возможно – неполный) перечень метрик, однако ориентирован на представление разработчика. Второй – более соответствует взглядам пользователя, но для его применения требуется разработка модели качества ПС, включающая метрики и методы оценивания и ранжирования с указанием применимости на стадиях ЖЦ продукта.

1.5. Модели и метрики оценки качества программного средства

Современная программная индустрия за полвека исканий накопила внушительную коллекцию моделей и метрик, оценивающих отдельные производственные и эксплуатационные свойства ПС. Однако погоня за их универсальностью, неучет области применения разрабатываемого ПС, игнорирование этапов жизненного цикла и, наконец, необоснованное их использование в разноплановых процедурах принятия производственных решений, существенно подорвало к ним доверие разработчиков и пользователей ПС.

Тем не менее, анализ технологического опыта лидеров производства ПС показывает, насколько дорого обходится несовершенство ненаучного прогноза разрешимости и трудозатрат, сложности программ, негибкость контроля и управления их разработкой и многое другое, указывающее на отсутствие сквозной методической поддержки и приводящее, в конечном итоге, к его несоответствию требованиям пользователя, требуемому стандарту и к последующей болезненной и трудоемкой его переделке. Эти обстоятельства требуют тщательного отбора методик, моделей, методов оценки качества ПС, учета ограничений их пригодности для различных жизненных циклах и в пределах жизненного цикла, установления порядка их совместного использования, применения избыточного разномодельного исследования одних и тех же показателей для повышения достоверности текущих оценок, накопления и интеграции разнородной метрической информации для принятия своевременных производственных решений и заключительной сертификации продукции.

В табл. 1.4 приведены модели и метрики, хорошо зарекомендовавшие себя при оценке качества ПС, пригодные для прогнозирования и констатации различных показателей сложности и надежности программ.

В таблице представлены разнообразные метрики сложности ПС для различных форм их представления, модели, прогнозирующие ход развития процессов разработки системы, и вероятностные модели по оценке надежности.

Кратко рассмотрим метрики сложности. Одной из основных целей научно-технической поддержки является уменьшение сложности ПС. Именно это позволяет снизить трудоемкость проектирования, разработки, испытаний и сопровождения, обеспечить простоту и надежность программной системы. Целенаправленное снижение сложности системы представляет собой многошаговую процедуру и требует предварительного исследования существующих показателей

сложности, проведения их классификации и соотнесения с типами программ и их местоположением в жизненном цикле.

Таблица 1.4

Название модели	Формула, обозначение
1	2
Метрики сложности	
Метрики Холстеда	
- длина программы;	$N = n_1 \log_1 n_1 + n_2 \log_2 n_2$
- объем программы;	$V = N \log_2 n$
- оценка ее реализации;	$L^* = (2 n_2) / (n_1 N_2)$
- трудность ее понимания;	$E_c = V / L^*$
- трудоемкость кодирования;	$D = (n_1 N_2) (2 n_2) = 1 / L^*$
- уровень языка выражения;	$I^* = V / D^2 = V / L^{*2}$
- информационное содержание;	$I = V / D$
- оптимальная модульность	$M = n_2^* / 6$
Метрики Джилба	
- количество операторов цикла;	L_{loop}
- количество операторов условия;	L_{IF}
- число модулей или подсистем;	L_{mod}
- отношение числа связей между модулями к числу модулей;	$f = N_{SV}^4 / L_{mod}$
- отношение числа ненормальных выходов из множества операторов к общему числу операторов	$f^* = N_{SV}^* / L$
Метрики Мак-Кейба	
- цикломатическое число;	$\lambda(G) = m - n + p$
- цикломатическая сложность	$v(G) = \lambda(G) + 1 = m - n + 2$
Метрика Чепена	
- мера трудности понимания программ на основе входных и выходных данных	$H = 0,5T + P + 2M + 3C$
Метрика Шнадевида	
- число путей в управляющем графе	$S = \sum P_i C_i$
Метрика Майерса	
- интервальная мера.	$[v_1 - v_2]$
Метрика Хансена	
- пара (цикломатическое число, число операторов)	$\{v, N\}$
Метрика Чена	
- топологическая мера Чена	$M(G) = (v(G), \tilde{N}, Q_0)$
Метрика Вудворда	
- узловая мера	$\Psi \xi$
Метрика Кулика	

1	2
- нормальное число (число простейших циклов в нормальной схеме программы). Метрика Хура	$N_{\text{орм}}(P)$
- цикломатическое число сети Петри, отражающей управляющую структуру программы	$\lambda(G_p^*)$
Метрики Витворфа, Зулевского	
- мера сложности потока управления;	$\gamma(P)$
- мера сложности потока данных	$\Omega(P)$
Метрика Петерсона	
- число многоходовых циклов	$N_{\mu_{100p}}$
Метрики Харрисона, Мэйджела	
- функциональное число (сумма приведенных сложностей всех вершин управляющего графа);	$f_1 = \sum \chi_1$
- функциональное отношение (отношение числа вершин графа к функциональному числу);	$f^* = N \chi_1 / f_1$
- регулярные выражения (число операторов, операторов и скобок в регулярном выражении управляющего графа программы)	$\rho(G) = N + L + S_k$
Метрика Пивоварского	
- модифицированная цикломатическая мера сложности	$N(G) = v^*(G) + \sum P_i$
Метрика Пратта	
- тестирующая мера	$\text{Test}(P_r)$
Метрика Кантоне	
- характеристические числа полиномов, описывающих управляющий граф программы	PCN^*
Метрика Мак-Клура	
- мера сложности, основанная на числе возможных путей выполнения программы, числе управляющих конструкций и переменных	$C(V) = D(V) J(V) / N$
Метрика Кафура	
- мера на основе концепции информационных потоков	$I(G)$
Метрика Схуттса, Моханти	
- энтропийные меры	$\varepsilon(G)$

1	2
Метрика Коллофело	
- мера логической стабильности программ	$h(G)$
Метрика Зольновского, Симмонса, Тейера	
Взвешенная сумма различных индикаторов:	
- структура, взаимодействие, объем, данные;	$\sum (\alpha, \beta, \gamma, \nu)$
- сложность интерфейса, вычислительная сложность, сложность ввода/вывода, читабельность	$\sum (\chi, X, \upsilon, \pi)$
Метрика Берлингера	
- информационная мера	$I(R) = \mu (F^*(R) F(R))^2$
Метрика Шумана	
- сложность с позиции статистической теории языка	$\Xi (\Psi)$
Метрика Янгера	
- логическая сложность с учетом истории вычислений	$\Lambda (\omega)$
Сложность проектирования	$C_c = \sum \log_2 (i + 1) [\sum_n C_{xy} (n)]$
Насыщенность комментариями	$X = K/C$
Число внешних обращений	C_i
Число операторов	L_1
Прогноз модели	
Модели Холстеда	
- прогноз системных ресурсов;	$P=3/8 (R_a - 1) 2^{Ra}$
- прогноз числа ошибок	$B = N \log_2 n / 3000$
Модель фирмы IBM	$B = 23M_1 + M_{10}$
Модель общей сложности	$B = 21,1 + 0,1 V + \text{COMP} (S)$
Модели связности	$P_{ij} = 0,15 (S_i + S_j) + 0,7 C_{ij}$
Сплайн-модель	$P_{ij} = 1/2 \sum \lambda_i (\Delta Z_{ij}^2 + \Delta N_{ij}^2) \times$ $\times \ln (\Delta Z_{ij}^2 + \Delta N_{ij}^2) + \alpha + \beta Z_i + \gamma N_1$
Оценочные модели	
Джелиински — Моранды	$R(t) = e^{-(T-1+1) \Phi t}$
Вейса-Байеса	$R_1(t) = \int e^{-\lambda - (i-1)\Phi t} \psi(\lambda, \Phi/t_1, \dots, t_{i-1}) d\lambda d\Phi$
Шика-Волвертона	$R_1(t) = e^{-\Phi(N-1+1) t_i^2/2}$
Литтлвуда	$R_1(t) = (\beta + \tau / \beta + \tau + \tau)^{(N-i+1) \alpha}$
Нельсона	$R_j(t) = \exp \{ \sum \ln (1 - P_j) \}$
Халецкого	$R_j(t) = P_\infty - a(1 - \gamma^{nj}) / n_j$
Модель отлаженности	$R_j(t) = P_\infty - \rho f_j(\tau, \lambda, \pi)$
Мозаичная модель	$R_j(t) = 1 - \beta(\alpha - \omega_{j-1})$

Теория сложности программ ориентирована на управление качеством ПС и контроль его эталонной сложности в период эксплуатации. В настоящее время многообразие показателей (в той или иной степени описывающих сложность программ) столь велико, что для их употребления требуется предварительное упорядочение. В ряде случаев удовлетворяются тремя категориями метрик сложности. Первая категория определяется как словарная метрика, основанная на метрических соотношениях Холстеда, цикломатических мерах Мак-Кейба и измерениях Тейера. Вторая категория ориентирована на метрики связей, отражающих сложность отношений между компонентами системы – это метрики Уина и Винчестера. Третья категория включает семантические метрики, связанные с архитектурным построением программ и их оформлением.

Согласно другой классификации, показатели сложности делятся на две группы: сложность проектирования и сложность функционирования. Сложность проектирования, которая определяется размерами программы, количеством обрабатываемых переменных, трудоемкостью и длительностью разработки и другими факторами, анализируется на основе трех базовых компонентов: сложность структуры программы, сложность преобразований (алгоритмов), сложность данных. Во вторую группу показателей отнесены временная, программная и информационная сложности, характеризующие эксплуатационные качества ПС.

Существует еще ряд подходов к классификации мер сложности, однако они, фиксируя частные стороны исследуемых программ, не позволяют (пусть с большим допущением) отразить общее, то, чьи замеры могут лечь в основу производственных решений.

Общим, инвариантно присущим любому ПС (и связанному с ним корректностью), является его структура. Важно связать это обстоятельство с определенным значением структурной сложности в совокупности мер сложности ПС. И более того, при анализе структурной сложности целесообразно ограничиться только ее топологическими мерами, т. е. мерами, в основе которых лежат топологические характеристики граф-модели программы. Эти меры удовлетворяют подавляющему большинству требований, предъявляемых к показателям: общности применимости, адекватности рассматриваемому свойству, существенности оценки, состоятельности, количественному выражению, воспроизводимости измерений, малой трудоемкости вычислений, возможности автоматизации оценивания.

Именно топологические меры сложности наиболее часто применяются в фазе исследований, формирующей решения по управлению

производством (в процессах проектирования, разработки и испытаний) и составляют доступный и чувствительный эталон готовой продукции, контроль которого необходимо регулярно осуществлять в период ее эксплуатации.

Первой топологической мерой сложности является цикломатическая мера Мак-Кейба. В ее основе лежит идея оценки сложности ПС по числу базисных путей в его управляющем графе, т. е. таких путей, komponуя которые можно получить всевозможные пути из входа графа в выходы. Цикломатическое число $\lambda(G)$ орграфа G с n -вершинами, m -дугами и p -компонентами связности есть величина $\lambda(G) = m - n + p$.

Имеет место теорема о том, что число базисных путей в орграфе равно его цикломатическому числу, увеличенному на единицу. При этом цикломатической сложностью ПС P с управляющим графом G называется величина $v(G) = \lambda(G) + 1 = m - n + 2$. Практически цикломатическая сложность ПС равна числу предикатов плюс единица, что позволяет вычислять ее без построения управляющего графа простым подсчетом предикатов. Данная мера отражает “психологическую” сложность ПС.

К достоинствам меры относят простоту ее вычисления и повторяемость результата, а также наглядность и содержательность интерпретации. В качестве недостатков можно отметить: нечувствительность к размеру ПС, нечувствительность к изменению его структуры, отсутствие корреляции со структурированностью ПС, отсутствие различия между конструкциями *Развилка* и *Цикл*, отсутствие чувствительности к вложенности циклов. Недостатки цикломатической меры привели к появлению ее модификаций, а также принципиально иных мер сложности.

Дж. Майерс предложил в качестве меры сложности интервал $[v_1 - v_2]$, где v_1 — цикломатическая мера, а v_2 — число отдельных условий плюс единица. При этом оператор *DO* считается за одно условие, а *CASE* с n -исходами за $n-1$ -условий. Введенная мера получила название интервальной мерой.

У. Хансену принадлежит идея брать в качестве меры сложности ПС пару (цикломатическое число, число операторов). Известна топологическая мера $Z(G)$, чувствительная к структурированности ПС. При этом она $Z(G) = V(G)$ (равна цикломатической сложности) для структурированных программ и $Z(G) > V(G)$ для неструктурированных. К вариантам цикломатической меры сложности относят также меру $M(G) = (V(G), C, Q)$, где C — количество условий, необходимых для покрытия управляющего графа минимальным числом маршру-

тов, а Q — степень связности структуры графа программы и ее протяженность.

К мерам сложности, учитывающим вложенность управляющих конструкций, относят тестирующую меру M и меру Харрисона-Мейджела, которые учитывают уровень вложенности и протяженности ПС, меру Пивоварского — цикломатическую сложность и глубину вложенности, и меру Мак-Клура — сложность схемы разбиения ПС на модули с учетом вложенности модулей и их внутренней сложности.

Функциональная мера сложности Харрисона-Мейджела предусматривает приписывание каждой вершине графа своей собственной сложности (первичной) и разбиение графа на “сферы влияния” предикатных вершин. Сложность сферы называют приведенной и складывают ее из первичных сложностей вершин, входящих в сферу ее влияния, плюс первичную сложность самой предикатной вершины. Первичные сложности вычисляются всеми возможными способами. Отсюда функциональная мера сложности ПС есть сумма приведенных сложностей всех вершин управляющего графа.

Мера Пивоварского ставит целью учесть в оценке сложности ПС различия не только между последовательными и вложенными управляющими конструкциями, но и между структурированными и неструктурированными программами. Она выражается отношением $N(G) = v^*(G) + \sum P_i$, где $v^*(G)$ — модифицированная цикломатическая сложность, вычисленная так же, как и $V(G)$, но с одним отличием: оператор *CASE* с n -выходами рассматривается как один логический оператор, а не как $n-1$ оператор, P_i — глубина вложенности i -й предикатной вершины.

Для подсчета глубины вложенности предикатных вершин используется число “сфер влияния”. Под глубиной вложенности понимается число всех “сфер влияния” предикатов, которые либо полностью содержатся в сфере рассматриваемой вершины, либо пересекаются с ней. Глубина вложенности увеличивается за счет вложенности не самих предикатов, а “сфер влияния”. Сравнительный анализ цикломатических и функциональных мер с обсуждаемой для десятка различных управляющих графов программы показывает, что при нечувствительности прочих мер этого класса, мера Пивоварского возрастает при переходе от последовательных программ к вложенным и далее к неструктурированным.

Мера Мак-Клура предназначена для управления сложностью структурированных программ в процессе проектирования. Она применяется к иерархическим схемам разбиения программ на модули, что позволяет выбрать схему разбиения с меньшей сложностью задолго

до написания программы. Метрикой выступает зависимость сложности программы от числа возможных путей исполнения, числа управляющих конструкций и числа переменных (от которых зависит выбор пути). Методика расчета сложности по Мак-Клуру четко ориентирована на хорошо структурированные программы.

Тестирующей мерой M называется мера сложности, удовлетворяющая следующим условиям:

- мера сложности простого оператора равна единице;
- $M(\{F_1; F_2; \dots; F_n\}) = \sum_i^n M(F_i)$;
- $M(IF\ P\ THEN\ F_1\ ELSE\ F_2) = 2\ MAX(M(F_1), M(F_2))$;
- $M(WHILE\ P\ DO\ F) = 2\ M(F)$.

Мера возрастает с глубиной вложенности и учитывает протяженность программы. К тестирующей мере близко примыкает мера на основе регулярных вложений. Идея этой меры сложности программ состоит в подсчете суммарного числа символов (операндов, операторов, скобок) в регулярном выражении с минимально необходимым числом скобок, описывающим управляющий граф программы. Все меры этой группы чувствительны к вложенности управляющих конструкций и к протяженности программы. Однако возрастает уровень трудоемкости вычислений.

Рассмотрим меры сложности, учитывающие характер разветвлений. В основе узловой меры Вудворда–Хедли лежит идея подсчета топологических характеристик потока управления. При этом под узловой сложностью понимается число узлов передач управления. Данная мера отслеживает сложность линеаризации программы и чувствительна к структуризации (сложность уменьшается). Она применима для сравнения эквивалентных программ, предпочтительнее меры Холстеда, но по общности уступает мере Мак-Кейба.

Топологическая мера Чена выражает сложность программы числа пересечений границ между областями, образуемыми блок-схемой программы. Этот подход применим только к структурированным программам, допускающим лишь последовательное соединение управляющих конструкций. Для неструктурированных программ мера Чена существенно зависит от условных и безусловных переходов. В этом случае можно указать верхнюю и нижнюю границы меры. Верхняя – есть $m + 1$, где m – число логических операторов при их гнездовой вложенности. Нижняя – равна 2. Когда управляющий граф программы имеет только одну компоненту связности, мера Чена совпадает с цикломатической мерой Мак-Кейба.

Метрики Джилба оценивают сложность графоориентированных модулей программ отношением числа переходов по условию к обще-

му числу исполняемых операторов. Хорошо зарекомендовала себя метрика, которая относит число межмодульных связей к общему числу модулей. Названные метрики использовались для оценки сложности эквивалентных схем программ, в особенности схем Янова.

Используются также меры сложности, учитывающие историю вычислений, характер взаимодействия модулей и комплексные меры.

Совокупность цикломатических мер пригодна для оценивания сложности первичных формализованных спецификаций, задающих в совокупности исходные данные, цели и условия построения искомого ПС. Оценка этой “первичной” программы или сравнение нескольких альтернативных ее вариантов позволит изначально гармонизировать процесс разработки ПС и от стартовой точки контролировать и управлять его текущей результирующей сложностью.

Коды оценочных элементов составлены из пяти символов следующим образом:

1-й символ — буква русского алфавита — указывает на принадлежность элемента к тому или иному фактору;

2- и 3-й символы — номера метрики, которой принадлежит оценочный элемент;

4- и 5-й — порядковые номера данного оценочного элемента в метрике.

ГЛАВА 2

СТАНДАРТЫ, ОПРЕДЕЛЯЮЩИЕ ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНЫХ СРЕДСТВ

2.1. Модели жизненного цикла программных средств

Согласно источникам [1,4], в настоящее время просматривается тенденция в сторону увеличения объема работ, связанных с разработкой программного обеспечения по сравнению с работами, выполнение которых позволит получить аппаратные средства ЭВМ. Таким образом, имеет смысл более подробно рассматривать технологический процесс разработки программных средств, жизненный цикл (ЖЦ) программного обеспечения, реализующих специальное математическое, информационное, программное, лингвистическое обеспечение ЭВМ.

В основе деятельности по созданию и использованию программных средств лежит понятие жизненного цикла. В общем случае различают понятия жизненного цикла программного средства и технологического процесса разработки. Более четко различия между данными понятиями просматриваются в отношении программных средств. Жизненный цикл является моделью создания и использования программного обеспечения, отражающей его различные состояния, начиная с момента возникновения необходимости в программном средстве и заканчивая моментом его полного выхода из употребления у пользователей.

Существуют несколько моделей жизненного цикла. Традиционно выделяют следующие основные этапы жизненного цикла (рис. 2.1):

- стратегическое планирование;
- анализ требований;
- проектирование (предварительное и детальное);
- кодирование (программирование);
- тестирование и отладка;
- эксплуатация и сопровождение.

Каждому этапу соответствуют определенный результат и набор документации, являющиеся исходными данными для следующего этапа. В заключение каждого этапа производится верификация доку-

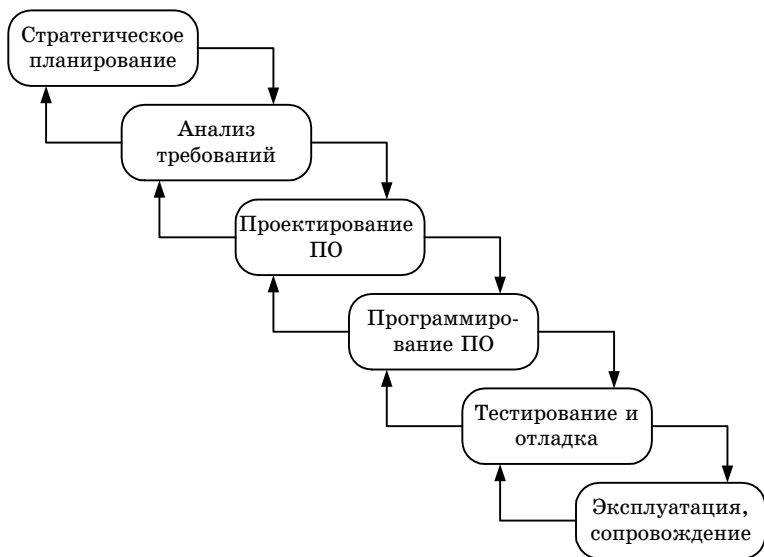


Рис. 2.1. Этапы ЖЦ программного средства

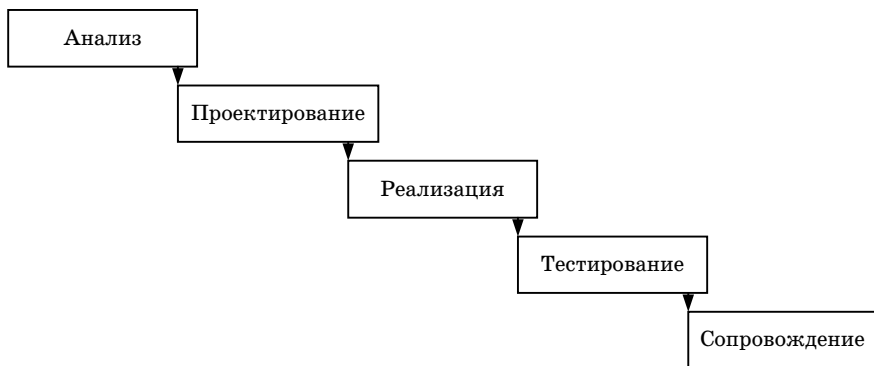


Рис. 2.2. Каскадная модель жизненного цикла

ментов и решений с целью проверки их соответствия первоначальным требованиям заказчика.

Исторически, в ходе эволюционного развития теории проектирования программного обеспечения и по мере его усложнения, сложились три основные модели жизненного цикла [5]. Эти модели выражают последовательность этапов ЖЦ ПС. До 80-х годов имела место каскадная модель ЖЦ (рис. 2.2), называемая моделью “водопада”, подразумевавшая переход на последующие этапы ЖЦ только после полного окончания работ на предыдущих этапах.

Модель предполагает следующие свойства взаимодействия этапов:

- модель состоит из последовательно расположенных этапов;
- каждый этап полностью заканчивается до того, как начнется следующий;
- этапы не перекрываются во времени, т. е. следующий этап не начинается пока не завершится предыдущий;
- возврат к предыдущим этапам не предусматривается или крайне ограничен;
- результат появляется только в конце разработки.

Рассмотрим подробнее отдельные этапы жизненного цикла.

Анализ и спецификация требований. На этапе анализа и спецификации требований определяются самые общие требования к программному средству. Данный этап предполагает решение следующих задач:

- определение целесообразности разработки и сравнение с аналогами;
- определение необходимых ресурсов для решения задачи;
- спецификация требований к средству в виде “что оно должно делать”, но не в виде “как это реализовать”;
- проверка корректности и реализуемости требований.

Проектирование. На этапе проектирования создается структура будущего программного средства. Можно определить следующие фазы проектирования:

- проектирование архитектуры, включает в себя определение состава подсистем;
- спецификация подсистем, определяет спецификацию каждой подсистемы;
- проектирование интерфейса, определяет интерфейс каждой подсистемы, т. е. метод взаимодействия данной подсистемы с другими;
- проектирование компонентов, каждая подсистема разделяется на компоненты;
- проектирование структур данных, определяет где и как хранятся данные;
- проектирование алгоритмов, определяются алгоритмы обработки данных.

Реализация. На этапе реализации выбирается язык программирования, составляется текст программы (кодирование) и производится отладка. Возможно выполнение тестирования отдельных фрагментов.

Тестирование. На этом этапе выполняется комплексное тестирования всего программного средства специальной группой.

Сопровождение. На этом этапе программное средство сдается в эксплуатацию, происходит обслуживание пользователей, возможно

устранение незначительных ошибок (сейчас это делается повсеместно с помощью распространения так называемых patch-файлов).

На каждом этапе составляется проектная документация, соответствующая его задачам.

С развитием вычислительной техники, в середине 80-х, сложность и объемы программных средств существенно возросли. В связи с этим возникли проблемы с разработкой и отладкой ПС. Продумать все шаги разработки нового ПС, наметить этапы проектирования и предусмотреть все варианты поведения при его отладке стало не под силу одному разработчику. Каскадная модель ЖЦ стала существенно сдерживать темпы создания сложных программных средств. Процесс отладки при этом затягивался и не давал гарантий безошибочной работы программ.

На смену каскадной модели, жестко регламентирующей последовательность этапов и критерии переходов между ними, пришла *поэтапная модель с промежуточным контролем*. Это итерационная модель разработки ПС с обратными связями между этапами. Проверки и корректировки разрабатываемого ПС проводятся на каждом из этапов, что позволяет существенно снизить трудоемкость отладки по сравнению с каскадной моделью. Итерационность модели проявляется в обработке ошибок, выявленных промежуточным контролем. Если на каком-либо из этапов в ходе промежуточной проверки обнаружилась ошибка, допущенная на более ранней стадии разработки, работы этапа, повлекшего ошибку, необходимо провести повторно. При этом анализируются причины ошибки и по необходимости корректируются исходные данные этапа или перечень проводимых работ. Аналогичная ситуация возникает, если в ходе разработки ПС возникают новые требования заказчика или изменяются какие-либо условия его функционирования.

Спиральная модель поддерживает итерации поэтапной модели, но особое внимание уделяется начальным этапам проектирования: анализу требований, проектированию спецификаций, предварительному проектированию и детальному проектированию. Каждый виток спирали соответствует поэтапной модели создания фрагмента или версии ПС, уточняются цели и требования к нему, оценивается качество разработанного фрагмента или версии ПС и планируются работы следующего витка. Таким образом, углубляются и конкретизируются все детали проектируемого ПС, и в результате получается вариант, который удовлетворяет всем требованиям заказчика.

Широко известна, благодаря монографии [2], *объектно-ориентированная модель* жизненного цикла.

Фирма Rational Software, разработавшая язык UML, предложила также и свою модель жизненного цикла (рис. 2.3), которая называется *Rational Objectory Process* (данный термин труден для перевода, так как, во-первых, слово Rational имеет значение “рациональный” и название фирмы одновременно, во-вторых, слова objectory нет в английском языке, оно построено по аналогии со словом repository (накопитель)).

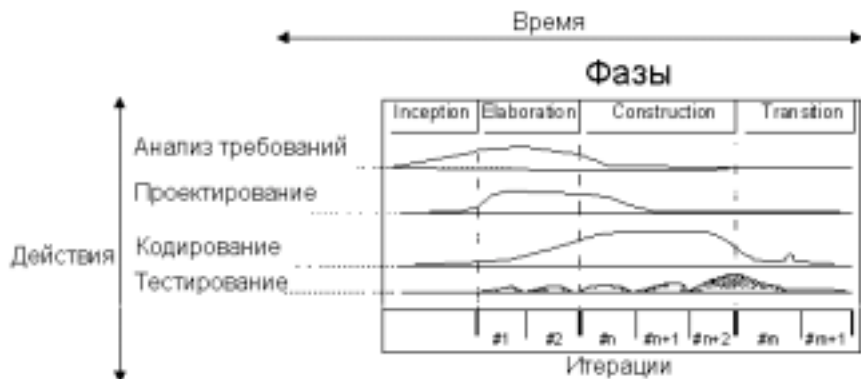


Рис. 2.3. Модель жизненного цикла UML

Можно перечислить следующие основные свойства данной технологии:

- процесс итеративный, т. е. происходит последовательное уточнение результатов;
- действия процесса направлены на создание моделей, а не других элементов проекта, например, текстовых документов;
- действия жизненного цикла определяются в первую очередь блоками использования (use case).

Жизненный цикл разбит на циклы, результатом каждого из которых является собственная версия программной системы. Каждый цикл состоит из четырех фаз:

- начало (Inception);
- совершенствование (Elaboration);
- построение (Construction);
- переход (Transition).

Количество, состав и последовательность этапов ЖЦ для каждого конкретного ПС определяются на ранних стадиях планирования его создания. При этом учитываются особенности эксплуатации, наличие разного рода ограничений, численность и квалификация пер-

сона разработчиков и эксплуатационников, а также множество других факторов.

Как было отмечено выше, жизненные циклы программных средств, процессов их разработки, эксплуатации и сопровождения регламентированы в стандартах. При этом стандарты, разрабатываемые международными организациями в той или иной области деятельности, носят рекомендательный характер и не возведены в ранг закона. Руководящие принципы, определенные в стандартах, имеют официальную силу тогда, когда приняты правительством той или иной страны. В настоящее время общепризнанными международными лидерами в области стандартизации разработки ПС стали следующие организации: Американский национальный институт по стандартизации – ANSI, Международная организация по стандартизации – ISO, Министерство обороны США – DOD, Институт инженеров по электронике и радиотехнике (IEEE).

Далее рассмотрим, какие конкретные жизненные циклы разработки ПС созданы в нашей стране и за рубежом, а также какими стандартами они регламентированы.

2.2. Стадии разработки программных средств, регламентированных ГОСТами

В нашей стране жизненный цикл разработки ПС установлен стандартом ГОСТ 19.102-77 “Стадии разработки программ и программной документации” и содержит следующие стадии и этапы:

- техническое задание (ТЗ);
- эскизный проект (ЭП);
- технический проект (ТП);
- рабочий проект (РП);
- внедрение.

В табл. 2.1 показаны стадии разработки и этапы, их составляющие.

Неверно предполагать, что жизненный цикл разработки ПС, согласно ГОСТ 19.102-77, есть последовательное выполнение стадий и этапов, определенных в табл. 2.1. В реальном жизненном цикле трудно провести четкую и определенную границу между этапами, а сам процесс создания ПС является итеративным: после завершения некоторого этапа почти всегда есть необходимость в коррекции уже выполненных этапов и стадий с целью внесения уточнений. При разработке принципиально нового ПС иногда бывает необходимо осуществить пробную реализацию с целью получения информации, требующейся для принятия решения на некоторой стадии.

Таблица 2.1

Стадии разработки	Этапы работ
Техническое задание	1. Обоснование необходимости разработки программ 2. Выполнение научно-исследовательских работ (НИР) 3. Разработка и утверждение технического задания
Эскизный проект	1. Разработка эскизного проекта 2. Утверждение эскизного проекта
Технический проект	1. Разработка технического проекта 2. Утверждение технического проекта
Рабочий проект	1. Разработка программы 2. Разработка программной документации 3. Испытание программы
Внедрение	1. Подготовка и передача программы

Специалистам в области разработки ПС известно, что наиболее важными стадиями в жизненном цикле разработки являются начальные, так как ошибки, допущенные в них, требуют значительных затрат на исправление на конечных стадиях.

Техническое задание. На стадии “Техническое задание” выполняются следующие работы, входящие в состав соответствующих этапов:

1. Обоснование необходимости разработки программ:

- постановка задачи;
- сбор исходных материалов;
- выбор и обоснование критериев эффективности и качества;
- обоснование необходимости проведения НИР.

2. Выполнение научно-исследовательских работ:

- определение структуры входных и выходных данных;
- предварительный выбор методов решения задач;
- обоснование целесообразности применения ранее разработанных программ;
- определение требований к техническим средствам;
- обоснование принципиальной возможности решения поставленных задач.

3. Разработка и утверждение технического задания:

- определение требований к программе;
- разработка технико-экономического обоснования разработки программы;
- определение стадий, этапов и сроков разработки программы и документации на нее;

- выбор языков программирования;
- определение необходимости проведения НИР на последующих стадиях;
- согласование и утверждение ТЗ.

Результатом выполнения данной стадии является *техническое задание*, оформленное в соответствии с ГОСТ 19.105-78 (изм. 09.1981.) “Общие требования к программным документам и ГОСТ 19.106-78 Общие требования к программным документам, выполненным печатным способом”.

Эскизный проект. Основные этапы и содержание работ на стадии “Эскизный проект” приведены в табл. 2.2.

Таблица 2.2

Этапы работ	Содержание
Разработка ЭП	<ol style="list-style-type: none"> 1. Предварительная разработка структуры входных и выходных данных 2. Уточнение методов решения задач 3. Разработка общего описания алгоритма решения задачи 4. Разработка технико-экономического обоснования
Утверждение ЭП	<ol style="list-style-type: none"> 1. Разработка пояснительной записки 2. Согласование и утверждение эскизного проекта

Конкретное содержание работ стадии эскизного проекта и их объем определяет степень сложности разрабатываемого ПС. Результатом выполнения данной стадии является полное описание архитектуры ПС. Как правило, это описание делается на нескольких уровнях иерархии. На верхнем уровне детализации выделяются основные подсистемы, которым присваиваются имена, устанавливаются связи между подсистемами, их функции, получаемые путем декомпозиции предполагаемых функций ПС. Затем процедура декомпозиции выполняется для каждой подсистемы, выделяются модули, составляющие данную подсистему. В конечном итоге, получается иерархически организованная система, состоящая из уровней, каждый из которых представляет собой совокупность взаимосвязанных модулей.

Единицы, выделяемые на различных иерархических уровнях функциональной архитектуры системы, определяются по усмотрению разработчика. Стандарты Единой системы программной документации (ЕСПД) различают программные единицы только с точки зрения их документирования.

Результаты эскизного проекта отображаются в документе “Пояснительная записка к эскизному проекту”, оформленному в соответ-

ствии с ГОСТ 19.105-78 и ГОСТ 19.404-79 “Пояснительная записка”.

После утверждения пояснительной записки она становится программным документом, правила дублирования, учета, хранения которого определяются ГОСТ 19.601-78 Общие правила дублирования, обращения, учета и хранения и ГОСТ 19.602-78 “Правила дублирования, учета и хранения программных документов, выполненных печатным способом”. Последующие стадии и этапы разработки ПС могут выявить необходимость внесения изменений в эскизный проект. Эти изменения должны быть отражены в пояснительной записке в соответствии с ГОСТ 19.603-78 “Общие правила внесения изменений в программные документы и ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом”.

В качестве примера ниже приводится фрагмент расширенного описания работ стадии эскизного проекта:

- разработка плана совместных работ на разработку ПС;
- разработка и обоснование математической модели системы на ЭВМ и описание результатов моделирования;
- разработка и обоснование алгоритмов и временных графиков функционирования ПС по всем режимам работы;
- разработка и обоснование ресурсов памяти для реализации алгоритмов;
- разработка перечня документов на ПС;
- разработка и обоснование структуры БД, внешних входных и выходных данных;
- разработка и обоснование алгоритмов информационного обеспечения;
- определение взаимосвязей между видами программ;
- разработка и обоснование набора тестов для проверки ПС;
- разработка и обоснование организации наращивания и развития ПС;
- оформление пояснительной записки и ведомости эскизного проекта ПС (в соответствии с ГОСТ 19.105-78, ГОСТ 19.404 -79 и ГОСТ 2.106-68 “ЕСКД. Текстовые документы”);
- согласование и утверждение ЭП.

Технический проект. Основные этапы и содержание работ на стадии “Технический проект” приведены в табл. 2.3.

Содержанием работ на этой стадии является проектирование структуры ПС. Результат – реализующий заданный и утвержденный в техническом задании комплекс программ в качестве иерархи-

ческой структуры программных модулей, заданных своими функциональными спецификациями. Форма представления результата – “Пояснительная записка к техническому проекту” согласно ГОСТ 19.105-78, ГОСТ 19.404-79.

Разработка структуры ПС заключается в выделении всех программных компонентов по функциональным признакам, определении функциональных спецификаций модулей и уточнении внешних функциональных спецификаций, структуре входных и выходных данных, определении операционной среды, языковых средств, конфигурации аппаратных средств.

Спецификации модулей являются внешними характеристиками и содержат все сведения, необходимые вызывающим модулям. На последующих стадиях разработки спецификации оформляются в виде комментариев в начале текста исходной программы модуля. На данной стадии спецификации оформляются в виде комментария на принятом в организации, занимающейся разработкой ПС, языке спецификаций.

Таблица 2.3

Этапы работ	Содержание
Разработка ТП	<ol style="list-style-type: none"> 1. Уточнение структуры входных и выходных данных 2. Разработка алгоритмов решения задач 3. Определение формы представления входных и выходных данных 4. Определение синтаксиса и семантики языка 5. Разработка структуры программы 6. Окончательное определение конфигурации технических средств
Утверждение ТП	<ol style="list-style-type: none"> 1. Разработка плана мероприятий по разработке и внедрению программ 2. Разработка пояснительной записки 3. Согласование и утверждение ТП

Фрагмент описания работ стадии технического проекта приведен ниже:

- разработка плана совместных работ на разработку ПС;
- распараллеливание задач применительно к возможностям ЭВМ и систем обмена информации, и разработка организационной структуры программ;
- формализация программно-аппаратных и внутрizaдачных интерфейсов;
- разработка алгоритмов управления вычислительным процессом и процессами обмена, его исследование и оптимизация;

- разработка и комплексное решение вопросов обеспечения устойчивой работы на программном уровне;
- разработка модели и исследование на ней возможности реализации всех функций системы управления в рамках имеющихся вычислительных ресурсов на основе выбранного управляющего алгоритма и оптимизация структуры программ до получения удовлетворяющего результата;
- уточнение структуры и определение формы представления БД, внешних входных и выходных данных;
- разработка проекта описания программы для каждого вида программ ПС;
- уточнение алгоритмов ПС и алгоритмов информационного обеспечения в процессе решения задач;
- разработка проекта описания текстовых программ ПС;
- разработка плана реализации ПС;
- разработка проектов технических условий на ПС;
- разработка и обоснование решений по надежности функционирования ПС;
- разработка программы и правил испытаний ПС;
- уточнение временных графиков функционирования программ ПС в процессе решения задач;
- оформление пояснительной записки и ведомости технического проекта ПС (в соответствии с ГОСТ 19.105-78, ГОСТ 19.404-79 и ГОСТ 2.106-68 “ЕСКД. Текстовые документы”);
- согласование и утверждение ТП.

Рабочий проект. Основные этапы и содержание работ на стадии “Рабочий проект” приведены в табл. 2.4.

Таблица 2.4

Этапы работ	Содержание
Разработка ПС	Программирование и отладка программ
Разработка программной документации	1. Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77
Испытание ПС	1. Разработка, согласование и утверждение программ и методики испытаний 2. Проведение предварительных государственных, межведомственных приемо-сдаточных и других видов испытаний 3. Корректировка ПС и программной документации по результатам испытаний

Содержанием работ на этой стадии является описание ПС на выбранном проблемно-ориентированном языке (кодирование), отладка, разработка, согласование и утверждение порядка и методики испытаний, разработка программных документов, проведение тестирования, корректировка программ и программной документации по результатам тестирования, проведение приемо-сдаточных испытаний. Результат – ПС в форме программной документации, в форме документации на ПС или в форме ПС.

На стадии “*Внедрение*” осуществляется подготовка и передача ПС и программной документации для сопровождения и/или изготовления, оформление и утверждение акта о передаче ПС на сопровождение или изготовление, передача ПС в фонд алгоритмов и программ.

Кроме рассмотренного выше жизненного цикла программ, установленного 19-й системой стандартов, необходимо иметь в виду, что существует жизненный цикл автоматизированных систем (АС), установленный ГОСТ 34.601-90 “ИТ. Автоматизированные системы. Стадии создания”. Имеет смысл рассмотреть стадии и этапы разработки АС, так как 19-я система стандартов в настоящее время морально устарела и более современным представляется жизненный цикл АС. Анализ жизненных циклов разработки ПС и АС, установленных ГОСТ 19.102-77 и ГОСТ 34.601-90, а также международных стандартов, регламентирующих данный процесс, будет приведен ниже.

Стандарт ГОСТ 34.601-90 распространяется на автоматизированные системы, представляющие собой организационно-технические системы, обеспечивающие выработку решений на основе автоматизации информационных потоков в различных видах деятельности (исследование, проектирование, управление и т.п.), включая их сочетания, создаваемые в организациях, объединениях и на предприятиях. Данный стандарт устанавливает стадии и этапы создания АС.

В процессе разработки АС создают, в общем случае, следующие виды обеспечения: техническое, программное, информационное математическое и др. Проектные решения по программному, техническому и информационному обеспечению реализуют как изделия в виде взаимоувязанной совокупности компонент и комплексов, входящей в состав АС с необходимой документацией. Справочное приложение к ГОСТ 34.601-90 устанавливает, что программное обеспечение АС – совокупность программ на носителях информации с программной документацией по ГОСТ 19.101-77 “Виды программ и программных документов”. Таким образом, при разработке программного обеспечения можно пользоваться как стандартом ГОСТ 19.102-77, так и ГОСТ 34.601-90.

Процесс создания АС представляет собой совокупность упорядоченных во времени, взаимосвязанных, объединенных в стадии и этапы работ, выполнение которых необходимо и достаточно для создания АС, соответствующей заданным требованиям. Стадии и этапы создания АС выделяются как части процесса создания по соображениям рационального планирования и организации работ, заканчивающихся заданным результатом.

Как и в стандарте ГОСТ 19.102-77, работы по развитию АС осуществляются по стадиям и этапам, применяемым для создания АС. Состав и правила выполнения работ на установленных настоящим стандартом стадиях и этапах определяют в соответствующей документации организаций, участвующих в создании конкретных видов АС.

Стадии и этапы создания АС в общем случае приведены в табл. 2.5.

Таблица 2.5

Стадии	Этапы работ
1. Формирование требований к АС	1.1. Обследование объекта и обоснование необходимости создания АС 1.2. Формирование требований пользователя к АС 1.3. Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания)
2. Разработка концепции АС	2.1. Изучение объекта 2.2. Проведение необходимых научно-исследовательских работ 2.3. Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователя 2.4. Оформление отчета о выполненной работе
3. Техническое задание	3.1. Разработка и утверждение технического задания на создание АС
4. Эскизный проект	4.1. Разработка предварительных проектных решений по системе и ее частям 4.2. Разработка документации на АС и ее части
5. Технический проект	5.1. Разработка проектных решений по системе и ее частям 5.2. Разработка документации на АС и ее части 5.3. Разработка и оформление документации на поставку изделий для комплектования АС и/или технических требований (технических заданий) на их разработку 5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации

1	2
6. Рабочая документация	6.1. Разработка рабочей документации на систему и ее части 6.2. Разработка или адаптация программ
7. Ввод в действие	7.1. Подготовка объекта автоматизации к вводу АС в действие 7.2. Подготовка персонала 7.3. Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями) 7.4. Строительно-монтажные работы 7.5. Пусконаладочные работы 7.6. Проведение предварительных испытаний 7.7. Проведение опытной эксплуатации 7.8. Проведение приемочных испытаний
8. Сопровождение АС	8.1. Выполнение работ в соответствии с гарантийными обязательствами 8.2. Послегарантийное обслуживание

Стандартом ГОСТ 34.601-90 допускается исключать стадию “Эскизный проект” и отдельные этапы работ на всех стадиях, объединять стадии “Технический проект” и “Рабочая документация” в одну стадию “Технорабочий проект”. В зависимости от специфики создаваемых АС и условий их создания допускается выполнять отдельные этапы работ до завершения предшествующих стадий, параллельных во времени выполнения этапов работ, включение новых этапов работ.

Рассмотрим этапы, касающиеся разработки программных средств.

На этапе “Обследование объекта и обоснование необходимости создания АС” проводят сбор данных об объекте автоматизации и осуществляемых видах деятельности, оценку качества функционирования объекта и осуществляемых видов деятельности, выявление проблем, решение которых возможно средствами автоматизации, а также оценку (технико-экономической, социальной и т. п.) целесообразности создания АС.

На этапе “Формирование требований пользователя к АС” проводят подготовку исходных данных для формирования требований к АС (характеристика объекта автоматизации, описание требований к системе, ограничения допустимых затрат на разработку, ввод в дей-

ствие и эксплуатацию, эффект, ожидаемый от системы, условия создания и функционирования) и формулировку, оформление требований пользователя к АС.

На этапе “Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания)” проводят оформление отчета о выполненных работах на данной стадии и оформление заявки на разработку АС (тактико-технического задания) или другого заменяющего ее документа с аналогичным содержанием.

На этапах “Изучение объекта” и “Проведение необходимых НИР” организация-разработчик проводит детальное изучение объекта автоматизации и необходимые НИР, связанные с поиском путей и оценкой возможности реализации требований пользователя, оформляют и утверждают отчеты о НИР.

Этап “Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователя” направлен на разработку альтернативных вариантов концепции создаваемой АС и планов реализации, оценку необходимых ресурсов на их реализацию и обеспечение функционирования, оценку преимуществ и недостатков каждого варианта, сопоставление требований пользователя и характеристик предлагаемой системы и выбор оптимального варианта, определение порядка оценки качества и условий приемки системы, а также оценку эффектов, получаемых от системы.

На этапе “Оформление отчета о выполненной работе” готовят и оформляют отчет, содержащий описание выполненных работ на стадии, описание и обоснование предлагаемого варианта концепции системы.

На этапе “Разработка и утверждение технического задания на создание АС” проводят разработку, оформление, согласование и утверждение технического задания на АС и технических заданий на части АС.

Результатом выполнения этапа “Разработка предварительных проектных решений по системе и ее частям” являются: функции АС, функции подсистем, их цели и эффекты, состав комплекса задач и отдельных задач, концепции информационной базы, ее укрупненная структура, функции СУБД, состав вычислительной системы, функции и параметры основных программных средств.

На этапе “Разработка проектных решений по системе и ее частям” обеспечивают разработку общих решений: по системе и ее частям, функционально-алгоритмической структуре системы; по функциям персонала и организационной структуре; по структуре техни-

ческих средств; по алгоритмам решения задач и применяемым языкам; по организации и ведению информационной базы, системе классификации и кодирования информации; по программному обеспечению.

На этапе “Разработка документации на АС и ее части” проводят оформление, согласование и утверждение документации в объеме, необходимом для описания полной совокупности принятых проектных решений и достаточном для дальнейшего выполнения работ по созданию АС.

На этапе “Разработка и оформление документации на поставку изделий для комплектации АС и (или) технических требований (технических заданий) на их разработку” проводят подготовку и оформление документации на поставку изделий для комплектования АС, определение технических требований и составление ТЗ на разработку изделий, не изготавливаемых серийно.

На этапе “Разработка заданий на проектирование в смежных частях проекта объекта автоматизации” осуществляют разработку, оформление, согласование и утверждение заданий на проектирование в смежных частях проекта объекта автоматизации для проведения строительных, электротехнических, санитарно-технических и других подготовительных работ, связанных с созданием АС.

При выполнении этапа “Разработка рабочей документации на систему и ее части” осуществляют разработку рабочей документации, содержащей все необходимые и достаточные сведения для обеспечения выполнения работ по вводу АС в действие и ее эксплуатации, а также для поддержания уровня эксплуатационных характеристик (качества) системы в соответствии с принятыми проектными решениями, ее оформление, согласование и утверждение.

На этапе “Разработка и адаптация программ” проводят разработку программ и ПС, выбор, адаптацию и (или) привязку приобретенных программных средств, разработку программной документации в соответствии с ГОСТ 19.101-77.

На этапе “Подготовка объекта автоматизации к вводу АС в действие” проводят работы по организационной подготовке объекта автоматизации к вводу АС в действие.

Этап “Подготовка персонала” направлен на обучение персонала и проверку его способности обеспечить функционирование АС.

На этапе “Комплектация АС поставляемыми изделиями” обеспечивают получение комплектующих изделий серийного и единичного производства, материалов и монтажных изделий. Проводят входной контроль их качества.

На этапе “Пусконаладочные работы” проводят автономную наладку технических и программных средств, загрузку информации в БД и проверку системы ее ведения; комплексную наладку всех средств системы.

На этапе “Проведение предварительных испытаний” осуществляют испытание АС на работоспособность и соответствие ТЗ в соответствии с программой и методикой предварительных испытаний, устранение неисправностей и внесение изменений в документацию на АС, в том числе эксплуатационную в соответствии с протоколом испытаний, а также оформление акта о приемке АС в опытную эксплуатацию.

На этапе “Проведение опытной эксплуатации” проводят опытную эксплуатацию АС, анализ результатов опытной эксплуатации АС, доработку (при необходимости) программного обеспечения АС, дополнительную наладку (при необходимости) технических средств АС и оформление акта о завершении опытной эксплуатации.

На этапе “Проведение приемочных испытаний” проводят испытания на соответствие ТЗ в соответствии с программой и методикой приемочных испытаний, анализ результатов испытаний АС и устранение недостатков, выявленных при испытаниях, а также оформление акта о приемке АС в постоянную эксплуатацию.

На этапе “Выполнение работ в соответствии с гарантийными обязательствами” осуществляются работы по устранению недостатков, выявленных при эксплуатации АС в течение установленных гарантийных сроков, внесению необходимых изменений в документацию на АС.

На этапе “Послегарантийное обслуживание” осуществляются работы по анализу функционирования системы, выявлению отклонений фактических эксплуатационных характеристик АС от проектных значений, установлению причин этих отклонений, устранению выявленных недостатков и обеспечению стабильности эксплуатационных характеристик АС, внесению необходимых изменений в документацию на АС.

Требования к содержанию документов, разрабатываемых на каждой стадии и этапе, устанавливает руководящий документ РД-50-34.698-90. Кроме того, необходимо использовать для их разработки соответствующие стандарты ЕСПД, ЕСКД (Единой системы конструкторской документации) и ГОСТ 34.602-89 “Техническое задание на создание АС”. Виды и комплектность документов регламентированы в ГОСТ 34.201.

2.3. Жизненный цикл разработки ПС с повышенными требованиями к безопасности системы

Жизненный цикл разработки ПС, регламентированный документом DO-178 [6], заслуживает внимания из следующих соображений. Программные средства, используемые в бортовых системах и оборудовании летательных аппаратах, исполняют свою функцию с уровнем доверия к безопасности, которая соответствует требованиям авиационного приложения. Данные требования авиационного приложения отличаются от аналогичных в других сферах более жесткими требованиями по качеству, надежности, так как влияют на безопасность пассажиров и характеристики самого летательного аппарата. Необходимо заметить, что описанные ниже руководящие принципы относительно жизненного цикла, не возведены в ранг закона, но согласованы с авиационным сообществом.

На рис. 2.4. показан краткий обзор аспектов безопасности информационного потока между этапами жизненного цикла системы и этапами жизненного цикла ПС.



Рис. 2.4. Информационные потоки между системой и этапами жизненного цикла ПС

Этап оценки безопасности системы определяет категорию отказа системы. В рамках данного этапа, анализ проекта системы опреде-

ляет требования, связанные с безопасностью, для аппаратных средств ЭВМ и ПС, для устранения и ограничения влияния ошибок, с ними связанных. Требования, связанные с безопасностью – часть требований к системе, которые вводятся в этапы жизненного цикла ПС. В целях гарантии того, что требования, связанные с безопасностью, учитываются на всех этапах жизненного цикла ПС, требования к системе включают:

- описание системы и аппаратных средств ЭВМ;
- требования сертификации;
- требования системы, относящиеся к ПС, включая функциональные и эксплуатационные требования, а также требования, связанные с безопасностью;
- уровни ПС, состояния отказа, их категории и связанные с этим функции, относящиеся к ПС;
- стратегии безопасности и проект разработки.

Этап оценки безопасности системы определяет воздействия процессов проектирования и внедрения ПС на безопасность системы путем использования информации, поступающей от этапов жизненного цикла ПС. Эта информация включает границу распространения неисправности, требования ПС и его архитектуру, а также источники ошибок, которые могут быть обнаружены и устранены благодаря ПС или при помощи инструментов и методов, используемых при разработке ПС.

Имеют место следующие этапы ЖЦ ПС, установленные данным документом:

- *этап планирования разработки*, который определяет и координирует действия по разработке программного обеспечения;
- *этап разработки*, на котором создается ПС и включающий в себя работы по определению требований, проектированию, кодированию, получению исполняемого кода;
- *интегрированный этап*, обеспечивающий корректировку, проверку и определение полноты ПС, а также его выпуск и включающий в себя верификацию, контроль за конфигурацией ПС, оценку качества и проверку взаимодействия этапом друг с другом.

На рис. 2.5 показаны этапы ЖЦ ПС и их взаимодействие друг с другом.

Проект ПС определяет один или более жизненных циклов посредством выбора действий на каждом этапе, назначения последовательности этих действий и ответственных за них.

Для разработки конкретного ПС последовательность этапов определяется на основе анализа таких свойств, как функциональные

возможности, сложность, размер, устойчивость, использование ранее полученных результатов, разрабатываемой стратегией и аппаратной поддержкой. Обычная последовательность этапов жизненного цикла ПС, как показано на рис. 2.5, включает определение требований, проектирования, кодирования и получения исполняемого кода.



Рис. 2.5. Этапы жизненного цикла ПС

На рис.2.6 показана последовательность этапов жизненного цикла для компонентов единичного программного продукта с различными жизненными циклами.

Компонент W выполняет множество системных требований для разработки программных требований R , использует эти требования для проектирования ПС D , исполняет этот проект в исходном коде C и интегрирует компонент с аппаратурой I .

Компонент X показывает использование ранее разработанного ПС, которое уже было сертифицировано.

Компонент Y показывает использование простой отдельной функции, которая может быть закодирована прямо на основе требований к ПС.

Компонент Z показывает использование прототипной стратегии. Обычно, целями прототипирования является лучшее понимание требований к ПС и уменьшение технических рисков и рисков разработки. Исходные требования используются как базис для получения прототипа. Этот прототип преобразуется в среду, типичную для конкретного использования системы при разработке. Результатом преобразования является использование уточненных данных.

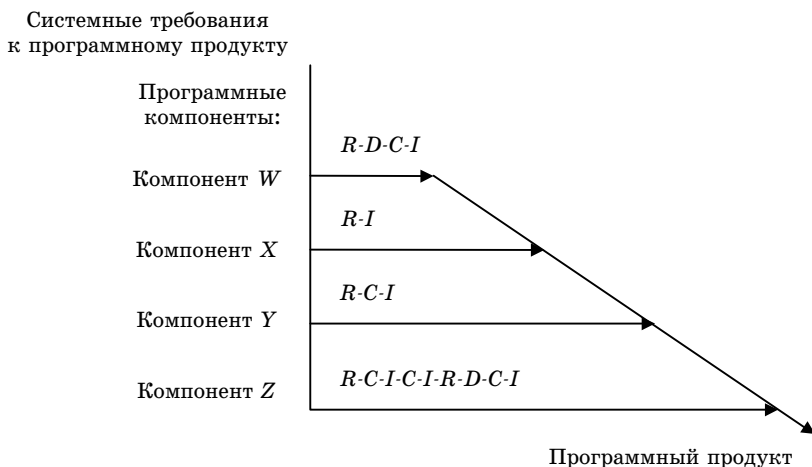


Рис. 2.6. Пример проекта ПС, использующего четыре различные последовательности разработки

Этапы жизненного цикла могут носить итерационный характер, т. е. повторяться несколько раз. Количество этапов и итераций варьируется из-за доработок функций ПС, сложности, появления новых требований или нового аппаратного обеспечения, результатов выполнения предыдущих этапов, которые претерпели изменения, а также других особенностей.

Практически все этапы жизненного цикла ПС объединяются с интегрированным этапом и действиями по верификации.

Для перехода с одного этапа на другой необходимо предусмотреть критерии перехода. Критерии перехода будут зависеть от запланированной последовательности шагов разработки и шагов интегрированного этапа, а также от уровня ПС (уровень ПС определяется на основе анализа вклада, который он вносит в набор потенциальных состояний отказа системы). Например, в качестве критерия перехода можно назвать:

- что было выполнено при рецензировании на этапе верификации;

- вход определенного элемента конфигурации;

- трассировку, проведенную для входа.

Если удовлетворен критерий перехода к последующему этапу, то нет необходимости в том, чтобы каждый вход этого этапа был сформирован до начала его выполнения. Руководящий принцип следующий: если этап производит действия над отдельными входами, последовательность входов должна быть проверена с целью определения того, что выходы предшествующих шагов этапа разработки и интегрированного этапа еще имеют силу.

Рассмотрим подробно цели и действия каждого этапа ЖЦ.

На этапе планирования разработки ПС создаются планы и выбираются стандарты, которые направляют этап разработки и интегрированный этап. Его целью является определение средств для создания ПС, которое будет удовлетворять требованиям, предъявляемым к нему и обеспечивать достаточный уровень надежности. На этом этапе производится:

- определение действий на этапах разработки и интегрированном этапе, которые будут посвящены определению системных требований и уровня ПС;

- определение ЖЦ ПС, включая взаимодействие между этапами, механизм взаимного влияния этапов, критерии оценки ПС при переходе от одного этапа к другому;

- определение среды ЖЦ, т. е. методов и инструментальных средств, используемых на каждом этапе;

- формирование дополнительных замечаний к ПС;

- рассмотрение стандартов разработки ПС, соотношение их с системными целями безопасности, относящихся к разрабатываемому ПС;

- разработка плана создания ПС;

- доработка и проверка плана создания ПС.

Эффективность планирования — определяющий фактор при разработке ПС. Основные руководящие принципы этого этапа следующие.

1. План разработки ПС должен быть разработан в такой момент ЖЦ, чтобы обеспечить своевременное управление конкретными действиями на этапе разработки и интегрированном этапе.

2. Стандарты разработки ПС, используемые в проекте, должны быть строго определенными и четкими.

3. Выбранные методы и инструментальные средства должны быть такими, чтобы обеспечить предотвращение ошибок на этапе разработки или свести их к минимуму.

4. Этап планирования разработки ПС должен обеспечить координацию между остальными этапами с целью согласования их стратегий в плане разработки ПС.

5. Этап планирования разработки ПС должен включать в себя средства проверки плана на этапе реализации проекта.

6. На завершающей стадии этапа планирования, план и стандарты разработки ПС должны быть проанализированы на предмет полноты и непротиворечивости.

Другие этапы ЖЦ могут быть начаты до окончания этапа планирования при условии, что имеются планы и процедуры для действий на этих этапах.

План разработки включает в себя следующие компоненты:

- план программных аспектов сертификации, служащий основным средством объединения предлагаемых методов разработки со службами сертификации;

- собственно план разработки, определяющий жизненные циклы и среду разработки;

- план верификации ПС, определяющий средства, с помощью которых удовлетворяются цели этапа верификации;

- план управления конфигурацией ПС, определяющий средства, с помощью которых удовлетворяются цели этапа оценки качества.

Задачей определения среды ЖЦ является обоснование методов, инструментальных средств, процедур, языков программирования и аппаратного обеспечения, которые будут использоваться при разработке, верификации, контроле и выпуске данных этапов жизненного цикла и ПС в целом. Примером того, как выбор среды оказывает полезное влияние на ПС, может служить строгое соблюдение стандартов, определение ошибок, предотвращение ошибок исполнения, методов смягчения последствий сбоев в ПС. Среда ЖЦ является потенциальным источником ошибок, которые способствуют возникновению сбоев ПС. На состав среды влияют требования по безопасности, определяемые этапом оценки безопасности системы.

Цель методов предотвращения ошибок – недопущение ошибок в ходе этапа разработки. Основной принцип при выборе требований к

разработке ПС, методам его проектирования и программирования заключается в ограничении числа случаев внедрения ошибок и применении таких методов верификации, которые гарантируют установление факта внедрения ошибки.

Целью методов смягчения последствий сбоев является включение характеристик безопасности в проект ПС для гарантии того, что ПС будет корректно реагировать на ошибки входных данных и предотвращать ошибки выходных данных и ошибки управления. Необходимость в методах предотвращения ошибок и смягчения последствий сбоев определяются системными требованиями и этапом оценки безопасности системы.

Среда разработки ПС – определяющий фактор при разработке высококачественного и надежного ПС. Среда может оказывать также и неблагоприятное влияние на процесс разработки ПС. Например, компилятор может вносить ошибки в процессе компиляции, загрузчик может давать сбои при обнаружении ошибок распределения памяти. Руководящие принципы для выбора методов и инструментальных средств среды разработки ПС следующие.

1. В ходе этапа планирования среда разработки должна быть выбрана исходя из минимума потенциального риска для конечного ПС.

2. Использование специализированных инструментальных средств или инструментов и компонентов среды должно производиться путем достижения необходимого уровня гарантии того, что ошибки, вносимые одним компонентом, обнаружались бы другим. В этом случае среда является приемлемой.

3. Действия на этапе верификации и стандарты разработки должны быть определены так, чтобы свести к минимуму потенциальные ошибки, относящиеся к среде разработки ПС.

4. При поиске определенного уровня доверия к сертификации при использовании комбинации инструментов последовательность действий, производимых этими инструментами, должна быть определена в соответствующем плане.

5. Если определены необязательные характеристики инструментальных средств среды для использования в проекте, то влияние этих характеристик должно быть проверено и специфицировано в соответствующем плане. Это особенно важно там, где инструмент непосредственно создает часть ПС (в этом смысле компиляторы, вероятно, наиболее важный инструмент).

Назначением стандартов разработки ПС является определение правил и ограничений для этапа разработки. Эти стандарты включают в себя:

– стандарт требований к ПС, который предназначен для определения методик, правил и средств, используемых при разработке требований высокого уровня, и включает:

а) методы, используемые при разработке требований к ПС (структурные, объектно-ориентированные и др.);

б) нотации, используемые для выражения этих требований (диаграммы потоков данных, спецификации формальных языков и др.);

в) ограничения на использование средств разработки требований;

– стандарты проектирования ПС, которые позволяют определить методики, правила и средства, используемые для создания архитектуры ПС, и требования низкого уровня, включающие:

а) используемые методы описания процесса проектирования;

б) используемые соглашения по именам;

в) условия, налагаемые на разрешенные методы проектирования (например, использование прерываний и архитектур, управляемых событиями, динамическое управление задачами и т. д.);

г) ограничения на использование средств проектирования;

д) ограничения на конструкцию (например, исключение, рекурсия, динамические объекты, ссылки на данные под разными именами и т. д.);

е) ограничения по сложности (например, максимальное число вложенных вызовов процедур или условных структур, использование безусловных ветвлений и т. д.);

– стандарты кодирования ПС, которые позволяют определить языки программирования, методы, правила и средства, используемые для кодирования ПС и включающие:

а) используемые языки программирования и/или их подмножества (для языка необходимо указать данные, которые непротиворечиво определяют синтаксис, поведение управления, поведение данных и сторонние эффекты языка);

б) стандарты на представление исходного кода (например, ограничения на длину строки, отступы, использование пустых строк), стандарты на документирование исходного кода (например, имя автора, дата написания, входные и выходные данные, используемые глобальные переменные);

в) соглашения по именам для компонентов, подпрограмм, переменных, констант;

г) условия и ограничения, налагаемые на разрешенные соглашения по коду (например, степень связей между компонентами ПС и сложность логических и числовых выражений);

д) ограничения на использование средств кодирования.

Этап верификации использует перечисленные стандарты как основу для сравнения выходов этапов с требуемыми результатами. Руководящими принципами, которыми необходимо пользоваться при определении стандартов, являются следующие.

1. Стандарты должны давать возможность компонентам ПС или взаимосвязанному множеству продуктов разрабатываться единообразно.

2. Стандарты должны обеспечить невозможность использования таких конструкций или методов, в результате выполнения которых получают выходы, не соответствующие требованиям безопасности.

Как отмечалось выше, составляющими этапа разработки ПС являются следующие подэтапы:

- разработка требований к ПС;
- проектирование ПС;
- кодирование ПС;
- интеграция ПС.

При разработке ПС вырабатывается один и более уровней требований к ПС. Требования высокого уровня вытекают непосредственно из анализа требований к системе и ее архитектуре. Далее они разбиваются на подэтапе проектирования ПС, формируя один и более последующих уровней требований, более низких. Однако если исходный код генерируется непосредственно из требований высокого уровня (например, компонент Y на рис. 2.6), то они могут рассматриваться одновременно и как требования низкого уровня. В этом случае к ним применяются соответствующие основополагающие принципы формирования.

Разработка архитектуры ПС предполагает принятие решения о структуре ПС. На подэтапе проектирования ПС определяется как архитектура ПС, так и требования низкого уровня – такие требования к ПС, исходя из которых, без какой либо дополнительной информации, может быть непосредственно реализован исходный код.

Каждая из составляющих этапа разработки ПС может порождать производные требования – требования, которые прямо не сводятся к требованиям высокого уровня. Например, необходимость разработки ПС поддержки прерываний для выбранного целевого компьютера. Требования как высокого (ВУ), так и низкого уровня (НУ) могут включать в себя производные требования. Влияние последних на требования безопасности определяется на этапе оценки безопасности системы.

В табл. 2.6 показаны цели каждого подэтапа, составляющие этап разработки, их входные данные, результаты, основные принципы, применяемые на каждом из них.

Таблица 2.6

Наименование	Цель и входные данные	Результат и основные принципы
1	2	3
Разработка требований к ПС	<p>Получение требований ВУ и выработка производных от них требований для этапа оценки безопасности</p> <p>Входные данные:</p> <ul style="list-style-type: none"> – требования к системе, аппаратный интерфейс, архитектура системы; – план разработки ПС; – стандарты на требования 	<p>Первичный результат – данные о требованиях</p> <p>Основные принципы:</p> <ul style="list-style-type: none"> – интерфейсные и функциональные требования к системе, реализуемые на базе ПС, должны быть проанализированы на предмет противоречий, несоответствия и неопределенности; – некорректные входные данные должны быть направлены на выработавшие их подэтапы для исправления; – каждое требование к системе, реализуемое на базе ПС, должно быть включено в требования ВУ; – должны быть особо отмечены требования ВУ, соответствующие системным требованиям по предотвращению выхода системы из строя; – требования ВУ должны соответствовать стандартам на требования к ПС; – требования ВУ должны формулироваться в количественных терминах; – требования ВУ не должны описывать детали разработки или тестирования, за исключением указанных и проверенных ограничений конструкции; – каждое системное требование, реализуемое на базе ПС, должно быть сводимо к одному или более требованиям ВУ к ПС; – каждое требование ВУ, за исключением производных требований, должно быть сводимо к одному или нескольким системным требованиям; – производные требования ВУ должны быть направлены к этапу оценки безопасности системы
Проектирование ПС	Создание архитектуры ПС и требований НУ, а	Первичный результат – описание разработки, включающее архитектуру ПС и требования НУ

1	2	3
	<p>также выработка производных от них требований для этапа оценки безопасности</p> <p>Входные данные:</p> <ul style="list-style-type: none"> – данные о требованиях к ПС; – план разработки ПС; – стандарты проектирования 	<p>Основные принципы:</p> <ul style="list-style-type: none"> – создаваемые требования НУ и архитектура ПС должны соответствовать стандартам разработки ПС, быть непротиворечивыми и допускать трассировку и проверку; – определяемые производные требования должны быть проанализированы на предмет соответствия требованиям ВУ; – на подэтапе проектирования ПС следует ввести возможные типы сбоев ПС и наоборот предотвратить остальные, что может изменить ранее назначенный программный уровень и определить дополнительные данные в качестве производных требований, передаваемых этапу оценки безопасности системы; – потоки данных и управления должны быть наблюдаемы согласно требованиям безопасности; – реакции на условия сбоев должны соответствовать требованиям безопасности; – неадекватные или некорректные входные данные должны быть переданы либо этапу оценки жизненного цикла системы, либо подэтапу разработки требований, либо этапу планирования разработки ПС по принципу обратной связи для разъяснения или исправления
Кодирование ПС	<p>Получение исходного кода, который должен допускать трассировку, проверку, быть непротиворечивым и корректно реализовывать требования НУ</p> <p>Входные данные:</p> <ul style="list-style-type: none"> – требования НУ; – архитектура ПС; – план разработки ПС; – стандарты кодирования ПС 	<p>Результат – исходный код, объектный код</p> <p>Основные принципы:</p> <ul style="list-style-type: none"> – исходный код должен реализовывать требования НУ и соответствовать архитектуре ПС; – исходный код должен соответствовать стандартам кодирования ПС; – исходный код должен сводиться к описанию проекта; – неадекватные или некорректные входные данные должны быть переданы либо подэтапу разработки требований к ПС, либо подэтапу проектирования ПС, либо этапу планирования разработки ПС по принципу обратной связи для разъяснения или исправления

1	2	3
Интеграция ПС	Загрузка исполняемого объектного кода в аппаратное или программно-аппаратное обеспечение Входные данные: – архитектура ПС; – исходный код; – объектный код	Результат – исполняемый объектный код, а также компоуемые и загружаемые данные Основные принципы: – исполняемый объектный код должен быть сгенерирован из исходного кода и компоуемых и загружаемых данных; – ПС должно быть загружено в целевой компьютер для программно-аппаратной интеграции; – неадекватные или некорректные входные данные должны быть переданы либо подэтапу разработки требований к ПС, либо подэтапу проектирования ПС, либо подэтапу кодирования ПС, либо этапу планирования разработки ПС по принципу обратной связи для разъяснения или исправления

Подэтап разработки требований к ПС использует выходные данные жизненного цикла системы для выработки таких требований высокого уровня, как функциональные требования, требования производительности, интерфейсные требования и требования безопасности.

Требования высокого уровня к ПС на подэтапе проектирования перерабатываются в ходе одной или более итераций в архитектуру ПС и требования низкого уровня, которые в свою очередь используются для реализации исходного кода.

На подэтапе кодирования исходя из архитектуры ПС и требований низкого уровня, вырабатывается исходный код ПС.

Целевой компьютер, а также исходный и объектный код, получаемые из подэтапа кодирования, используются совместно с компоновкой и загрузкой данных на подэтапе интеграции для создания интегрированной системы. Подэтап интеграции включает программную и программно-аппаратную интеграцию.

Все подэтапы этапа разработки могут начинаться при удовлетворении планируемого промежуточного критерия перехода. Подэтапы считаются завершенными, когда их цели, а также цели всех связанных с ними подэтапов, достигнуты.

Верификация, входящая в состав интегрированного этапа – это техническая оценка результатов как этапа разработки ПС, так и этапа верификации ПС (последний применяется согласно этапу плани-

рования разработки ПС и плана верификации ПС). Верификация не сводится только к тестированию ПС. Тестирование, в общем случае, не в состоянии показать отсутствие ошибок, поэтому употребляется термин “верификация”, а не “тестирование”.

Целью этапа верификации является обнаружение ошибок, которые могли быть внесены в ПС на этапе его разработки и корректировка таких ошибок должна производиться на этом же этапе. Основной целью этапа верификации является проверка следующего.

1. Системные требования, реализуемые на базе ПС, были соответствующим образом воплощены в требования высокого уровня.
2. Требования высокого уровня были соответствующим образом воплощены в архитектуре ПС и требованиях низкого уровня.
3. Архитектура и требования низкого уровня были соответствующим образом воплощены в исходном коде.
4. Исполняемый объектный код соответствует требованиям к ПС.
5. Средства, используемые для достижения выше перечисленных целей, являются технически корректными и функционально законченными для программного уровня.

Методы этапа верификации базируются на комбинации обзоров, анализе, разработке тестовых событий и процедур и последующего их исполнения. Обзоры и анализ обеспечивают оценку точности, полноты, проверяемости требований к ПС, архитектуры ПС и исходного кода. Разработка тестовых событий обеспечивает более полную оценку внутренней состоятельности ПС и полноту требований к нему. Выполнение тестовых процедур обеспечивает демонстрацию соответствия ПС назначенным требованиям.

Входные данные этапа верификации: требования к системе, требования к ПС и архитектуре ПС, данные о трассируемости, исходный код, исполняемый объектный код, план верификации ПС.

Выходные данные этапа верификации ПС фиксируются в Сводке ситуаций и процедур верификации ПС, а также в Сводке результатов верификации ПС.

Этап верификации обеспечивает проверку соответствия реализации требованиям к ПС и верификацию этих требований следующими методами.

1. Соответствие между требованиями к ПС и тестовыми ситуациями выполняется путем анализа области покрытия требований.
2. Соответствие между структурой кода и тестовыми ситуациями выполняется путем анализа области покрытия структуры.

Основные методы этапа верификации ПС:

– необходимо верифицировать требования высокого уровня и сводимость реализации к этим требованиям;

— результаты анализа соответствия реализации, анализа области покрытия требований и анализа области покрытия структуры должны показывать, что каждое требование к ПС сводимо к области кода, реализующего это требование, а также к обзорам, анализам и тестовым ситуациям, проверяющим его;

— если нет возможности верифицировать специфические требования к ПС путем эксплуатации ПС в реальных условиях, должны быть обеспечены иные средства, а в плане верификации ПС или сводке результатов верификации ПС должно быть отмечено их использование в этапе верификации ПС.

К результатам этапов разработки и верификации ПС могут быть применены различные виды анализа и обзоров. Различия между обзором и анализом заключается в том, что анализ дает повторяемые доказательства корректности, а обзор обеспечивает количественную оценку корректности. Обзор может состоять из исследования выходных данных этапа, выполненного в виде диаграммы. Анализ может исследовать в деталях функционирование, производительность, трассируемость и вопросы безопасности компонента ПС.

Обзоры и анализ требований высокого (ВУ), низкого уровней (НУ), архитектуры, исходного кода показаны в табл. 2.7. Цель таких обзоров и анализов заключается в обнаружении ошибок в требованиях, которые могли быть внесены в ходе выполнения подэтапа разработки требований к ПС (проектирования, разработки архитектуры, кодирования).

Таблица 2.7

Критерии	Требования ВУ	Требования НУ	Архитектура ПС	Исходный код
1	2	3	4	5
Подчинение системным требованиям (требованиям ВУ), (требованиям НУ)	Функции системы, выполняемые на базе ПС, определены; функциональные требования, требования к производительности и безопасности отражены в требованиях ВУ; производные требования корректны и необходимы	Требования ВУ отражены в требованиях НУ; производные требования корректны и действительно необходимы	Архитектура ПС не конфликтует с требованиями ВУ, особенно в функциях обеспечения целостности системы	Исходный код корректен и полноценен по отношению к требованиям НУ; не реализуются недокументированные функции

Продолжение табл. 2.7

1	2	3	4	5
Точность и состоятельность	Каждое требование ВУ точно, корректно, непротиворечиво и не конфликтует с другими требованиями	Каждое требование НУ точно, корректно, непротиворечиво и не конфликтует с другими требованиями	Между компонентами архитектуры ПС существует корректные отношения. Отношения выражаются через потоки данных и управления	Определить корректность и состоятельность исходного кода
Совместимость с целевым компьютером (Ц)	Нет конфликтов между требованиями ВУ и возможностями Ц	Нет конфликтов между требованиями к ПС и возможностями Ц	Нет конфликтов между архитектурой ПС и возможностями Ц	
Проверяемость	Каждое требование ВУ может быть проверено	Каждое требование НУ может быть проверено	Архитектура ПС может быть проверена	Исходный код не содержит операторов и структур, которые не могут быть проверены
Соответствие стандартам	На подэтапе разработки требований к ПС соблюдались стандарты на требования, а все отклонения от них исправлены	На подэтапе проектирования соблюдались стандарты на разработку ПС, а все отклонения от них исправлены	На подэтапе проектирования соблюдались стандарты на разработку ПС, а все отклонения от них исправлены	На подэтапе кодирования соблюдались стандарты на кодирования ПС, а все отклонения от них исправлены
Трассируемость	Системные функциональные требования, требования производительности и безопасности, реализуемые на базе ПС, были отражены в требованиях ВУ	Требования ВУ и производные от них были отражены в требованиях НУ	Нет	Низкоуровневые требования были отражены в исходном коде

1	2	3	4	5
Аспекты алгоритмов	Убедиться в точности и корректности поведения используемых алгоритмов, особенно в областях перехода от одного алгоритма к другому	Убедиться в точности и корректности поведения используемых алгоритмов, особенно в областях перехода от одного алгоритма к другому	Нет	Нет

Цель обзоров и анализа результатов подэтапа интеграции – убедиться в том, что результаты данного этапа полны и корректны. Это может быть выполнено путем детального исследования компонентов и загружаемых данных и карты памяти и рассмотрения следующих вопросов: некорректность аппаратных адресов, перекрытия областей памяти, отсутствия отдельных компонентов ПС.

Тестирование ПС имеет две взаимодополняющие цели. Одна из них – продемонстрировать, что ПС удовлетворяет требованиям к нему; вторая – продемонстрировать с высокой точностью, что ошибки, приводящие к сбою системы, согласно этапу оценки безопасности, были удалены. На рис. 2.7 показана диаграмма тестирования ПС. Целями трех типов тестирования являются:

- тестирование программно-аппаратной интеграции – для проверки корректности выполнения программной операции в среде целевого компьютера;

- тестирование программной интеграции – для проверки взаимодействия между требованиями к ПС и его компонентами, а также правильности реализации требований к ПС и его компонентов в рамках архитектуры ПС;

- низкоуровневое тестирование – для проверки правильности реализации требований низкого уровня к ПС.

Для достижения целей тестирования необходимо:

- тестовые ситуации должны базироваться, главным образом, на требованиях к ПС;

- тестовые ситуации должны быть разработаны для проверки правильности функционирования и выявления условий, в которых могут быть выявлены потенциальные ошибки;

- анализ покрытия требований к ПС должен определить, какие требования не тестировались;
- анализ покрытия структуры должен определить, какие программные структуры не исследовались.

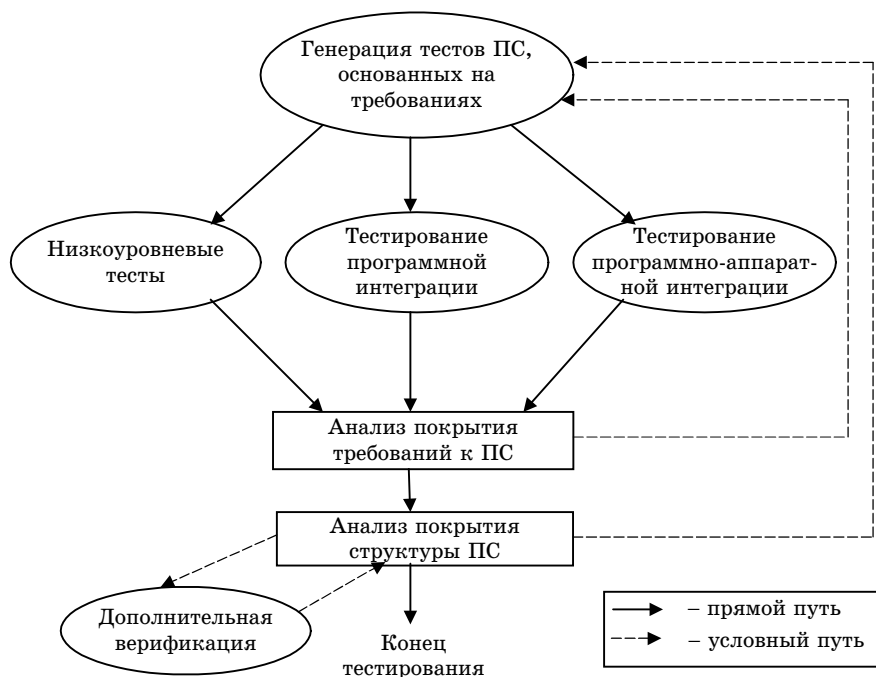


Рис. 2.7. Диаграмма тестирования ПС

Этап контроля конфигурацией работает совместно с другими этапами жизненного цикла ПС и позволяет достичь следующих целей.

1. Обеспечение определенной управляемой конфигурации ПС в течение жизненного цикла.

2. Обеспечение возможности размножить полноценный исполняемый объектный код для выпуска ПС или воспроизвести его в случае необходимости.

3. Обеспечение управления входными и выходными данными этапа в течение ЖЦ ПС, которое обеспечит состоятельность и воспроизводимость деятельности этапа.

4. Выработка исходных данных для обзоров и оценки статуса: изменение раз за разом параметров конфигурации до установления базового режима.

5. Обеспечение управления, которое даст возможность уделять внимание проблемам и записывать, оценивать и реализовывать изменения.

6. Обеспечение открытости информации о ПС путем управления выходными данными этапов ЖЦ ПС.

7. Помощь в оценке соответствия ПС требованиям на него.

8. Обязательное включение в параметры конфигурации поддержки безопасного физического архивирования, восстановления и управления.

Этап управления и контроля конфигурации включает в себя идентификацию конфигурации, управление изменениями, установление базовой конфигурации, архивацию ПС и данных ЖЦ ПС. Данный этап не останавливается после сертификации ПС, а продолжает функционировать в течение всего срока службы системы.

На этапе оценки качества ПС оцениваются выходные данные подэтапов жизненного цикла ПС для принятия решения об удовлетворении поставленных целей, обнаружении, оценке и устранении ошибок, согласования программных средств и данных ЖЦ с требованиями сертификации. Этот этап применяется, согласно этапу планирования разработки и плана оценки качества ПС. Выходные данные этапа оценки качества формируются в Реестре оценки качества ПС или других данных жизненного цикла ПС.

Этап оценки качества должен определить, произошла ли в результате выполнения подэтапов ЖЦ ПС выработка такого ПС, которое удовлетворяло бы поставленным требованиям, допуская, что эти подэтапы выполнялись в соответствии с принятыми планами и стандартами разработки ПС.

Цель этапа оценки качества — убедиться в том, что этап разработки ПС удовлетворяет принятым планам и стандартам разработки ПС, а также удовлетворен промежуточный критерий для подэтапов ЖЦ ПС, проведена ревизия программного продукта.

Для достижения целей этапа оценки качества необходимо следующее.

1. Активная роль в жизненном цикле ПС.
2. Обеспечивать создание и оценку корректности планов и стандартов разработки ПС.
3. Обеспечивать протекание этапов жизненного цикла ПС, согласно указанным планам и стандартам.
4. Фиксировать события этапов разработки и интеграции в течение жизненного цикла ПС с целью определения того, что:
 - отклонения от планов и стандартов разработки обнаружены, записаны, оценены, проверены и исправлены (общепринято мнение

о том, что, чем раньше обнаружены такие отклонения, тем более эффективно достигаются цели этапов ЖЦ ПС);

- санкционированные отклонения зафиксированы особо;
- среда разработки ПС используется, согласно планам разработки ПС;
- процесс оповещения о проблемах, их исправления и проверки проводятся, согласно плану управления конфигурации ПС;
- успешно получены результаты этапа оценки безопасности системы и внесены в жизненный цикл системы.

5. Этап управления и контроля конфигурации ПС должен контролировать удовлетворение промежуточных критериев ЖЦ ПС, согласно плану разработки ПС.

6. Этап управления и контроля конфигурации ПС должен проверить контролируемость данных его ЖЦ.

7. Прежде чем рассылать копии ПС для проведения его сертификации, нужно выполнить его ревизию.

8. Этап управления и контроля конфигурации ПС должен фиксировать развитие событий и результаты ревизии ПС для каждого сертифицируемого программного продукта.

Основная цель ревизии заключается в том, чтобы убедиться, что для сертифицированного программного продукта этапы ЖЦ завершены, их выходные данные сформированы, исполняемый объектный код контролируется и может быть воспроизведен.

Целью проведения сертификации является установление взаимопонимания между производителем ПС и службами, осуществляющими сертификацию, в течение всего ЖЦ, что облегчит этап сертификации. Данный этап применяется, согласно этапу планирования разработки ПС и плану программных аспектов сертификации.

Производитель ПС должен предоставить доказательства того, что жизненный цикл ПС протекает, согласно плану его разработки. Служба сертификации может выдавать свои обзоры для производителя ПС или его поставщика с обсуждением тех или иных аспектов ЖЦ. Производитель коррелирует замечания с методиками ЖЦ ПС и предоставляет требуемые данные. Производитель программного средства обязан:

- рассматривать обзоры служб сертификации;
- передавать сводку работ, выполненных над ПС и реестр конфигурации службам сертификации.

Кроме того, службам сертификации может быть передан план программных аспектов сертификации.

Обычно, если не возражают службы сертификации, меры по регулированию данных ЖЦ ПС, относящиеся к разработке типичного

образца, применяются к данным о требованиях к ПС, описанию проектирования, исходному коду, исполняемому объектному коду, реестру конфигурации ПС, сводке работ, выполненных над ПС.

Необходимо заметить, что руководящие принципы документа DO-178 В, в целом, удовлетворяют целям следующих международных стандартов: ИСО 9000-3-91 “Стандарты в области административного управления качеством и обеспечения качества. Часть 3. Руководящие положения по применению ИСО 9001 при разработке, поставке и техническом обслуживании ПО”, IEC 65A (Secretariat) 122 (ноябрь 1991) “ПО для компьютеров, используемых в промышленных системах безопасности”.

2.4. Процессы жизненного цикла разработки ПС

ИСО/МЭК 12207-95 определяет модель жизненного цикла процессов разработки программного обеспечения. Данная модель жизненного цикла ПС определяет на верхнем уровне фундаментальные цели, которые являются существенными для разработки высокоэффективного и надежного программного обеспечения. Цели верхнего уровня описывают то, что должно быть достигнуто, а не как их достигнуть.

Жизненный цикл, определенный данным стандартом, применим в любой software-организации, желающей утвердить, а потом и улучшить возможности по поставке, разработке, эксплуатации, развитию и поддержке программного обеспечения. Модель не предполагает использование специфических организационных структур, философии управления, технологии или методологии разработки ПС.

Архитектура данной модели организует процессы для помощи персоналу организации в их понимании и их использования для непрерывного усовершенствования процессов разработки программного обеспечения. Процессы, определенные в стандарте, образуют всеобъемлющий набор процессов. Организация в зависимости от своих целей и для их реализации может выбрать соответствующий поднабор. Поэтому стандарт разработан так, чтобы быть приспособленным как для отдельной организации, так и для конкретного проекта или приложения. Кроме того, он может использоваться в случаях, когда программное обеспечение является самостоятельной сущностью, встраивается или является составной частью общей системы.

ИСО/МЭК 12207-95 не поддерживает (не устанавливает) какую-либо конкретную модель жизненного цикла программного обеспечения, управление программным обеспечением, метод разработки или

локальную промышленную технологию. Еще раз повторим, что он также не предписывает, каким образом исполнять что-либо. Эти моменты очень сильно зависят от условий конкретного проекта и технологического уровня организации и остаются в ее компетенции.

Данный стандарт взаимосвязан со следующими стандартами:

– ИСО/МЭК 15504-97. “Информационная технология – Оценка процесса разработки программного обеспечения”;

– ИСО 9001-94. “Системы качества – Модель для обеспечения качества при проектировании, разработке, производстве, монтаже и обслуживании”.

Все действия, которые могут осуществляться в процессе жизненного цикла ПС, рассматриваемый стандарт разбивает на три группы процессов жизненного цикла, согласно типу действия, которым они направляются. Каждый процесс жизненного цикла подразделяется на ряд действий. Каждое действие подразделяется на ряд задач. Под задачей понимается действие, преобразующее входы в выходы. Перечень задач, приведенный ниже, не является исчерпывающим и приведен в качестве примера. Процессы, действия и задачи могут выполняться последовательно, параллельно, повторно и комбинированно. Группы и входящие в них процессы показаны на рис. 2.8.

Основные процессы. Раздел 5 ИСО/МЭК 12207-95 определяет, что основными процессами являются.

1. Процесс приобретения. Определяет действия заказчика, организации, приобретающих систему или программный продукт.

2. Процесс поставки. Определяет действия поставщика, организации, предоставляющих программный продукт заказчику.

3. Процесс разработки. Определяет действия разработчика, организации, определяющих и разрабатывающих программный продукт.

4. Процесс эксплуатации. Определяет действия оператора, организации, предоставляющих для пользователей услуги по эксплуатации компьютерной системы в ее рабочей среде.

5. Процесс сопровождения. Определяет действия сопровождающего, организации, предоставляющих услуги по сопровождению программного обеспечения, т. е. управлению изменениями в программном обеспечении для поддержки его в актуальном и работоспособном состоянии. Этот процесс включает перемещение и отставку программного обеспечения.

Сопроводительные процессы. Раздел 6 данного стандарта представляет собой соединение восьми процессов. Сопроводительный процесс поддерживает работу других процессов как единое целое с четкой определенной целью и способствует успеху и качеству проекта.



Рис. 2.8. Структура стандарта ИСО/МЭК 12207-95

Сопроводительный процесс используется, как правило, другими процессами. Сопроводительные процессы:

1. Процесс документирования. Определяет действия для записи информации, происходящие во время жизненного цикла.

2. Процесс управления конфигурацией. Определяет действия по управлению конфигурацией.

3. Процесс гарантии качества. Определяет действия, обеспечивающие гарантию того, что программные продукты соответствуют определенным для них требованиям и придерживаются установленных планов.

4. Процесс верификации. Определяет действия (для заказчика, поставщиков или независимой стороны) по верификации программных продуктов различной глубины, в зависимости от конкретного проекта.

5. Процесс аттестации. Определяет действия (для заказчика, поставщиков или независимой стороны) для аттестации программных продуктов.

6. Процесс общего обзора. Определяет действия оценки статуса и качества продуктов. Этот процесс может быть использован любыми из двух сторон, где одна сторона (проверяющая) проверяет другую сторону (проверяемую).

7. Процесс аудита. Определяет действия для установления согласия с требованиями, планами и контрактами. Этот процесс может быть использован двумя сторонами, где одна сторона (аудитор) проводит аудит программных продуктов или действий другой стороны (подлежащей аудиту).

8. Процесс разрешения проблем. Определяет действия для анализа и устранения проблем (включая несогласие), природа или источник которых обнаружены в течение разработки, эксплуатации, сопровождения или других процессов.

Организационные процессы. Раздел 7 ИСО/МЭК 12207-95 состоит из четырех процессов. Они используются организацией на верхнем уровне для установления, выполнения и усовершенствования структуры нижнего уровня, построенной на связи процессов жизненного цикла и личностей организации. Обычно они используются вне сферы конкретных проектов и контрактов; однако уроки, извлеченные из этих проектов и контрактов, способствуют усовершенствованию организации. Организационные процессы:

1. Процесс управления. Определяет основные действия по управлению, включая управление проектом в течение процессов жизненного цикла.

2. Процесс инфраструктуры. Определяет основные действия на установления нижележащей структуры процесса.

3. Процесс усовершенствования. Определяет основные действия, выполняемые организацией (заказчиком, поставщиком, разработчиком, эксплуатирующей, сопровождающей организацией или менеджером другого процесса) для установления, оценки, проверки и совершенствования процессов жизненного цикла.

4. Процесс обучения. Определяет действия по обеспечению соответственно подготовленного персонала.

Кратко рассмотрим перечисленные выше процессы.

5.1. Приобретение.

Процесс приобретения содержит действия и задачи заказчика. Процесс начинается с определения потребностей в приобретении системы или программных продуктов, продолжается подготовкой и выпуском заявок на предложение, выбором поставщика и управлением процесса приобретения посредством принятия программного продукта.

Заказчик управляет процессом приобретения на уровне проекта в соответствии с процессом управления (7.1), который подтверждается примерами в этом процессе, закладывает инфраструктуру в процессе, согласно процессу инфраструктуры (7.2); подгоняет процесс для проекта, согласно процессу подгонки и управляет процессом на организационном уровне, согласно процессу усовершенствования (7.3).

Перечень действий. Этот процесс содержит следующие действия.

5.1.1. Инициализация.

5.1.2. Подготовка заявки для предложения.

5.1.3. Подготовка контракта и корректировка.

5.1.4. Поиск поставщиков.

5.1.5. Принятие и заключение.

5.1.1. Инициализация.

Это действие состоит из следующих основных задач.

Заказчик начинает процесс приобретения, выявляя идею или необходимость приобретения, разработки или усиления системы или программного продукта.

Заказчик должен определить и анализировать системные требования. Системные требования должны включать безопасность, надежность и другие критичные требования, касающиеся проектирования, тестирования и согласованными со стандартами и процедурами. Если заказчик приглашает поставщика для исполнения анализа требований системы, он должен одобрить проанализированные требования. Заказчик может выполнить установление требований к программному продукту самостоятельно или может пригласить поставщика для выполнения этой задачи. Процесс разработки (5.3) может быть использован для выполнения задач, указанных выше.

Заказчик будет обсуждать варианты приобретения, анализируя риск при каждом варианте. Варианты включают:

- приобретение готового к использованию программного продукта, удовлетворяющего требованиям;
- разработку программного продукта самостоятельно;
- разработку программного продукта через контракт;
- комбинацию перечисленных выше вариантов;
- усовершенствование существующего программного продукта.

Заказчик должен подготовить план приобретения, определяющий требования к планируемому использованию системы, тип контракта, который будет использован, обязанности привлеченных организаций, концепцию поддержки, которая будет использована, и риски, принимаемые во внимание, методы управления рисками. План должен быть зарегистрирован и выполнен.

Заказчик должен определить и зарегистрировать стратегию приемки и условия (критерии).

5.1.2 Подготовка заявки для предложения (тендера). Эта деятельность содержит следующие задачи.

Заказчик должен зарегистрировать требования приобретения (например, заявку на участие), содержание которых зависит от варианта, выбранного при выполнении предыдущего действия. Документация должна включать требования системы, формулировку работы, инструкции для покупателей, перечень программных продуктов, статьи договора, статьи поддоговора, технические ограничения (например, среда использования).

Заказчик должен определить, какие процессы, действия и задачи стандарта относятся к проекту и должны быть соответственно подогнаны. Особо заказчик должен определить приемлемые сопроводительные процессы и их выполняющие организации, так что поставщики могут в своих целях определить подход к каждому из определенных сопроводительных процессов.

Документация также должна определять точки контракта, на которых будут рассматриваться и проверяться действия поставщика, как часть мониторинга (6.6 и 6.7). Требования по приобретению должны быть даны организации, выбранной для исполнения этих действий.

Остальные действия этого процесса касаются требований к подготовке контракта, его корректировке, выполнению и завершению и здесь не рассматриваются.

5.2. Процесс поставки.

Процесс поставки содержит действия и задачи поставщика. Процесс может быть инициирован, как принятием решения подготовить предложения в ответ на объявленный заказчиком тендер, так и подписанием и вступлением в силу контракта с заказчиком по предоставлению программного продукта. Процесс продолжается идентификацией процедур и ресурсов, необходимых для управления и обеспечения проекта, включая разработку планов проекта и выполнение планов поставки программного продукта заказчику.

Перечень действий. Этот процесс состоит из следующих действий: введения, подготовки ответа, контракта, планирования, исполнения и контроля, обзора и оценки, поставки и заключения.

5.3. Процесс разработки.

Процесс разработки содержит действия и задачи разработчика. Процесс содержит действия для анализа требований, проектирования, программирования, интеграции, тестирования и инсталляции

и приемки, относящихся к программному обеспечению. Он может содержать системные действия, если это оговаривается в контракте.

Разработчик руководит процессом разработки на уровне проекта, согласно процессу управления (7.1), который сопровождается примерами в этом процессе; представляет инфраструктуру процесса, согласно процессу инфраструктуры (7.2); доводит процесс для проекта, следуя процессу доводки, и руководит процессом на организационном уровне, следуя процессу усовершенствования (7.3).

Перечень действий. Этот процесс состоит из следующих действий.

5.3.1 Реализация процесса. Это действие состоит из следующих задач:

Если не оговорено в контракте, разработчик должен определить или выбрать модель жизненного цикла программного средства в соответствии с областью, важностью и сложностью проекта. Действия и задачи этого процесса должны быть выбраны и отображены в модели жизненного цикла. Эти действия и задачи могут перекрываться или взаимодействовать и могут быть исполнены итеративно или рекурсивно.

Разработчик должен выполнять следующее:

- документировать результаты в соответствии с процессом документирования (раздел 6.1);
- разместить результаты (выходы) в процессе конфигурации (6.2) и выполнить контроль изменений в соответствии с этим;
- документировать и разрешить проблемы и несоответствия, найденные в программном продукте и задачах в соответствии с процессом разрешения проблем (6.8);
- другие сопроводительные процессы (6), как определено в контракте.

Разработчик должен выбрать, довести и использовать внутренние стандарты, методологии, процедуры и языки программирования (если это не оговорено в контракте), которые должны быть зарегистрированы, приемлемы и установлены организацией для исполнения действий Процесса разработки и сопроводительными процессами.

Разработчик должен разработать планы управления действиями в процессе разработки. Планы должны включать конкретные стандарты, методы, инструментальные средства, действия и обязанности, связанные с разработкой всех требований, включая надежность и безопасность. Эти планы должны быть зарегистрированы и исполнены.

Официально непоставляемые элементы могут быть использованы в разработке или сопровождении программного средства. Однако

должна быть гарантия того, что эксплуатация и поддержка программного средства после его поставки заказчику не зависят от таких элементов. Другими словами, эти элементы становятся официально поставляемыми.

5.3.2. Анализ системных требований. Это действие состоит из следующих задач, которые разработчик должен исполнить или поддерживать, как требуется по контракту.

Предполагаемое использование разрабатываемых программных средств должно быть проанализировано на предмет определения системных требований. Спецификация системных требований должна описывать: функции и возможности системы; требования надежности, безопасности, разработки, интерфейсов, эксплуатации и поддержки; ограничения проекта и квалификационные требования. Спецификации системных требований должны быть зарегистрированы.

Системные требования должны быть оценены с позиций следующих критериев: прослеживаемость потребностей заказчика, согласованность с потребностями заказчика, тестируемость, выполнимость разработки архитектуры системы (системного проектирования), осуществимость эксплуатации и поддержки.

5.3.3. Системное проектирование. Это действие состоит из следующих задач, которые разработчик должен выполнить или поддерживать, как требует контракт.

Должна быть представлена архитектура верхнего уровня системы. Архитектура должна определять элементы аппаратного, программного обеспечения и ручные операции. Должна быть гарантия того, что все системные требования полностью распределены среди элементов. Эти элементы должны быть впоследствии определены как элементы аппаратной конфигурации (ЭАК), элементы конфигурации программного средства (ЭКПС) и ручные операции. Архитектура системы и системные требования, распределенные между элементами аппаратной конфигурации, конфигурации ПС и ручными операциями, должны быть зарегистрированы.

Архитектура системы и требования для элементов конфигурации и ручных операций должны быть оценены в соответствии со следующими критериями:

- прослеживаемость системных требований;
- согласованность с системными требованиями;
- соответствие стандартам и используемым методам проектирования;
- реализация требований в элементах конфигурации ПС;
- осуществимость эксплуатации и поддержки.

5.3.4. Анализ требований к программному средству. Для каждого ЭКПС это действие состоит из следующих задач, которые разработчик должен исполнить.

Разработчик должен представить и зарегистрировать требования к программному средству, включая спецификации следующих характеристик качества:

- функциональные спецификации и возможности, включая физические характеристики, условия окружающей среды, при которых выполняется и реализуется программное средство;
- внешний интерфейс ЭКПС;
- квалификационные требования;
- спецификации надежности;
- спецификации безопасности;
- требования определения данных и требования к базам данных;
- требования по установке и приемке поставляемого программного средства на месте эксплуатации и сопровождения;
- документация пользователя;
- требования по эксплуатации пользователя;
- требования по сопровождению.

Руководство по определению характеристик качества может быть найдено в ИСО/МЭК 9126-93.

Разработчик должен оценить следующие требования к программному средству:

- прослеживаемость системных требований;
- внешняя согласованность с системными требованиями;
- внутренняя согласованность;
- тестовое покрытие требований;
- тестируемость;
- выполнимость проектирования;
- осуществимость эксплуатации и сопровождения.

Разработчик должен руководить общим обзором в соответствии с процессом 6.6. После успешного завершения обзора должна быть представлена основа для требований к ЭКПС.

5.3.5 Проектирование программного обеспечения. Для каждого ЭКПС это действие состоит из следующих задач, которые разработчик должен был выполнить.

Разработчик должен преобразовать требования для ЭКПС в архитектуру, описывающую структуру его верхнего уровня и определяющую главные компоненты. Должна быть гарантия того, что требования к ЭКПС полностью распределены между его компонентами и далее уточнены в целях облегчения детального проектирования.

Разработчик должен разработать и зарегистрировать:

- проект верхнего уровня для внешнего взаимодействия с ЭКПС и между компонентами программного средства;
- проект верхнего уровня для баз данных;
- предварительные версии руководства для пользователя;
- предварительные тестовые требования и план интеграции программного средства.

Разработчик должен оценить архитектуру ЭКПС и проекты интерфейса и баз данных:

- прослеживаемость требований к ЭКПС;
- внешняя согласованность с требованиями к ЭКПС;
- внутренняя согласованность между компонентами;
- соответствие методам проектирования и стандартам, которые были использованы;
- выполнимость детализированного проектирования;
- осуществимость эксплуатации и сопровождения.

Разработчик должен руководить общим обзором в соответствии с процессом 6.6.

5.3.6. Детальное проектирование программного средства. Для каждого ЭКПС это действие состоит из следующих задач, которые разработчик должен выполнить.

Разработчик должен выполнить следующее.

1. Разработать детальный проект для каждого ЭКПС. Компоненты программного средства должны быть уточнены на нижних уровнях, содержащих модули, которые могут кодироваться, компилироваться и тестироваться. Должна быть гарантия того, что требования к программному средству полностью локализованы в модулях. Детальный проект должен быть зарегистрирован.

2. Разработать и зарегистрировать детальный проект для внешнего интерфейса ЭКПС между компонентами программного средства и между модулями. Детальный проект интерфейса должен позволять писать код программы без необходимости в дополнительной информации.

3. Разработать и зарегистрировать детальный проект базы данных.

4. Обновить руководство пользователя насколько это необходимо.

5. Определить и зарегистрировать тестовые требования и расписание тестирования модулей. Тестовые требования должны включать испытания программного средства на пределе требований.

6. Обновить тестовые требования и расписание интеграции программного средства.

7. Оценить детальный проект программного средства и тестовые требования с точки зрения следующих критериев:

- прослеживаемость требований ЭКПС;
- внешняя согласованность с архитектурой проекта;
- внутренняя согласованность между компонентами и модулями;
- соответствие методам проектирования и используемым стандартам;
- выполнимость тестирования;
- выполнимость эксплуатации и сопровождения.

Разработчик должен руководить объединением, согласно процессу 6.6.

5.3.7. Программирование и отладка. Для каждого ЭКПС это действие состоит из следующих задач, которые должен выполнить разработчик. Разработчик должен разработать и зарегистрировать следующее:

- каждый модуль программного средства и базы данных;
- процедуры тестирования и данные для тестирования каждого модуля и базы данных.

Разработчик должен тестировать каждый модуль ПС и базы данных, убеждаясь в том, что они удовлетворяют требованиям. Результаты тестирования должны быть зарегистрированы.

Разработчик должен обновить руководство пользователя, тестовые требования и расписание интеграции ПС, оценить код ПС и результаты теста в соответствии с критериями, приведенными ниже:

- прослеживаемость требований ЭКПС и проекта;
- внешняя согласованность с требованиями ЭКПС и архитектурой проекта;
- внутренняя согласованность между требованиями модулей;
- тестирование модулей;
- соответствие методам кодирования и используемым стандартам;
- выполнимость интеграции ПС и тестирования;
- выполнимость эксплуатации и сопровождения.

5.3.8. Интеграция программного средства. Для каждого ЭКПС это действие состоит из следующих задач, выполняемых разработчиком.

Разработчик программного средства должен выполнить следующее.

Разработать план интеграции для объединения модулей ПС и компонентов в ЭКПС. План должен включать требования, процедуры, данные, ответственность и расписание.

Объединить модули ПС и компоненты. Интеграция и результаты теста должны быть зарегистрированы.

Обновить руководство пользователя, если это требуется.

Разработать и задокументировать для каждого квалификационного требования ЭКПС полный набор тестов, ситуаций (вход, выход, критерии тестирования) и процедуры тестирования для управления квалификационным тестированием ПС.

Оценить план интеграции, результаты проектирования, кодирования, тестирования и руководства пользователя с точки зрения критериев, приведенных ниже:

- прослеживаемость системных требований;
- внешняя согласованность с системными требованиями;
- внутренняя согласованность;
- тестирование требований ЭКПС;
- соответствие используемым стандартам и методам тестирования;
- соответствие с ожидаемыми результатами;
- выполнимость квалификационного тестирования ПС;
- выполнимость эксплуатации и сопровождения.

Руководить общим обзором в соответствии с процессом 6.6.

5.3.9. Квалификационное тестирование программного обеспечения. Для каждого ЭКПС это действие состоит из следующих задач, которые должен выполнить разработчик.

Руководить квалификационным тестированием в соответствии с квалификационными требованиями, определенными для ЭКПС. Должно быть гарантировано, что реализация каждого требования к ПС полностью протестирована. Результаты квалификационного тестирования должны быть зарегистрированы.

Оценить результаты проектирования, кодирования, тестирования и руководство пользователя в соответствии с приведенными ниже критериями:

- тестирование требований к ЭКПС;
- согласованность с ожидаемыми результатами;
- выполнимость системной интеграции и тестирования;
- выполнимость эксплуатации и сопровождения.

Поддерживать аудит в соответствии с процессом 6.7. Результаты аудита должны быть зарегистрированы. Если разрабатываются или интегрируются ПС и аппаратное обеспечение, аудит может быть отложен до квалификационного тестирования системы.

После успешного завершения аудита, если предписано, разработчик должен:

- обновить и подготовить к поставке ПС для системной интеграции, квалификационного тестирования системы, инсталляции или поддержки приема ПС;

- представить основную линию проектирования и кодирования ЭКПС;

5.3.10. Системная интеграция. Это действие состоит из следующих задач, которые должен выполнить разработчик.

ЭКПС должен быть интегрирован с ЭАК руководством по эксплуатации и другими системами в единую систему. Составляющие должны быть протестированы на соответствие требованиям. Интеграция и результаты тестирования должны быть зарегистрированы.

Для каждого квалификационного требования к системе должны быть разработаны и зарегистрированы полный набор тестов, ситуаций (входных, выходных, критериев тестирования), процедур тестирования. Разработчик должен гарантировать, что интегрированная система готова для квалификационного тестирования.

Интегрированная система должна быть оценена в соответствии с приведенными ниже критериями:

- требования к системе полностью охвачены тестами;
- приемлемость используемых методов и стандартов тестирования;
- согласованность с ожидаемыми результатами;
- выполнимость квалификационного тестирования системы;
- выполнимость эксплуатации и сопровождения.

5.3.11. Квалификационное тестирование системы. Это действие состоит из следующих задач, выполняемых разработчиком.

Квалификационное тестирование системы должно проводиться в соответствии с квалификационными требованиями, определенными для системы. Должно быть гарантировано, что каждого требования к системе протестировано полностью и система готова к поставке. Результаты квалификационного тестирования должны быть зарегистрированы.

Система должна быть оценена в соответствии с приведенными ниже критериями:

- требования к системе полностью охвачены тестами;
- ожидаемые результаты подтверждены;
- выполнимость эксплуатации и сопровождения.

Разработчик должен поддерживать аудит в соответствии с процессом 6.7. Результаты аудита должны быть зарегистрированы. Этот пункт не применяется к таким ЭКПС, для которых аудит был выполнен ранее.

После успешного завершения аудита, если предписано, разработчик должен обновить и подготовить к поставке ЭЖПС для инсталляции ПС и приемки.

5.3.12. Инсталляция ПС. Это действие состоит из следующих задач, выполняемых разработчиком.

Разработчик должен разработать план инсталляции ПС в наменченную среду. Ресурсы и информация, необходимые для установки ПС, должны быть определены и доступны. Разработчик должен помогать заказчику при установке. После того, как ПС установлено в существующую систему, разработчик должен поддерживать некоторые, параллельно выполняемые действия. План установки должен быть зарегистрирован.

Разработчик должен установить ПС в соответствии с планом установки. Должно быть гарантировано, что ПС и базы данных инициализируются, функционируют и прекращают работу, как указано в контракте. Процесс установки и результаты должны быть задокументированы.

5.3.12. Поддержка приемки ПС. Разработчик должен поддерживать процесс приемки заказчиком и тестирование ПС. Приемка и тестирование должны основываться на общем обзоре, аудите, квалификационном тестировании, квалификационном тестировании системы (если оно выполнялось). Результаты приемки и тестирования должны быть задокументированы.

5.4. Процесс эксплуатации.

Процесс эксплуатации состоит из действий и задач того, кто эксплуатирует разработанное ПС. Процесс включает эксплуатацию ПС и поддержку пользователей. Поскольку эксплуатация ПС является интеграционной составляющей эксплуатации системы, действия и задачи этого процесса относятся и к системе.

Оператор управляет процессом эксплуатации на уровне проекта, следуя процессам управления (7.1), инфраструктуры (7.2), процессам подгонки и усовершенствования (7.3).

Этот процесс состоит следующих задач: выполнения процесса, тестирования, эксплуатации системы и поддержки пользователей.

5.5. Процесс сопровождения.

Процесс сопровождения состоит из действий и задач лица, выполняющего сопровождение. Этот процесс начинается, когда необходима модификация ПС из-за допущенных ошибок, неучтенных проблем, необходимости усовершенствования или адаптации кода ПС и соответствующей документации. Его цель — модифицировать

существующее ПС, сохранив его целостность. Этот процесс включает распространение и замену ПС. Процесс завершается заменой ПС.

Процесс состоит из следующих действий: реализации процесса, анализа проблем и модификации, реализации модификации, приемки, распространения, замены ПС.

В силу ограниченности объема данного учебного пособия остальные процессы жизненного цикла ПС, установленные в ИСО/МЭК 12207-95, рассматривать не будем.

2.5. Сравнительный анализ жизненных циклов программных средств

В первую очередь, проведем анализ понятия “жизненный цикл”, употребляемый нами довольно часто. Согласно ГОСТ 34.003-90, “жизненный цикл АС – совокупность взаимосвязанных процессов создания и последовательного изменения состояния АС от формирования исходных требований к ней до окончания эксплуатации и утилизации комплекса средств автоматизации АС”. При рассмотрении жизненного цикла ПС, регламентированного 34-й системой стандартов, было приведено такое же определение. При этом подразумевалось, что под понятием автоматизированная система понимается программное обеспечение. Тогда, более точным будет следующее определение. Жизненный цикл ПС – совокупность взаимосвязанных процессов создания и последовательного изменения состояния ПС от момента формирования исходных требований к нему до полной его утилизации.

По смыслу и объему это определение полностью согласуется с определением модели жизненного цикла, данное в проекте стандарта ИСО/МЭК 12207-1-95. Модель жизненного цикла – ПС-структура, содержащая процессы, действия и задачи, касающиеся разработки, эксплуатации и поддержки программных продуктов, охватывающая ЖЦ средства от определения требований к нему до окончания его использования.

В документе DO-178B нет четкого определения понятия жизненного цикла программного обеспечения. Тем не менее, проведя анализ определенных в нем этапов жизненного цикла ПС, можно сделать вывод, что, по сути, оно аналогично определению, используемому в ГОСТ 34.003-90. Однако жизненный цикл, установленный в DO-178B, не содержит аналогичных этапов: “Выполнение работ в соответствии с гарантийными обязательствами” и “Послегарантийного обслуживания”, определенных в ГОСТ 34.601-90.

Можно сделать вывод, что современное представление ЖЦ ПС отражено в проекте стандарта ИСО/МЭК 12207-95, но представляет

ЖЦ не только как процесс разработки ПС, но и как процесс эксплуатации и сопровождения ПС.

ГОСТ 19.102-77 и ГОСТ 34.601-90 не охватывают всего жизненного цикла ПС. Кроме того, ГОСТ 19.102-77 устарел, его требования не могут быть выполнены в полном объеме (например, требование обязательной передачи программы в фонд алгоритмов и программ). Основное отличие ИСО/МЭК 12207-95 от ГОСТ 19.102-77, ГОСТ 34.601-90, DO-178В заключается также в том, что первый из них устанавливает общий каркас для любого типа моделей жизненного цикла ПС, а остальные устанавливают более конкретные схемы. Схема ГОСТ 19.102-77 не предполагает использование высокопроизводительных моделей ЖЦ, широко применяемых в современной программной технике. Из отечественных стандартов предпочтительнее использовать схему ГОСТ 34.601-90, которая охватывает практически все современные модели ЖЦ ПС. DO-178В признает, что приемлемы различные модели ЖЦ для разработки ПС и подчеркивает, что выбранный жизненный цикл должен быть определен на этапе планирования проекта.

Кроме того, выполнение требований ГОСТ 19.102-77 и ГОСТ 34.601-90 не гарантирует качества ПС, так как в них явно не определены процессы обеспечения качества разрабатываемого ПС. В этом отношении более прогрессивны проект стандарта ИСО/МЭК 12207-95 и документ DO-178В, руководящие принципы которых удовлетворяют целям стандартов ИСО 9000-3-91 “Стандарты в области административного управления качеством и обеспечения качества. Часть 3. Руководящие положения по применению стандарта ИСО 9001 при разработке, поставке и обслуживанию ПО” и МЭК 65А (Secretariat) 122 (1991 г.) “ПО для компьютеров, используемых в промышленных системах безопасности”.

Общая структура процессов ЖЦ во всех рассматриваемых выше стандартах и документах совпадает, за исключением DO-178В. Структура ИСО/МЭК 12207-95, ГОСТ 19.102-77, ГОСТ 34.601-90 представляют собой иерархию, содержащую три уровня: стадия (процесс), этапы (действия), содержание работ (задачи). Различия этих стандартов заключаются в наполнении этой структуры. Точки контроля жизненного цикла привязаны к указанным трем уровням структуры.

DO-178В является преимущественно этапно-ориентированным документом. В нем для каждого этапа определены цели и способы достижения этих целей. В нем имеется описание данных этапов жизненного цикла, которые показывают, что цели были достигнуты.

Кроме того, в DO-178В явно определена взаимосвязь между этапами жизненного цикла системы и этапами жизненного цикла ПС. Условия сбоев, связанных с функциями, которые выполняют ПС, описаны в ходе выполнения этапа оценки безопасности системы. Это является основой для установления уровня ПС. Уровень программного обеспечения основывается на вкладе, вносимом ПС в набор потенциальных состояний отказов системы.

ГОСТ 34.601-90 в качестве первой и обязательной стадии создания предусматривает формирование требований пользователя. В ИСО/МЭК 12207-95 эта стадия соответствует двум первым действиям процесса приобретения (5.1). Основным результатом этой стадии являются формально сформулированные и зарегистрированные требования пользователя к разрабатываемому ПС. Такой подход позволяет на последующих стадиях проводить оценку качества ПС ГОСТ 28195-89 (ГОСТ Р ИСО/МЭК 9126-93) на основе идентифицированных требований пользователя.

В ГОСТ 19.102-77 формулировка требований пользователя опущена до самого нижнего уровня иерархии (стадия “Технический проект”, этап 3 “Разработка и утверждение технического задания”, работа “Определение требований к программе”). В недавнем прошлом эта работа, зачастую, совсем исключалась из проекта, так как этот стандарт допускает исключать стадии, этапы работ и их содержание.

Стадии “Разработка концепции”, “Техническое задание”, “Эскизный и технический проект” ГОСТ 34.601-90 в основном соответствуют процессу разработки (5.3) ИСО/МЭК 12207-95 и частично — стадиям “Техническое задание”, “Эскизный и технический проект”, “Рабочий проект” ГОСТ 19.102-77. Детализация этапов и содержания работ на этих стадиях по ГОСТ 19.102-77 представляются на современном уровне недостаточными для оценки качества ПС. Детализация этапов (действий) и содержания работ (задач) по ГОСТ 34.601-90 и ИСО/МЭК 12207-1-95 для этой стадии (процесса) в целом хорошо согласуются между собой, хотя в первом из этих документов содержание работ в большей мере учитывает реальности РФ, а во втором дана большая детализация. Так как содержание работ ГОСТ 34.601-90 вынесено в справочное приложение к этому стандарту, то в случае необходимости в работы можно включать задачи ИСО/МЭК 12207-95.

Последние две стадии “Ввод в действие” и “Сопровождение” ГОСТ 34.601-90 связаны с процессами поставки (5.2), эксплуатации (5.4) и сопровождения (5.5) ИСО/МЭК 12207-95. При этом большинство

этапов и работ, выполняемых на этих стадиях, применяются к автоматизированным системам, а не к программному обеспечению (например, этап 7.4 “Строительно-монтажные работы”). Содержание работ этапа “Внедрение” ГОСТ 19.102-77 не точно соответствует сегодняшним реальностям, о чем уже говорилось выше.

Что касается сравнения документа DO-178В с ИСО/МЭК 12207-95, ГОСТ 34.601-90 в отношении процессов ЖЦ, то в силу того, что они имеют разную структуру, анализ несколько затруднен. Можно только сказать, что цели этапа планирования соответствуют в некоторой степени целям стадии 2 ГОСТ 34.601-90, этап разработки идентичен процессу разработки ИСО/МЭК 12207-95, а интегрированный этап – поддерживающим процессам ИСО/МЭК 12207-95.

Вопросы обеспечения качества, верификации, подтверждения почти не отражены в ГОСТ 19.102-77 и ГОСТ 34.601-90. В проекте ИСО/МЭК 12207-95 эти вопросы охвачены поддерживающими процессами обеспечения качества, верификации, подтверждения, общего обзора, аудита, разрешения проблем. В DO-178В эти вопросы отрабатываются в процессе выполнения интегрированного этапа. Непосредственное отношение к данным вопросам имеют работы по тестированию ПС, почти не отраженные в ГОСТ 19.102-77 и ГОСТ 34.601-90, но которым уделяется достаточное внимание в задачах основных процессов ИСО/МЭК 12207-95 и DO-178В.

ГЛАВА 3

ДОКУМЕНТАЦИЯ И ЕЕ РОЛЬ В ОБЕСПЕЧЕНИИ КАЧЕСТВА ЖИЗНЕННОГО ЦИКЛА ПС

3.1. Документация и ее роль в обеспечении качества

Документирование ПС определяется стандартами ГОСТ серии 19, 34, ГОСТ Р ИСО/МЭК 8631-94 “ИТ. Программные конструктивы и условные обозначения для их представления”, ГОСТ Р ИСО 9127-94 “Системы обработки информации. Документация пользователя и информация на упаковке для потребительских пакетов”, ГОСТ Р ИСО/МЭК ТО 9294-93 “ИТ. Руководство по управлению документированием ПО”. Полный перечень нормативных документов на программную документацию приведен в документе “Методика экспертизы программной документации”, находящегося на стадии утверждения.

ГОСТ Р ИСО/МЭК ТО 9294-93 является одним из основных стандартов в данной области и представляет собой руководство по документированию ПС для тех руководителей, которые отвечают за разработку ПС. Данное руководство предназначено для помощи руководителям в обеспечении эффективного проведения документирования в организациях. Данный стандарт направлен на определение стратегий, стандартов, процедур, ресурсов и планов, которыми должны заниматься сами руководители для того, чтобы эффективно управлять документированием ПС.

Руководство по управлению документированием ПС предназначено для применения ко всем типам программных средств — от простейших программ до наиболее сложных программных систем, охватывающих все типы программной документации, относящейся ко всем стадиям жизненного цикла ПС.

Принципы управления документированием программного средства одинаковы для проектов любого объема. Для небольших проектов значительную часть положений, приведенных в данном стандарте, можно не применять, но принципы остаются теми же. Руководители могут адаптировать данные рекомендации для своих конкретных потребностей.

Эффективность выполнения руководящей роли можно рассматривать как основанную на следующих трех элементах.

1. *Руководящая обязанность по документированию* требует признания того, что программная документация важна и что ее следует планировать, описывать, проверять, утверждать, выпускать, распространять и сопровождать.

2. *Руководящая поддержка обязанностей персонала по документированию* требует руководства и стимулирования персонала при проведении требуемого документирования и обеспечения его ресурсами для содействия в данной работе.

3. *Признаки руководящих обязанностей и поддержки* требует обеспечить:

- опубликованные официальные отчеты о стратегии документирования;
- стандарты и руководства, определяющие все аспекты документирования ПС;
- опубликованные процедуры документирования;
- выделение соответствующих ресурсов для документирования;
- планирование документирования, осуществляемое как неотъемлемая часть процесса разработки ПС;
- постоянную проверку, осуществляемую для обеспечения соответствия со стратегией, стандартами, процедурами и планами по документированию.

В ГОСТ Р ИСО/МЭК ТО 9294-93 программная документация рассматривается как имеющая следующие шесть основных функций.

1. *Информация для управления.* Во время разработки ПС администрации необходимо оценивать ход работы, возникающие проблемы и вероятности развития процесса разработки ПС. Периодические отчеты, согласно которым проверяют ход работ по графику и представляют планы на следующий период, обеспечивают контрольные механизмы и обзор проекта.

2. *Связь между задачами.* Большинство проектов разработки ПС разделяется на задачи, выполняемые различными группами: специалистами в предметной области, аналитиками, проектировщиками, специалистами по обеспечению качества и др. Этим группам, а также персоналу в пределах группы, необходимы средства общения друг с другом, обеспечивающие информацию, которую можно воспроизводить, распространять и на которую можно ссылаться. Большинство методологий разработки ПС устанавливают официальные документы для связи между задачами. Например, аналитики представляют официальные спецификации требований проектировщикам, а они, в свою очередь, официальные проектные спецификации программистам.

3. Обеспечение качества. Требуется документация разработки и документация продукции для выполнения задач, связанных с обязанностями по обеспечению качества ПС.

4. Инструкции и справки. Документация, требующаяся операторам, пользователям, руководителям и другим заинтересованным лицам для того, чтобы понимать и использовать ПС.

5. Сопровождение ПС. Специалистам, сопровождающим ПС, требуется детальное описание ПС, такое, чтобы они могли локализовать и корректировать ошибки и модернизировать или изменять ПС соответствующим образом.

6. Исторические справки. Документация, требуемая в качестве исторической справки по проекту. Данная документация может помочь в переносе и переводе ПС в новое окружение.

Рассмотрим стратегии документирования. Стратегии документирования, подготовленные и контролируемые руководителем организации, обеспечивают руководящими положениями ответственных лиц, принимающих решения на всех нижних уровнях. Стратегия обеспечивает главное направление, но не дает рекомендаций, что делать и как это делать. Из-за существенной роли, которую играет документация на всех стадиях ЖЦ ПС, стратегия должна быть официально утверждена, доведена до каждого исполнителя проекта. Официальная стратегия устанавливает дисциплину, требуемую для эффективного документирования ПС.

Стратегия должна поддерживать основные элементы эффективного документирования:

- требования документации охватывают весь жизненный цикл ПС;
- документирование должно быть управляемым;
- документация должна соответствовать читательской аудитории;
- работы по документированию должны быть объединены в общий процесс разработки ПС;
- должны быть определены и использованы стандарты по документированию;
- должны быть определены средства поддержки.

Внутри организации, помимо стратегии, должны быть приняты стандарты и руководства для:

- модели жизненного цикла ПС;
- типов и взаимосвязей документов;
- содержания документа;
- качества документа;
- форматов документа;
- обозначения документа.

Данные стандарты и руководства будут определять, как следует выполнять задачи документирования, и будут обеспечивать критерии для оценки полноты, полезности и соответствия программной документации, создаваемой в организации.

Большинство стандартов и руководств выдают рекомендации, которые применимы на общем уровне. Зачастую будут требоваться управленческие решения для адаптации общих рекомендаций к конкретным проектам. Применение стандартов, распространяющихся на организацию документирования, облегчит руководителям проекта определение следующих вопросов:

- какие типы документов требуются?
- каков объем представляемой документации?
- что документы содержат?
- какой уровень качества будет достигнут?
- где документы будут созданы?
- как документы будут хранить, сопровождать и обращать?

В организации должны быть установлены процедуры для применяемых стратегий документирования. Процедуры определяют последовательность документирования: планирование, подготовку, конфигурационное управление, проверку, утверждение, производство, хранение, дублирование, распространение и модернизацию, продажу. Кроме того, процедуры должны определять контрольные пункты и методы обеспечения качества.

Основными ресурсами, требуемыми для документирования, являются следующие: персонал, средства, финансирование.

Персонал. Для процесса разработки ПС необходимы люди со знанием программирования, предметной области, документирования, проектировщики и другие. Важно, чтобы все специалисты, участвующие в разработке проекта, были обучены методам документирования, полностью понимали и выполняли свою роль в документировании.

Средства. Важно предусмотреть обеспечение задач документирования соответствующими и подходящими средствами. Инструментальные средства полезны для подготовки и контроля документации. Они могут быть применены для повышения эффективности многих процессов документирования и использования стандартов данной организации, распространяющихся на документирование.

Финансирование. Важно, чтобы стоимость документирования определяли как отдельную статью бюджета, так как она нередко составляет значительную часть стоимости разработки ПС.

Процесс документирования, как уже отмечалось выше, должен планироваться. План документирования определяет, что должно быть

сделано, как это должно быть сделано, когда это должно быть сделано и кто это должен делать.

План документирования может быть как частью общего плана проекта, так и отдельным документом. Он должен быть доведен до всех участников проекта. План документирования включает в себя изложение общей структуры документации, типов, содержания, качества, форматов, обозначения, комплектности и хранения документов, их обращения и графика документирования.

График документирования должен распределять время для:

- планирования документов;
- проверки плана и принципов документирования;
- подготовки проектов и проверки их на техническую точность, полноту и соответствие;
- редактирования при внесении изменений, появившихся при проверке;
- проведения согласования;
- перевода;
- распространения.

Планирование документирования следует начинать заранее, и план необходимо проверять на всем протяжении проекта. Подобно любому плану, план документирования отражает намечаемые действия и является объектом для необходимых изменений. В проекте должны быть предусмотрены регулярные проверки результативности изменений в плане.

Более подробно рассмотрим стандарты и руководства, принимаемые в организации.

Выбор модели жизненного цикла программного средства. Как уже отмечалось выше, существует ряд моделей жизненного цикла ПС с отличающейся терминологией для различных стадий. С точки зрения программной документации, нет никакой разницы в выбранной модели ЖЦ до тех пор, пока стадии и соответствующая им документация четко определены, спланированы и выполнимы для любого конкретного программного проекта. Руководители должны выбрать соответствующую модель ЖЦ и гарантировать, чтобы ее применяли в данной организации. При этом руководители должны убедиться, что выбранные стадии и соответствующие задачи помогут им контролировать ход разработки ПС.

Проведем краткий анализ жизненных циклов программных средств, описанных выше.

В рамках стандарта ИСО/МЭК 12207-95 процедуры документирования решаются при вызове каким-либо процессом ЖЦ процесса документирования.

Процесс документирования — это процесс записи информации, получаемой в процессе жизненного цикла или деятельности. Процесс состоит из набора действий, таких как планирование, проектирование, производство, редактирование, распространение и сопровождение документов, необходимых всем заинтересованным лицам проекта: менеджерам, инженерам и пользователям системы или ПС.

Процесс документирования состоит из следующих действий: реализации, проектирования и разработки, производства и сопровождения.

Реализация. Это действие подразумевает следующее.

Планирование, определяющее, какие документы должны быть выпущены во время жизненного цикла ПС, что должно быть разработано, задокументировано и реализовано. Для каждого установленного документа должно быть указано следующее:

- название или имя;
- цель;
- назначение;
- процедуры и ответственность по вводу, разработке, осмотру, модификации, утверждению, производству, хранению, распространению, сопровождению и конфигурированию;
- расписание промежуточной и конечной версий.

Проектирование и разработка. Это действие состоит из следующих задач.

1. Каждый документ должен быть спроектирован в соответствии с применяемыми стандартами по формату, содержанию, нумерации страниц, размещению рисунков/таблиц, отметке о приоритете/секретности, упаковке и другим пунктам.

2. Источник и соответствие входных данных для документов должны быть гарантированы. Должны быть использованы автоматические средства документирования.

3. Подготовленные документы должны быть просмотрены и отредактированы с точки зрения формата, технического содержания и стиля представления в соответствии со стандартами по документации. Они должны быть утверждены ответственным за их выпуск.

Производство. Это действие состоит из следующих задач.

1. Документы должны быть произведены и упакованы. В соответствии с планом ими должны быть обеспечены все заинтересованные стороны. Производство и распространение документов может быть выполнено в бумажном, электронном виде. Исходные материалы должны храниться в соответствии с условиями записи, секретности, сопровождения и резервных копий проекта.

2. Контроль должен быть представлен в соответствии с процессом управления конфигурированием.

Сопровождение. Это действие подразумевает следующее.

Задачи, необходимые для выполнения при модификации существующего ПО, должны быть исполнены.

Стадия “Рабочая документация” ГОСТ 34.601-90 частично соответствует привлечению процесса документирования процессов разработки по ИСО/МЭК 12207-95. Название данной стадии не точно отражает ее содержание, так как она содержит этап 6.2 — “Разработка или адаптация программ”. Кроме того, процесс документирования выполняется на стадиях эскизного и технического проекта. По ГОСТ 19.102-77 этап разработки программной документации появляется только на стадии “Рабочий проект”. Такое позднее начало документирования процессов ЖЦ не в состоянии обеспечить ни требуемого качества документации, ни требуемого качества ПС.

Таким образом, терминология и концепция в области документирования ИСО/МЭК 12207-95 представляются более удачными. Можно рекомендовать при использовании ГОСТ 34.601-90 для этапов 4.2, 5.2, 5.3, 6.1 учитывать в содержании работ задачи, определенные в ИСО/МЭК 12207-95 для процесса документирования.

Определение типов и содержания документов. Программные документы можно представить разделенными на три категории:

- документация разработки;
- документация продукции;
- документация управления проектом.

Данная схема не является исчерпывающей и окончательной, но служит контрольной таблицей основных типов программных документов, которые руководители должны предусмотреть, когда определяют стандартные типы своих документов.

Документация разработки. Документы, описывающие процесс разработки ПС, определяют требования, которым должно удовлетворять ПС, определяют проект программного обеспечения, определяют, как его контролируют и как обеспечивают его качество. Документация разработки также включает в себя подробное описание ПС (программную логику, программные взаимосвязи, форматы и хранения данных и т. д.). Разработка документов преследует пять целей:

а) они являются средством связи между всеми участниками проекта и описывают подробности решений, принятых относительно требований к ПС, проекту, программированию и тестированию;

б) они описывают обязанности группы разработки и определяют, кто, что и когда делает, учитывая роль программного обеспечения, предмета работ, документации, персонала, обеспечивающего качество, и каждого вовлеченного в процесс разработки;

в) они выступают, как контрольные пункты, которые позволяют руководителям оценивать ход разработки (если документы разработки отсутствуют, неполны или устарели, то руководители проекта теряют важное средство для отслеживания и контроля проекта);

г) они образуют основу документации сопровождения ПС, требуемой лицам, сопровождающим ПС, как часть документации продукции;

д) они описывают историю разработки ПС.

Типовыми документами разработки являются:

- анализы осуществимости и исходные заявки;
- спецификации требований и функций;
- проектные спецификации, включая спецификации программ и данных;

– планы разработки, сборки и тестирования ПС;

– планы обеспечения качества, стандарты и графики;

– защитная и текстовая информация.

Документация продукции. Документация продукции обеспечивает информацию, необходимую для эксплуатации, сопровождения, модернизации, преобразования и передачи программной продукции.

Документация продукции преследует три цели:

– обеспечение учебной и справочной информацией для любого, использующего или эксплуатирующего ПС;

– облегчение сопровождения и модернизации ПС программистам, не разрабатывавших ПС;

– помощь при продаже или приемке ПС.

Документация продукции должна включать в себя материалы для следующих типов читателей: пользователей, операторов, сопровождающих программистов. Кроме того, данная документация может включать в себя руководства и материалы для руководителей, вспомогательные материалы, общую информацию.

Типовые документы продукции включают в себя:

– учебные руководства;

– справочные руководства и руководства пользователя;

– руководства по сопровождению ПС;

– брошюры и информационные листки, посвященные продукции.

ГОСТ Р ИСО 9127-94 вводит определение документации пользователя, как документации, которая обеспечивает пользователей информацией, необходимой для установки и прогона ПС и является обязательной в поставке (документация пользователя выполняется в виде одного или нескольких руководств и вкладывается вместе с программным продуктом внутрь упаковки).

Данный стандарт определяет три категории информации:

- обязательная — информация, поставляемая с каждым пакетом;
- условная — информация, поставляемая с каждым пакетом, для которого она необходима;
- факультативная — информация, поставляемая с каждым пакетом, по усмотрению изготовителя или торгующей организации.

Назначением документации пользователя является обеспечение конечного пользователя достаточной информацией для ясного понимания:

- цели, функций и характеристик ПС;
- того, как ввести в действие и использовать ПС;
- договорных прав и обязанностей.

Документация пользователя должна включать в себя справочную документацию для повседневного использования программы. Дополнительно может быть выполнена учебная документация. В качестве справочной документации выступают: обозначение пакета, компоненты пакета, функциональное описание ПС, ввод в действие ПС, использование ПС, техническая информация о ПС, тестирование, договорная информация, словарь, указатель, замечания конечных пользователей.

Таким образом, ГОСТ Р ИСО 9127-94, не имея формальных ссылок на ГОСТ Р ИСО/МЭК ТО 9294-93, фактически дополняет введенное в нем понятие документации продукции.

Документация управления проектом. Документы создаются на основе информации управления проектом, такой как:

- графики для каждой стадии процесса разработки и отчеты об изменениях графиков;
- отчеты о согласованных изменениях ПС;
- отчеты о решениях, связанных с разработкой;
- распределение обязанностей.

Данная документация обеспечивает информацию, относящуюся, с точки зрения руководства, к долговечности продукции.

Определение качества документов. Руководители должны выбирать стандарты, распространяющиеся на уровень качества, соответственно различным типам документов и различным типам проектов и должны определять, как это качество будет достигнуто и поддержано.

Понятия качества, применимые к содержанию, структуре и представлению документации:

- качество содержания можно измерять в элементах точности, полноты и ясности;
- качество структуры, можно измерять легкостью, с которой читатель имеет возможность определить местоположение информации;

— качество представления должно соответствовать типу проекта.

Определение форматов документов. Стандартизированные форматы документов важны для контроля качества документов, для читаемости документов и для облегчения их сопровождения.

Информация может быть представлена в различных форматах, причем форматы могут меняться от проекта к проекту. Форматы зависят от таких факторов, как объем проекта, аудитория, для которой предназначены документы, количество стадий разработки и бюджет документирования. Кроме того, должно быть учтено соображение о том, будут ли документы переводить для международного распространения.

Стандарты и руководства организации, распространяющие форматы документов, должны быть установлены таким образом, чтобы допускать гибкость для руководителей в выборе форматов, подходящих для их проектов.

Определение системы обозначения документов. Стандартные обозначения документов необходимы для эффективного контроля документации. Обозначающая информация может включать в себя заглавие документа, ссылочный номер документа, номер версии документа, дату выпуска и пересмотра, реквизиты автора, реквизиты утвердившего лица, обозначение защищенности (авторских прав), обозначение организации.

Если документы выпускаются в виде разрозненных листов, каждая страница должна иметь индивидуальное обозначение (например, со ссылочным номером документа, номером страницы и номером издания).

3.2. Требования стандартов к программной документации

Как уже было отмечено, качество программного обеспечения, наряду с другими факторами, определяется полнотой и качеством пакета документов, сопровождающих ПС. К программным документам относятся документы, содержащие сведения, необходимые для разработки, изготовления, сопровождения программ и эксплуатации.

Единая система программной документации (ЕСПД) устанавливает следующие виды программной документации.

1. Спецификация. Состав программы и документация на нее.
2. Ведомость держателей подлинников. Перечень предприятий, на которых хранят подлинники программных документов.
3. Текст программы. Запись программы с необходимыми комментариями.
4. Описание программы. Сведения о логической структуре и функционировании программ.

5. Программа и методика испытаний. Требования, подлежащие проверке при испытании программы, а также порядок и методы контроля.

6. Техническое задание. Назначение и область применения программы, технические, технико-экономические и специальные требования, предъявляемые к программе, необходимые стадии и сроки разработки, виды испытаний.

7. Пояснительная записка. Схема алгоритма, общее описание алгоритма и функционирования программы, а также обоснование принятых технических и технико-экономических решений.

8. Эксплуатационные документы. Сведения для обеспечения функционирования и эксплуатации программы. Перечень эксплуатационных документов представлен в табл. 3.1.

Таблица 3.1

Вид эксплуатационного документа	Содержание эксплуатационного документа	Регламентирующие стандарты
Ведомость эксплуатационных документов	Перечень эксплуатационных документов на программу	ГОСТ 19.507-79
Формуляр	Основные характеристики программы, комплектность и сведения об эксплуатации программы	ГОСТ 19.501-78
Описание применения	Сведения о назначении программы, области применения, классе решаемых задач, применяемых методах, ограничениях для применения, минимальной конфигурации технических средств	ГОСТ 19.502-78
Руководство системного программиста	Сведения для проверки, обеспечения функционирования и настройки программы на условия конкретного применения	ГОСТ 19.503-79
Руководство программиста	Сведения для эксплуатации программы	ГОСТ 19.504-79
Руководство оператора	Сведения, необходимые для осуществления действий, связанных с выполнением программы вычислительной системой	ГОСТ 19.505-79
Описание языка	Описание синтаксиса и семантики языка	ГОСТ 19.506-79
Руководство по техническому обслуживанию	Сведения для применения текстовых и диагностических программ при обслуживании технических средств	ГОСТ 19.508-79

Полный пакет документов, разрабатываемых при создании автоматизированной системы и, в частности, программного обеспечения, установленный в отечественных стандартах, включает:

ГОСТ 34.602-89 – техническое задание на создание АС;

ГОСТ 34.201-90 – виды и комплектность документов;

РД 50-34.698-90 – пояснительная записка, схема функциональной структуры, общее описание системы, описание постановки задачи, описание информационного обеспечения системы, описание организации информационной базы, перечень входных сигналов и данных, перечень выходных сигналов/документов, описание программного обеспечения;

ГОСТ 19.201-78 – техническое задание;

ГОСТ 19.402-78 – описание программы;

ГОСТ 19.404-79 – пояснительная записка;

ГОСТ 19.301-79 – программа и методика испытаний.

Техническое задание. Содержание технического задания на разработку программных продуктов должно соответствовать ГОСТ 19.201-78 “Техническое задание. Требования к содержанию и оформлению”. Помимо разработки технического задания на все ПС могут разрабатываться технические задания на этапы, например, техническое задание на выполнение НИР.

Согласно ГОСТ 19.201-78, техническое задание на разработку ПС должно включать следующие разделы:

- введение;
- основания для разработки;
- назначение разработки;
- требования к программе;
- требования к программной документации;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки;
- приложения.

В зависимости от особенностей разрабатываемого ПС стандарт допускает уточнение содержания разделов, введение новых разделов или их объединение.

В разделе “Введение” указывается наименование, краткая характеристика области применения ПС.

В разделе “Основания для разработки” указываются:

- документ (документы), на основании которых ведется разработка;
- организация, утвердившая документ, и дата утверждения;
- наименование (условное обозначение) темы разработки.

В разделе “Назначение разработки” должно быть указано функциональное и эксплуатационное назначение ПС.

В раздел “Требования к программе” включаются следующие подразделы.

1. Требования к функциональным характеристикам, в которых указываются требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам. В данном подразделе описывается поведение ПС с точки зрения соотношения входа и выхода, без конкретизации его внутренней структуры. Описание выполняемых функций делается либо в текстовом виде, либо на специальном графическом языке. Описание входа заключается в фиксации синтаксиса и семантики всех входных данных. Описание выхода должно содержать точное описание всех возможных выходных данных в тесной взаимосвязи с входными.

2. Требования к надежности, где указываются требования к обеспечению надежного функционирования ПС, его защите (контроль входной и выходной информации, описание последствий отказов ПС и т.д.).

3. Условия эксплуатации, в которых должны быть указаны характеристики операционной среды, вид обслуживания, необходимое количество и квалификация персонала и др., а также допустимые параметры окружающей среды (относительная влажность, температура и др.).

4. Требования к составу и параметрам технических средств – необходимый состав технических средств (конфигурация) с указанием их основных технических характеристик.

5. Требования к информационной и программной совместимости, в которых указываются требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования и программным средствам, используемым ПС. Кроме того, могут указываться протоколы межмашинного сетевого обмена данными, стандарты протоколов формализации данных и управления терминалами, стандарты и форматы сообщений, протоколы транзакций, протоколы запросов данных, стандарты представления данных, требования к СУБД и операционным системам.

6. Требования к маркировке и упаковке.

7. Требования к транспортированию и хранению.

В разделе “Требования к программной документации” должен быть указан предварительный состав программной документации и при необходимости специальные требования к ней.

В разделе “Технико-экономические показатели” указываются: ориентировочная экономическая эффективность, предполагаемая годовая потребность, экономические преимущества разработки по

сравнению с лучшими отечественными и зарубежными образцами или аналогами.

В разделе “Стадии и этапы разработки” устанавливают необходимые стадии разработки, этапы и содержание работ (перечень документов, которые должны быть разработаны, согласованы и утверждены), а также сроки разработки и исполнителей.

В разделе “Порядок контроля и приемки” должны быть указаны виды испытаний и общие требования к приемке ПС. Здесь фиксируют важнейшие характеристики ПС в некоторой количественной или иной достаточно простой форме, с тем, чтобы можно было установить степень соответствия готового ПС принятым техническим условиям.

В приложениях к техническому заданию при необходимости приводят:

- перечень научно-исследовательских и других работ, обосновывающих разработку;
- схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые могут быть использованы при разработке;
- другие источники разработки.

Техническое задание на создание АС разрабатывается в соответствии с ГОСТ 34.602-89. Данный стандарт устанавливает следующие разделы, включаемые в техническое задание.

1. Общие сведения, включающие полное наименование системы, условное обозначение системы, шифр темы (шифр (номер) договора), наименование предприятий разработчика и заказчика системы и их реквизиты, перечень документов, на основании которых создается система, плановые сроки начала и окончания работ по созданию АС, сведения об источниках и порядке финансирования работ.

2. Назначение и цели создания АС, в котором указывают назначение системы и цели ее создания.

3. Характеристика объекта автоматизации.

4. Требования к системе. Данный раздел состоит из следующих подразделов:

а) Требования к системе в целом. Здесь указывают перечень подсистем, их назначение и основные характеристики, требования к числу уровней иерархии и степени централизации системы, требования к способам и средствам связи для информационного обмена между компонентами системы, требования к характеристикам взаимосвязей АС со смежными системами, требования к ее совместимости, способы обмена информацией. Кроме того, требования к численности

и квалификации персонала и режиму его работы, к надежности, безопасности и т.д..

б) Требования к функциям.

в) Требования к видам обеспечения (математическому, информационному, лингвистическому программному, техническому организационному и т. д.).

5. Состав и содержание работ по созданию (развитию) АС.

6. Порядок контроля и приемки системы.

7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу в действие.

8. Требования к документированию.

9. Источники разработки.

Основное внимание следует уделить руководящему документу РД 50-34.698-90, устанавливающему требования к содержанию документов на автоматизированные системы. Структура РД 50-34.698-90 приведена в табл. 3.2.

Содержание документов, разрабатываемых на предпроектных стадиях (“Формирование требований к АС” и “Разработка концепции АС”), приведено в рекомендуемом приложении к РД 50-34.698-90. На первой стадии разрабатывается отчет (ГОСТ 7.32) и заявка на разработку АС. На второй — отчет согласно ГОСТ 7.32.

Содержание организационно-распорядительных документов установлено также в рекомендуемом приложении. К организационно-распорядительным документам относятся:

- акт завершения работы;
- акты приемки в опытную и промышленную эксплуатацию;
- план-график работ;
- приказы о проведении работ и составе приемочной комиссии;
- протоколы испытаний и согласования.

Документ “*Описание программного обеспечения*” содержит вводную часть и разделы: структура ПО, функции частей ПО, методы и средства разработки ПО, операционная система, средства, расширяющие возможности операционной системы.

Во вводной части приводят основные сведения о техническом, информационном и других видах обеспечения АС, необходимые для разработки ПО или ссылку на соответствующие документы проекта АС.

В разделе “Структура ПО” приводят перечень частей ПО с указанием их взаимосвязей и обоснованием выделения каждой из них. В разделе “Функции частей ПО” приводят назначение и описание основных функций для каждой части ПО.

В разделе “Методы и средства разработки ПО” приводят перечень методов программирования и средства разработки ПО АС с

указанием частей ПО, при разработке которых следует использовать соответствующие средства и методы.

В разделе “Операционная система” указывают следующее.

1. Наименование, обозначение и краткую характеристику выбранной операционной системы и ее версии, в рамках которой будут выполняться разрабатываемые программы, с обоснованием выбора и указанием источников, где дано подробное описание выбранной версии.

2. Наименование руководства, в соответствии с которым должна осуществляться генерация выбранного варианта операционной системы.

3. Требования к варианту генерации выбранной версии ОС.

Раздел “Средства, расширяющие возможности операционной системы (ОС)” содержит подразделы, в которых для каждого используемого средства, расширяющего возможности операционной системы, указывают:

– наименование, обозначение и краткую характеристику средства с обоснованием необходимости его применения и указанием источников, где дано подробное описание выбранного средства;

– наименование руководства, в соответствии с которым следует настраивать используемое средство на конкретное применение;

– требования к настройке используемого средства.

Таблица 3.2

Область	Документы	Содержание
1	2	3
Документы по общесистемным решениям	1. Ведомость эскизного (технического) проекта	Выполняется по ГОСТ 2.106
	2. Пояснительные записки к эскизному, техническому проектам	Согласно РД 50-34.698-90
	3. Схема функциональной структуры	Согласно РД 50-34.698-90
	4. Ведомость покупных изделий	Выполняется по ГОСТ 2.106
	5. Описание автоматизируемых функций	Согласно РД 50-34.698-90
	6. Описание постановки задачи	Согласно РД 50-34.698-90
	7. Паспорт	Согласно РД 50-34.698-90
	8. Проектная оценка надежности системы	Согласно РД 50-34.698-90
	9. Общее описание системы	Согласно РД 50-34.698-90
	10. Программы и методика испытаний	Согласно РД 50-34.698-90
	11. Схема организационной структуры	Согласно РД 50-34.698-90

1	2	3
Документы с решениями по организационному обеспечению	1. Описание организационной структуры	Согласно РД 50-34.698-90
	2. Методика автоматизированного проектирования	Согласно РД 50-34.698-90
	3. Технологическая инструкция	Согласно РД 50-34.698-90
	4. Руководство пользователя	Согласно РД 50-34.698-90
	5. Описание технологического процесса обработки данных	Согласно РД 50-34.698-90
Документы с решениями по техническому обеспечению (основные)	1. Схема автоматизации	Согласно РД 50-34.698-90
	2. Описание комплекса технических средств	Согласно РД 50-34.698-90
	3. ТЗ на разработку специализированных технических средств	Согласно ГОСТ 15.001
	4. Схема структурная комплекса технических средств (ТС)	Согласно РД 50-34.698-90
Документы с решениями по информационному обеспечению (основные)	1. Перечень входных сигналов и данных	Согласно РД 50-34.698-90
	2. Перечень выходных сигналов	Согласно РД 50-34.698-90
	3. Описание информационного обеспечения системы	Согласно РД 50-34.698-90
	4. Описание организации информационной базы	Согласно РД 50-34.698-90
	5. Описание системы классификации и кодирования	Согласно РД 50-34.698-90
	6. Описание массива информации	Согласно РД 50-34.698-90
	7. Массив входных данных	Согласно РД 50-34.698-90
	8. Каталог БД	Согласно РД 50-34.698-90
	9. Состав выходных данных	Согласно РД 50-34.698-90
	10. Инструкция по формированию и ведению БД	Согласно РД 50-34.698-90
Документы с решениями по техническому обеспечению	Описание программного обеспечения	Согласно РД 50-34.698-90
Документы с решениями по математическому обеспечению	Описание алгоритма (проектной процедуры)	Согласно РД 50-34.698-90

“Пояснительная записка к эскизному (техническому) проекту” содержит следующие разделы (согласно РД 50-34.698-90): общие положения, описание процесса деятельности, основные технические решения, мероприятия по подготовке объекта автоматизации к вводу системы в действие.

В разделе “Общие положения” приводят:

- наименование проектируемой АС и наименование документов, их номера и дату утверждения, на основании которых ведут проектирование АС;
- перечень организаций, участвующих в разработке системы, сроки выполнения стадий;
- цели, назначение и область использования АС;
- подтверждение соответствия проектных решений действующим нормам и правилам техники безопасности и т.п.;
- сведения об использованных при проектировании нормативно-технических документах;
- сведения о НИР, передовом опыте, изобретениях, использованных при разработке проекта.

В разделе “Описание процесса деятельности” отражают состав процедур (операций) с учетом обеспечения взаимосвязи и совместимости процессов автоматизированной и неавтоматизированной деятельности, формируют требования к организации работ в условиях функционирования АС.

В разделе “Основные технические решения” приводят:

- решения по структуре системы, подсистем, средствам и способам связи для информационного обмена между компонентами системы, подсистем;
- решения по взаимосвязям АС со смежными системами, обеспечению их совместимости;
- решения по режимам функционирования, диагностированию работы АС;
- решения по численности, квалификации и функциям персонала АС, режимам его работы, порядку взаимодействия;
- сведения об обеспечении заданных в техническом задании потребительских характеристик системы (подсистем), определяющих ее качество;
- состав функций, комплексов задач, реализуемых системой (подсистемой);
- решения по комплексу технических средств, его размещению на объекте;

– решения по составу информации, объему, способам ее организации, видам машинных носителей, входным и выходным документам и сообщениям, последовательности обработки информации и другим компонентам;

– решения по составу программных средств, языкам деятельности, алгоритмам процедур и операций и методам их реализации.

В разделе приводят в виде иллюстраций другие документы, которые допускается включать по ГОСТ 34.201-90.

В разделе “Мероприятия по подготовке объекта автоматизации к вводу системы в действие” приводят:

– мероприятия по приведению информации к виду, пригодному для обработки на ЭВМ;

– мероприятия по обучению и проверке квалификации персонала;

– мероприятия по созданию необходимых подразделений и рабочих мест;

– мероприятия по изменению объекта автоматизации;

– мероприятия, исходящие из специфических особенностей создаваемых АС.

Рассмотрим содержание ряда документов, определенных в стандарте DO-178B. Некоторые документы, именуемые в данном стандарте как “данные жизненного цикла ПО”, уже рассматривались в гл. 2.

В DO-178B определено, что данные жизненного цикла могут принадлежать к одной из двух категорий: Категории Управления (КУ) 1 и Категории Управления (КУ) 2. Эти категории связаны с элементами управления конфигурацией. Данное разделение позволяет иметь средства управления затратами на разработку там, где может применяться менее строгий контроль без снижения степени защищенности данных. В приложении А к документу DO-178B приведены минимальные категории управления, назначенные каждому виду данных и их вариации в зависимости от уровня ПС (А, В, С, D, Е). В табл. 3.3, представлены некоторые фрагменты целей этапов жизненного цикла и их выходы – данные жизненного цикла.

Документ DO-178B устанавливает, что приложение А не предназначено для представления всех данных, необходимых для создания ПС, и не предусматривает какого-либо определенного способа хранения этих данных или их организации внутри структур хранения. В дополнении к указанным документам могут вырабатываться другие, необходимые для сертификации ПС.

Таблица 3.3

№	Цели	Применяемость ¹				Выходы	КУ			
		A	B	C	D		A	B	C	D
ЭТАП ПЛАНИРОВАНИЯ ПС										
1	Определение действий на этапе разработки и интегрированном этапе	*	*	*	*	План программных аспектов сертификации План разработки ПС План верификации ПС	1	1	1	1
2	Определение критериев перехода, взаимосвязей и согласованности между этапами	*	*	*		План управления конфигурацией ПС План определения качества ПС	1	1	2	2
3	Определение среды ЖЦ ПС	*	*	*	*		1	1	2	2
5	Определение стандартов разработки ПС	*	*	*		Стандарты требований Стандарты проектирования Стандарты кодирования	1	1	2	2
ЭТАП РАЗРАБОТКИ ПС										
1	Разработка ВУ требований	*	*	*	*	Данные требований к ПС	1	1	1	1
2	Определение установленных ВУ требований	*	*	*	*	Данные требований к ПС	1	1	1	1
3	Разработка архитектуры ПС	*	*	*	*	Описание проектирования	1	1	2	2
6	Разработка исходного кода	*	*	*	*	Исходный код	1	1	1	1
Верификация выходов подэтапа определения требований к ПС										
1	ВУ требований к ПС подчиняются требованиям к системе	—	—	*	*	Результаты верификации	2	2	2	2
2	ВУ требований точны и согласованы	—	—	*	*	Результаты верификации	2	2	2	2
7	Алгоритмы точны	—	—	*		Результаты верификации	2	2	2	

1	2	3	4	5	6	7	8	9	10	11
Тестирование выходов подэтапа интеграции										
1	Исполняемый объектный код подчиняются ВУ требованиям	*	*	*	*	Варианты и процедуры верификации Результаты верифика- ции	1 2	1 2	2 2	2 2
4	Исполняемый объектный код точно соответствует НУ требованиям	—	*	*		Варианты и процедуры верификации ПС Результаты верифика- ции ПС	1 2	1 2	2 2	

¹ — — цели должны быть удовлетворены независимо; * — цели должны быть удовлетворены; 1 — данные удовлетворяют КУ 1; 2 — данные удовлетворяют КУ 2

Характеристики данных жизненного цикла ПС следующие:

— *непротиворечивость*: информация непротиворечива, если она записана в терминах, допускающих единственное толкование и дополненная по необходимости определениями;

— *полнота*: информация полна, если она включает все необходимые требования и/или описательные материалы; все используемые диаграммы имеют комментарии, единицы измерения и термины полностью определены;

— *проверяемость*: информация проверяема, если она может быть проконтролирована на предмет корректности;

— *состоятельность*: информация состоятельна, если внутри нее нет конфликтов;

— *модифицируемость*: информация модифицируема, если она структурирована и стилизована таким образом, что вносимые изменения полноценны, непротиворечивы и корректны;

— *трассируемость*: информация трассируема, если могут быть определены источники ее компонентов;

— *форма*: форма должна обеспечить возможность эффективно получать доступ к данным жизненного цикла ПС в течение всего срока службы системы;

— *управление*: если предназначены для использования с этой целью, то данные должны быть определены в плане ПС, который регламентирует этап, для которого вырабатываются эти данные.

Дадим краткую характеристику основных данных жизненного цикла ПС.

План программных аспектов сертификации — первичное средство из используемых службами сертификации для определения,

предлагает ли разработчик такой жизненный цикл ПС, который удовлетворяет уровню разрабатываемого ПС. Данный план должен включать следующие разделы.

1. Обзор системы. Этот раздел содержит обзор системы, включая описание ее функций и их распределение между аппаратной и программной частями, архитектуры, программно-аппаратных интерфейсов, возможностей по обеспечению безопасности и т. д.

2. Обзор ПС. Здесь коротко описывают функции ПС с акцентом на предлагаемый уровень концепции безопасности.

3. Аспекты сертификации. Этот раздел содержит сводку базиса сертификации, включая средства обеспечения соответствия по отношению к программным аспектам сертификации. Здесь также устанавливается предлагаемый уровень ПС и приводятся выводы этапа оценки безопасности системы, включая потенциальную возможность работы ПС в неблагоприятных условиях.

4. Жизненный цикл ПС. В данном разделе описан используемый жизненный цикл и приводятся резюме по каждому его этапу и подэтапу, для которых детальная информация определяется в соответствующих планах на разработку ПС. Резюме определяет, какие цели каждого этапа и подэтапа ЖЦ ПС должны быть достигнуты, вовлекаемые структуры для их достижения и т. д.

5. Данные жизненного цикла ПС. Этот раздел определяет данные, которые будут выработаны и управляемы подэтапами ЖЦ ПС. Здесь также описывается отношение данных друг к другу и другим данным, определяющим систему, данные, требуемые для сертификации, форма данных и т. д.

6. Планировка. Этот раздел описывает средства, которые будут использованы разработчиком для предоставления отчетности по деятельности в течение жизненного цикла ПС при сертификации.

7. Дополнительные аспекты. Здесь приводятся специфические возможности, которые могут повлиять на этап сертификации: например, альтернативные методы обеспечения соответствия, оценка качества инструментальных средств, ранее разработанное ПС и т. д.

План оценки качества ПС устанавливает используемые методы для достижения целей этапа оценки качества ПС (ОКПС). План ОКПС может включать описание методов улучшения и прогрессивного управления этапом и может состоять из следующих разделов.

1. Среда. Описание среды оценки качества ПС, включая структуру, организационные ответственности и интерфейсы, стандарты, процедуры, средства, методы и метрики.

2. Полномочия. Характеристика полномочий, ответственности и независимости оценки качества ПС.

3. Методы. Методы оценки качества ПС, которые должны использоваться для каждого подэтапа жизненного цикла ПС на всем его протяжении, включая:

- методы ОКПС, такие, как обзоры, ведение протоколов, отчеты, проверки и наблюдение за этапами ЖЦ ПС;

- методы, относящиеся к оповещению о проблемах и их исправлению;

- методы описания проверки ПС.

4. Промежуточный критерий. Устанавливает промежуточный критерий для перехода к этапу оценки качества ПС.

5. Временные требования. Временные требования методик этапа ОКПС по отношению к методикам подэтапов ЖЦ ПС.

6. Результаты оценки качества ПС. Описание результатов, выработанных этапом оценки качества ПС.

7. Управление поставщиками. Описание средств оценки соответствия действий поставщиков нижнего уровня плану оценки качества ПС.

Требования к ПС определяют требования высокого уровня, включая и производные требования, если это необходимо. Этот документ должен включать следующее (но не ограничиваться этим).

1. Описание приведения системных требований к программным, с особым акцентом на требования безопасности и потенциальные условия возникновения сбойных ситуаций.

2. Функциональные и операционные требования для каждого режима работы.

3. Критерии функционирования, например, точность.

4. Временные требования и ограничения.

5. Ограничения по размеру памяти.

6. Программные и аппаратные интерфейсы, например, протоколы обмена, форматы данных, частота входных и выходных сигналов.

7. Требования на обнаружение сбоев и мониторинг за безопасностью функционирования.

8. Требования к расчленению ПС на отдельные компоненты и на их взаимодействие друг с другом (например, при реализации в виде распределенной системы).

Таким образом, хотя стандарт DO-178B и не устанавливает явно процессы документирования в своем жизненном цикле, ясно видно, что документации в нем уделено большое внимание.

В заключение данной главы кратко остановимся на документах, разрабатываемых в ходе процесса создания ПС согласно [3]. В про-

цессе разработки ПС появляются следующие документы, перечисленные ниже в хронологическом порядке:

- “Соглашение о требованиях”;
- “Внешняя спецификация”;
- “Внутренняя спецификация”.

Документ “Соглашение о требованиях” должен содержать первое письменное соглашение между заказчиком и разработчиком о том, что будет сделано, и что не будет делаться при разработке и выпуске программного средства. В отличие от него спецификация предполагает наличие более точных и исчерпывающих формулировок и определений. При этом первые два документа содержат информацию о том, что представляет собой ПС, а третий должен объяснять, как ПС устроено и как достигаются установленные для него цели и требования. Все документы имеют схожую структуру для облегчения контроля над проектом, а также для обеспечения прослеживаемости всех технических решений от требований до их реализации. По мере продвижения проекта разделы документа либо просто копируются в соответствующие разделы следующего создаваемого документа, либо расширяются описаниями технических решений текущего этапа.

Ниже приведена общая структура документа “Внешняя спецификация” с развернутыми комментариями в тех пунктах, которые касаются технической стороны дела (документ также включает экономические, управленческие и другие аспекты, которые не рассматриваются здесь).

1. Описание программного изделия

1.1. Наименование и шифры ПС (полное наименование, сокращенные наименования, шифры ПС и проекта).

1.2. Краткое описание ПС (включая сведения об авторском праве, иерархию документов, с указанием документов вышестоящих уровней).

1.3. Результирующие компоненты ПС (оформляется в виде таблицы или другой формы и включает в себя перечень спецификаций, другой документации и компонентов программного средства).

2. Цели

Этот раздел содержит причины выпуска ПС с указанием различного типа заявок, планов и т.п. и носит полностью управленческий характер.

3. Стратегия

3.1. Соглашения относительно представления материала.

3.1.1. Обозначения (определяются все обозначения, используемые в требованиях: например, если применяются индексы, то дается пример их использования и определяется принцип индексации).

3.1.2. Терминология (особенно специфическая для данного изделия).

3.1.3. Синтаксис (приводятся, если необходимо, синтаксические правила для дальнейшего описания требований).

3.2. Генерируемое программное средство (классифицируется как вспомогательное и порождаемое описываемым изделием).

3.3. Системные программные средства (все остальные ПС, включая операционные системы, утилиты, пакеты прикладных программ, которое классифицируется как основное, поскольку оно “генерирует” ПС предыдущего пункта).

Примечание. Причина такой расстановки пунктов состоит в том, что при правильном проектировании сверху вниз генерируемое программное средство является основной целью проектирования и должно быть описано раньше, чем его генератор. Другими словами, структура генерируемых программ должна определять структуру генератора, а не наоборот. Если все ПС является основным, то в п. 3.2 делается пометка “не используется” и опускаются все его подпункты. Структура п. 3.2 и 3.3 полностью дублируется и далее для простоты используется нумерация только п. 3.3.

3.3.л. Общие характеристики функции “*n*”. Если технически затруднительно и неестественно рассматривать ПС как один большой функциональный модуль, то следует привести его функциональную декомпозицию, показав связи между функциями (функциональными модулями) и присвоив каждой функции некоторое уникальное имя “*n*”. Затем для каждой функции отводится подраздел раздела 3.3 (т. е. 3.3.1, 3.3.2 и т. д.), в заглавии которого используется слово “функция” с последующим именем функционального модуля. Отметим, что такая функциональная декомпозиция не указывает, как именно ПС будет фактически разбито на программные модули (это составляет содержание документа “Внутренняя спецификация”). Для удобства работы, конечно, полезно иметь некоторое соответствие функционального и фактического разбиения, но это не является требованием и не должно уводить с правильного пути проектирования изделия.

3.3.л.1. Внешние ограничения.

3.3.л.1.1. Стандарты (список используемых промышленных стандартов и собственных стандартов предприятия).

3.3.л.1.2. Ограничения на совместимость. Необходимо рассматривать несколько аспектов совместимости: исходный язык, машинный язык, форматы данных и сообщений, форматы отчетов, форматы листингов и т. п. Специально должна оговариваться совместимость со следующими программными изделиями:

- изделиями-предшественниками (т. е. такими, которые пользователь может заменить новым изделием; если число функций при такой замене уменьшается, то следует привести обоснование этому);
- изделиями-компаньонами (т. е. относящимися к той же группе средств и являющимися альтернативой);
- подобными изделиями (т. е. выполняющими похожие функции в других программных изделиях);
- конкурирующими изделиями (других организаций).

3.3.л.1.3. Программные ограничения. Описываются программное окружение разрабатываемого ПС, включая указание средств для его загрузки и запуска. Также отмечаются все действующие программные ограничения, например использование вычислений с удвоенной точностью для некоторых функций.

3.3.л.1.4. Аппаратные ограничения. Приводится перечень устройств, необходимых для работы ПС (с указанием минимальной, оптимальной и максимальной конфигурации). Указываются все действующие ограничения на оборудование, например, физические характеристики терминала или требование запрещения использования звукового сигнального устройства.

3.3.л.2. Внешние характеристики. Если разрабатываемое ПС является расширением уже существующего, то описываются, главным образом, его дополнительные характеристики. В любом случае наибольшее внимание должно уделяться самым важным для конечного пользователя вопросам. Эти разделы являются основой документа и содержат полное и окончательное описание всех свойств программного изделия.

3.3.л.2.1. Результаты. Описываются все выходные данные ПС с точки зрения их функционального содержания и назначения (например, файлы, сообщения, программно-устанавливаемые сигналы и прерывания). При этом должны быть рассмотрены все возможные в системе носители и средства отображения информации. Указываются тип, структура, формат, объем, расположение и диапазон изменения. Для всех выходных данных, читаемых людьми (сообщения и отчеты) должны быть приведены образцы.

3.3.л.2.2. Процессы обработки. Описываются операции, выполняемые ПС в целом или функциональными модулями, рассматриваемыми как “черный ящик”. Если обсуждение идет на уровне модулей или этапов разработки, указываются также модули или этапы, требуемые для получения определенной выходной информации. Точно определяются все возможные ошибки, потенциальные условия их возникновения и способы рестарта и восстановления. Подраздел дол-

жен описывать инициацию, преобразование данных, все варианты завершения работы (нормального и аварийного).

3.3.л.2.3. Входы. Описание подобно п. 3.3.2.1.

3.3.л.3. Эргономические характеристики. Этот раздел описывает свойства, обеспечивающие надежность, комфорт и продуктивность работы пользователей и операторов, а также вопросы безопасности, секретности, восстанавливаемости после сбоев, мобильности ПС. Остановимся более подробно на двух подразделах: “Надежность” и “Рабочие характеристики”.

В разделе “Надежность” (это свойство программы понимается здесь как способность к восстановлению нормальной работы при ошибках и сбоях в работе оборудования) рассматриваются следующие вопросы:

- защита данных пользователя;
- степень защиты программ от ошибок, возникающих в других частях системы (например, работающих одновременно с данной программой в другой области памяти).

Следует рассмотреть, как могут повлиять на работу предлагаемых программных средств такие ошибки, учитывая следующие моменты: распределение ресурсов памяти (указать, если предусмотрено обеспечение изоляции отводимых областей памяти); связь программ через общие аппаратные ресурсы.

Раздел “Рабочие характеристики” описывает основные параметры или принципы, по которым должна оцениваться эффективность работы программы, по возможности в количественном виде с указанием возможных допусков. Все параметры должны быть измеряемыми, к их числу могут относиться быстродействие, пропускная способность, скорость передачи данных, расход машинных ресурсов, время реакции (или задержки) и т. д.

3.3.л.4. Внутренние характеристики (этот раздел полностью расширяется в документе “Внутренняя Спецификация”, однако частично может быть заполнен с целью полного описания соответствующих внешних свойств).

3.4. Внутренние ограничения (здесь речь идет о тех свойствах, которые пользователю логично ожидать, но которые по тем или иным причинам будут исключены из программного изделия или потенциально оставлены на усмотрение разработчика: например, неполная взаимозаменяемость носителей, отсутствие поддержки каких-либо возможностей оборудования, и т. п.).

4. *Используемые материалы (справочные).*

5. *Передача заказчику и ввод в действие.*

ГЛАВА 4

ОЦЕНКА ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО СРЕДСТВА

4.1. История проекта SPICE

В настоящее время нет необходимости никого убеждать в том, что разработка (приобретение) программного средства становится стратегическим направлением деятельности многих, если не большинства организаций в различных сферах жизни. Чтобы быть конкурентоспособными, организации должны использовать в своей деятельности различные передовые технологии, в первую очередь, информационные, а точнее, информационно-коммуникационные технологии (ICT – Information and Communication Technology). ICT является новым глобальным рынком, функционирующим по определенным законам, а рынок нуждается в стандартах. Программные средства являются ключевым элементом любого ICT-продукта.

Очевидна все возрастающая сложность ПС, наряду со все возрастающими требованиями к их качеству и надежности. Одним из факторов, влияющих на качество и надежность ПС, является стандартизация. Профессиональный подход к разработке ПС опирается на стандарты. Слово “профессиональный” как бы подразумевает, что в результате такого подхода будет создано высокоэффективное, надежное, качественное ПС.

Статистика в области программной индустрии констатирует не совсем примечательные факты. Так, более 15% всех программных проектов так и не достигли своего завершения, превышение стоимости проектов в 2–3 раза от запланированного является обычным явлением, при этом, если превышение стоимости составляет 30%, то это считается настоящим успехом.

Технологических и управленческих причин такого положения дел можно назвать несколько десятков, основными же являются следующие:

- помехи со стороны государственных органов;
- технологические факторы;

- низкий контроль процесса разработки;
- низкое качество управления проектом создания ПС, в первую очередь, плохое планирование производственных процессов.

Таким образом, заинтересованность в улучшении процессов разработки ПС для достижения требуемого клиентом качества становится более чем очевидной. Одним из путей повышения качества как самого ПС, так и процесса его разработки, является использование современных стандартов.

Основные стандарты в данной сфере рассмотрены нами выше. Прежде всего, это стандарты серии ИСО 9000. Кроме того, существуют еще такие стандарты, как ИСО 12207-95, СММ и ряд других.

ИСО 9000 используется в программной индустрии уже довольно давно, однако сам стандарт изначально создавался для производства вообще и не учитывал специфику, связанную с разработкой ПС. Таким образом, стало очевидно, что необходим стандарт, специально ориентированный на обеспечение качества при разработке ПС. Таких стандартов появилось несколько, а именно ИСО 12207-95 (гл. 2 данного пособия) и СММ.

Однако для построения системы качества недостаточно лишь декларировать цели такой системы и описать идеальный случай, необходим инструмент, позволяющий оценивать текущее состояние производственных процессов и определять пути их улучшения. Как ответ на подобное требования ИСО был начат проект, получивший аббревиатуру ИСО/МЭК 15504-97 (SPICE).

В июне 1991 г. четвертое полное заседание подкомитета SC7 совместного Технического Комитета 1 ИСО/МЭК одобрило программу (решение 144) в исследовании потребностей и требований для стандарта оценки процесса разработки ПС.

В июне 1993 г. был предложен проект SPICE, чтобы, во-первых, помочь проекту стандартизации в его подготовительной стадии разрабатывать начальные рабочие проекты, во-вторых, провести испытания у пользователя в порядке приобретения опыта, который формирует основу для изменения изданного стандарта до преобразования его в полный Международный стандарт, в третьих, сформировать понимание рынка и принять стандарт развития.

Проект SPICE завершил первую из этих задач. Рабочая группа WG10 подкомитета SC7 совместного Технического Комитета 1 ИСО/МЭК, которая является ответственной за развитие данного стандарта, впоследствии, в июне 1995 г., распространила рабочие проекты через SC7 для PDTR голосования. В настоящее время продолжают-

ся предприниматься поэтапные испытания у пользователей, чтобы обеспечить своевременную обработку отчетов об опыте его использования.

Проект ИСО/МЭК 15504-97 основывается и является дополнением к другим международным стандартам и моделям для оценки возможности и эффективности организаций и их процессов (рис. 4.1).

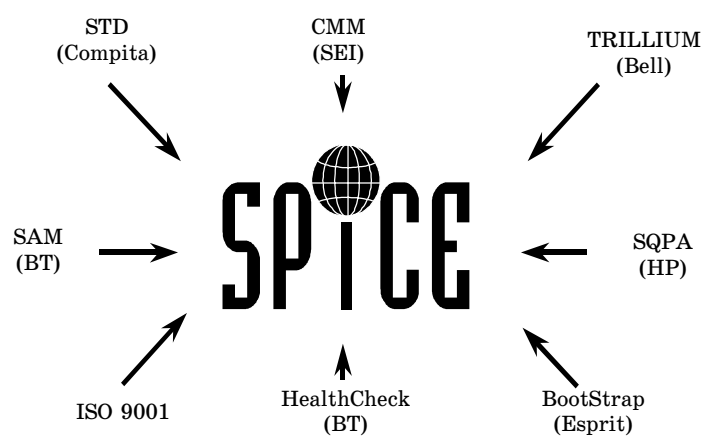


Рис. 4.1. Происхождение SPICE

SPICE вобрал в себя все самое лучшее из целого ряда популярных стандартов, он не стал простым их объединением. Для того чтобы показать, чем же SPICE отличается от своих предшественников, необходимо провести сравнение SPICE и наиболее известных стандартов из мира ПС.

Прежде всего, сравним процедуру оценки, которая будет детально рассмотрена нами ниже, принятую в SPICE с процедурой аудита, принятой в ИСО 9001-94. Результаты сравнения приведены в табл. 4.1.

Таблица 4.1

Оценка	Аудит
Детальные критерии	Абстрактные критерии
Внутреннее участие	Внешний, независимый
Сквозная	Краткий
Позитивное суждение	Негативное суждение
Поиск фактов	Поиск ошибок
Взаимодействие	Противостояние
Открытость	Защита
Общее обсуждение	Индивидуальные интервью

Приведенное сравнение однозначно показывает расположение SPICE относительно ИСО 9001-94. SPICE призван помочь пользователю оценить свою работу изнутри и взаимодействовать с внешними консультантами в конструктивном стиле, в отличие от ИСО 9001-94, где аудитор является экзаменатором, от которого хотят скрыть недостатки и промахи.

ИСО/МЭК 15504-97 включает намерения стандартов серии ИСО 9000 обеспечить уверенность в управлении качеством у поставщиков, обеспечивая пользователей руководством (каркасом) для независимой оценки возможности потенциальных поставщиков удовлетворить их потребности. Оценка процесса обеспечивает пользователей способностью оценить возможность процесса по непрерывной шкале простым и сравнимым способом, не используя характеристики “выполнено/не выполнено” аудита качества, базирующегося на ИСО 9001-94. Кроме того, руководство, описанное в проекте стандарта ИСО/МЭК 15504-97, предоставляет возможность регулировать сферу оценки для покрытия конкретных процессов, представляющих интерес, а не всех процессов, используемых в организации (табл. 4.2).

Таблица 4.2

Оценка	ИСО 9001-94
Развитая документация	Малая документация
Детальная модель	Абстрактная модель
Разработан для производства ПС	Разработан для производства в целом
Улучшение процессов и оценка возможностей	Сертификация
Шесть уровней возможности	Выполнено / не выполнено
Требования к оценке, руководство по применению	Только модель
Дополняет ИСО 9001-94	Дополняются SPICE

Как уже отмечалось, ИСО 12207-95 создавался специально для обеспечения качества программного средства. Следует напомнить, что ИСО 12207-95 ориентирован на программную индустрию, используется в специфическом контексте, содержит более детальную модель (во многом), полностью совместим со SPICE.

Теперь о связи SPICE и другого популярного стандарта, пришедшего из США – СММ (табл. 4.3). Эти два стандарта в некотором роде могут рассматриваться как конкурирующие, однако, при более детальном рассмотрении, выясняется, что между ними су-

ществует целый ряд слишком серьезных отличий, скорее, улучшающих жизнь пользователей, чем ухудшающих ее. С появлением SPICE у широкого круга пользователей появился реальный выбор между стандартами, и это уже дело каждого – решить, какой же из них более подходит к его конкретной деятельности.

Таблица 4.3

SPICE	CMM
Двумерная структура	Последовательная, одномерная структура
Допускает гибкость в выработке стратегии улучшения	Содержит предопределенный путь развития
Уровни возможностей для каждого процесса	Единый уровень зрелости для всех процессов
Результаты требуют упрощения	Результаты легко понимаемы
Результаты очень подробные	Упрощенные результаты

Модель зрелости возможностей (CMM) и ее структура показаны на рис. 4.2 и 4.3 соответственно.

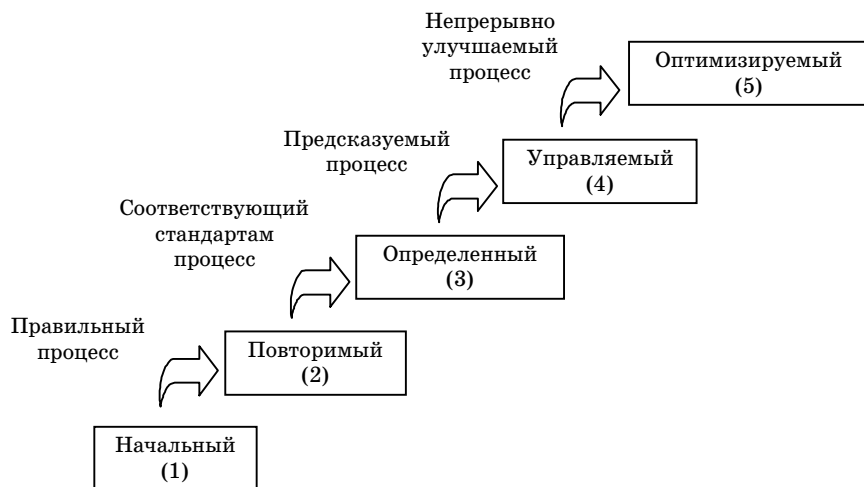


Рис. 4.2. Модель зрелости возможностей

Кроме перечисленных выше, достаточно известных стандартов, SPICE базируется на следующих, мало известных широкому кругу читателей, стандартах: Bootstrap и Trillium. Кратко приведем характеристики данных стандартов и результаты сравнения их с ИСО/МЭК 15504-97 (табл. 4.4, рис. 4.4, табл. 4.5).

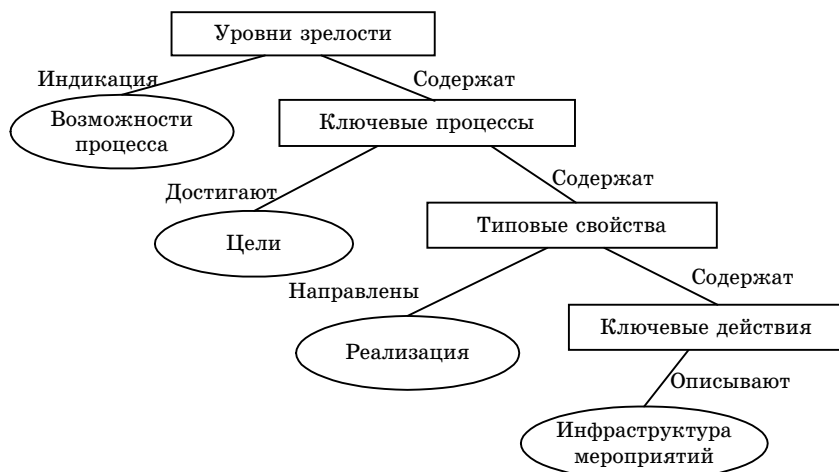


Рис. 4.3. Структура модели зрелости возможностей

Стандарт Bootstrap:

- модель базируется на уровнях зрелости CMM и SPICE;
- включает требования ИСО 9001-94, 9000-3-94, 9004-94 ;
- включает предписывающую модель процесса из DOD 2167 и ESA PS-005;
- направлен на организации, методологии и технологии;
- алгоритм уровня зрелости менее упрощенный, чем CMM;
- производит профиль зрелости для каждого процесса;
- оценка организации и проекта происходит посредством анкет;
- производит оценки процесса, квалификацию эксперта-консультанта;
- производит процесс для улучшения процесса разработки ПС.

Таблица 4.4

SPICE	Bootstrap
Независимый жизненный цикл	Неявный жизненный цикл
Стандартный процесс	Включает технологию и организацию
Любой метод оценки	Оценка, использующая анкеты
Требования для оценки	Метод оценки
Результат как профиль	Результат как профиль и фигура зрелости организации
Руководство по улучшению процесса и определения возможности	Процесс улучшения процесса разработки ПС включается в метод

Таблица 4.5

SPICE	Trillium
Общая цель	Конкретная
Программное средство	Перспективный продукт
Только процесс	Включает технологию
Улучшение процесса и определение возможности	Определение возможности
Любой метод улучшения	Маршрутная карта улучшения в модели
Детальный профиль	Простой профиль и единственная фигура возможности организации

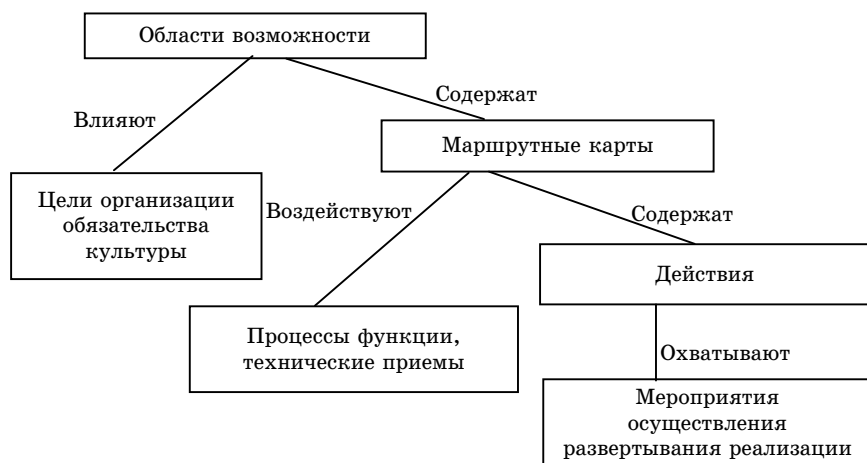


Рис. 4.5. Структура модели Trillium

ИСО/МЭК 15504-97 состоит из следующих частей, под общим названием *Оценка процесса разработки ПС*:

Понятия и вводное руководство (информативная).

Эталонная модель процессов и возможности процесса (нормативная).

Выполнение оценки (нормативная).

Руководство по выполнению оценки (информативная).

Модель оценки и показательное руководство (информативная).

Руководство по компетенции эксперта-консультанта (информативная).

Руководство для использования в улучшении процесса (информативная).

Руководство для использования в определении возможности процесса поставщика (информативная).

Словарь (информативная).

4.2. Основные положения

Проект ИСО (SPICE) по созданию стандарта оценки процесса производства ПС (ИСО/МЭК 15504-97) преследует следующие цели:

- удовлетворение реальных и растущих потребностей в оценке возможностей процесса разработки ПС в подразделениях организации;
- гармонизацию методов и моделей, используемых для оценки и улучшения процессов.

ИСО/МЭК 15504-97 обеспечивает руководство для оценки процесса разработки программного обеспечения. Оно может использоваться организациями для планирования, менеджмента, текущего контроля, управления и совершенствования приобретения, поставки, разработки, функционирования, развития и поддержки программного обеспечения.

ИСО/МЭК 15504-97 обеспечивает структурированный подход к оценке процесса разработки ПС:

- организацией или от ее имени с целью понимания состояния собственных процессов для улучшения процесса разработки ПС;
- организацией или от ее имени с целью определения пригодности собственных процессов для удовлетворения специфического требования или класса требований;
- организацией или от ее имени с целью определения пригодности процессов другой организации для специфического контракта или класса контрактов.

Руководство по оценке поощряет самостоятельную оценку, определяет управление для оцениваемых процессов, принимает во внимание контекст, в котором оцениваемые процессы функционируют, вырабатывает набор рейтингов процесса (профиля процесса), соответствует всем предметным областям и размерам организаций.

Использование оценки процесса разработки ПС внутри организации должно утвердить:

- культуру постоянного улучшения и учреждения соответствующих механизмов для поддержки и сопровождения этой культуры;
- инжиниринг процессов с целью удовлетворения деловых требований;
- оптимальное использование ресурсов.

Пользователи извлекут выгоду из использования оценки, определенной в данном стандарте. Ее использование в определении возможности процесса позволит:

- уменьшить неопределенность и риск в выборе поставщика программных систем при заключении контракта;
- поместить соответствующие средства управления в наиболее рискованные места жизненного цикла проекта;
- обеспечить определенную количественную основу для выбора финансовых потребностей, требований и оценить стоимость проекта относительно возможностей конкурирующих поставщиков.

Основные преимущества стандартизированного подхода к оценке процесса разработки заключаются в том, что:

- специалистам представлена общедоступная модель;
- достигнуто общее понимание в использовании оценки для улучшения процесса и определения его возможности;
- облегчена процедура определения возможности поставки оборудования;
- процесс разработки ПС управляется и регулярно просматривается в свете опыта использования;
- может быть изменен только с международного согласия;
- способствует гармонизации существующих моделей и схем оценки.

Подход к оценке процесса, определенный в ИСО/МЭК 15504-97, предоставляет основу для общего подхода к описанию результатов оценки, принимая во внимание некоторую степень сравнения оценок, базирующих на других, но совместимых, моделях и методах, рассмотренных нами выше.

Оценка имеет два принципиальных контекста для ее использования, как показано на рис. 4.6.

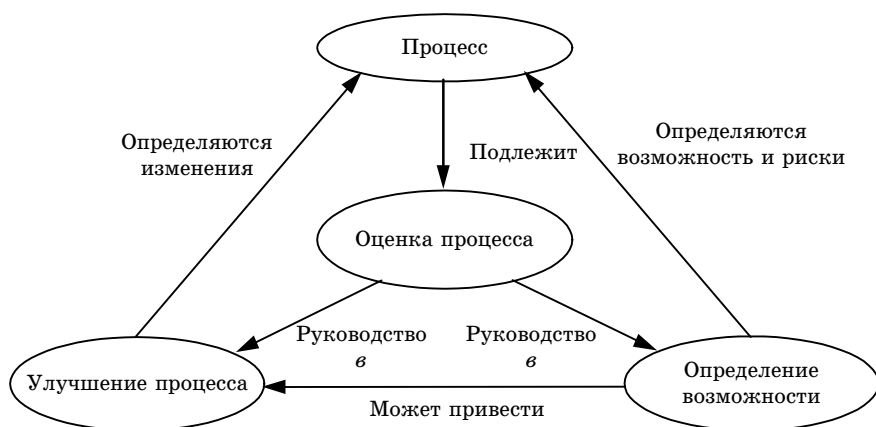


Рис. 4.6. Оценка процесса разработки ПО

В границах контекста улучшения процесса, результаты оценки характеризуют текущую деятельность внутри организации в понятиях возможности выбранных процессов. Анализ результатов оценки в свете бизнес-целей организации позволяет определить достоинства, недостатки и риски, свойственные ее процессам. Это, в свою очередь, приводит к ответу на вопрос, когда процессы эффективны в достижении своих целей, и определяет причины низкого качества, перерасхода времени и стоимости.

Определение возможности процесса достигать поставленной цели связано с анализом предлагаемой возможности процесса, выбранного для оценки относительно целевого профиля возможности процесса. Это необходимо для того, чтобы определить риски, включаемые в реализацию проекта, использующего данный процесс. Предлагаемая возможность может базироваться как на основе результатов предыдущих оценок, так и на основе результатов оценки, выполненной с целью установления предлагаемой возможности.

ИСО/МЭК 15504-97 разрабатывается для удовлетворения потребностей пользователей, поставщиков и экспертов-консультантов, и их требований.

Преимущества использования этого блока документов заключаются в следующем:

- для *пользователей*: в способности определить текущую и потенциальную возможность процессов разработки организации-поставщика ПС;
- для *поставщиков*: в способности определить текущую и потенциальную возможность своих собственных процессов разработки ПС;

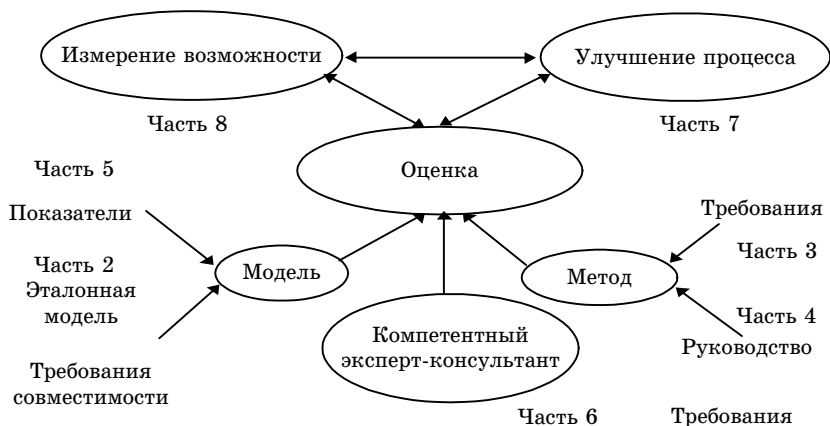


Рис. 4.7. Обзор связей элементов стандарта ИСО/МЭК 15504-97

в способности определить области и приоритеты в улучшении процессов организации; в определении схемы, которая указывает маршрут для улучшения процессов разработки ПС;

– для *экспертов-консультантов*: в руководстве для проведения оценки процесса разработки ПС.

Верхний уровень связей между оценкой процесса, улучшением процесса и определением возможности процессов показан на рис. 4.7 с указанием мест различных компонентов ИСО/МЭК 15504-97.

Каркас оценки

Контекст оценки процесса обобщен на рис. 4.8. Часть 2 ИСО/МЭК 15504-97 определяет эталонную модель, которая обеспечивает основу для рейтингов возможности процессов, базирующуюся на достижении определенных атрибутов процесса. Часть 3 ИСО/МЭК 15504-97 определяет требования для выполнения оценки и устанавливает обстоятельства, в которых может быть произведено сравнение результатов оценки. Часть 4 ИСО/МЭК 15504-97 обеспечивает

Из улучшения или определения
возможности процесса

Входы оценки:

- цель оценки
- сфера оценки
- ограничения оценки
- обязанности оценки
- дополнительная информация

Действия оценки:

- планирование
- сбор данных
- проверка данных
- ранжирование
- документирование процесса



Оценка
процесса

Набор показателей:

- показатели выполнения процесса
- показатели возможности процесса

Модель процесса

Эталонная модель:

- цель процесса
- атрибуты процесса

Выход оценки:

- профили процесса
- запись оценки

В улучшение процесса или
определение возможности

Рис. 4.8. Контекст оценки процесса

пользователей руководящими положениями по выполнению оценки и интерпретации требований Части 3. Эти руководящие положения являются достаточно общими, чтобы быть применимыми всеми организациями, а также для проведения оценок, использующих различные методы, технологии и инструментальные средства.

Формальный вход в оценку происходит с определения цели оценки (почему она выполняется), сферы оценки (процессы, которые должны быть оценены) и ограничений, если имеются, относящиеся к оценке. Входы оценки определяют также обязанности для выполнения оценки и другую дополнительную информацию.

Оценка выполняется над выбранными процессами на основе модели оценки. Эта модель оценки процессов должна быть совместима с эталонной моделью. Еталонная модель — это двухмерная модель, состоящая из набора процессов и набора атрибутов процесса. Атрибуты процесса применяются ко всем процессам. Они группируются в уровни возможности, которые могут использоваться для определения того, как процесс управляется. Выходы оценки включают набор профилей процессов и дополнительный рейтинг уровня возможности для каждого оцениваемого процесса.

Оценка содержит, по крайней мере, пять специфических действий: планирование, сбор данных, верификацию данных, ранжирование процессов и документирование. Процесс оценки должен быть документирован. Кроме того, эксперты-консультанты должны записать объективные показатели выполнения или использованной возможности, чтобы доказать достижение рейтингов. Оценка процесса может быть выполнена как группой, по крайней мере, с одним компетентным экспертом-консультантом, компетенция которого описана в ИСО/МЭК 15504-97-6, так и на непрерывной основе, с использованием пригодных инструментальных средств для сбора данных, подтвержденных компетентным экспертом-консультантом.

Еталонная модель процессов и возможности процесса формируют основу для любой модели, используемой для целей оценки процесса. Еталонная модель включает двухмерный подход к оценке возможности процесса — одно измерение определяет оцениваемые процессы, второе — описывает шкалу для измерения возможности. Любая модель, совместимая с эталонной моделью, может использоваться для оценки, и результаты любых совместимых оценок могут переноситься в общую базу.

Контекст улучшения процесса

Успешное улучшение процесса разработки ПС происходит в деловом контексте, учитывающем специфические потребности и бизнес-цели организации, ключевые ограничения, такие как, например, ресурсы, культура и т. п. Это отражено на рис. 4.9.



Рис. 4.9. Улучшение процесса

Часть 7 стандарта ИСО/МЭК 15504-97 обеспечивает пользователей руководящими положениями по использованию результатов оценки как метода для выполнения улучшения процесса ПС в непрерывном цикле. Руководящие положения определяют:

- введение оценки процесса разработки ПС;
- использование результатов оценки;
- измерение эффективности процесса ПО и улучшение этой эффективности;
- установление действий улучшения в соответствии с бизнес-целями;
- использование эталонной модели как маршрутной карты для улучшения;
- вопросы культуры в контексте улучшения процесса разработки ПС и вопросы управления улучшением данного процесса.

Предусмотренные руководящие положения непосредственно строятся на ИСО 9004-4. Они не рассматривают специфические организационные структуры, философию управления, жизненный цикл процессов ПС или методы разработки ПС. Понятия и принципы подхо-

дят для различных деловых потребностей, предметных областей и размеров организаций, так что они могут использоваться всеми организациями-разработчиками ПС для проведения улучшения своих процессов.

Контекст определения возможности

Процедура определения возможности процесса описана в ИСО/МЭК 15504-97-8. Определение возможности процесса строится, главным образом, на оценке процесса. Процессы ранжируются, а затем сравниваются с процессами эталонной модели, используя измерение и структуру рейтингов, описанных в Части 2 ИСО/МЭК 15504-97. Контекст определения возможности процесса показан на рис. 4.10.



Рис. 4.10. Определение возможности процесса

Пользователь программных продуктов или услуг имеет технические и другие потребности, что выражается в специфических требованиях. Прежде чем заключить контракт, пользователь должен определить возможность процесса поставщика, или поставщик может захотеть установить свою собственную возможность процесса прежде, чем ответить на предложение пользователя. Технические и дру-

гие потребности в определении возможности процесса подтверждаются в специфических требованиях.

Специфические требования переводятся в целевую возможность, которая представляет вход в оценку процесса. Поставщик может выдвинуть предлагаемую возможность процесса как набор “процесс – процесс” рейтингов уровня возможности, который должен предлагаться заинтересованным организациям. В простой ситуации предлагаемая возможность процесса может базироваться на последней самооценке или на других средствах. В более сложных случаях поставщик может предложить возможность процесса, которая может быть достигнута в будущем, базируясь на текущем профиле и планах улучшения процесса.

Доверие к предлагаемой возможности процесса анализируется вместе с рисками, включенными и описанными в отчете о возможности процесса.

Часть 8 ИСО/МЭК 15504-97 обеспечивает руководящие положения об использовании результатов оценки для целей определения возможности процесса поставщиков. Руководящие положения направлены на определение возможности процесса, как для использования в пределах организации с целью определения рисков, связанных с новым проектом, так и для использования пользователем для оценки внешних поставщиков.

4.3. Эталонная модель и выполнение оценки

Архитектура эталонной модели искусственно включает два измерения:

- *измерение процесса*, которое характеризует результаты процесса, являющиеся существенными измеримыми целями процесса;
- *измерение возможности процесса*, которое характеризует набор атрибутов процесса, применимых к любому процессу и представляющие собой измеримые характеристики, которые необходимы для управления процессом и улучшающие возможность его выполнения.

При измерении процесса эталонная модель группирует процессы в три группы, содержащие пять категорий процессов, согласно типу деятельности, к которым они обращены.

Начальные процессы жизненного цикла состоят из категорий процессов поставщик-заказчик и инжиниринга.

Категория процесса поставщик-заказчик состоит из процессов, на которые непосредственно воздействуют заказчик, разработка поддержки и перехода ПС заказчику, предусматривают корректное фун-

кционирование и использование программного продукта и/или услуг.

Категория процесса инжиниринга состоит из процессов, которые непосредственно определяют, осуществляют или поддерживают программный продукт, его отношение к системе и его документации потребителя (заказчика).

Поддерживающие процессы жизненного цикла состоят из категорий процесса поддержки.

Категория процесса поддержки состоит из процессов, которые могут использоваться любым из других процессов (включая другие поддерживающие процессы) в различных точках жизненного цикла ПС.

Организационные процессы жизненного цикла состоят из категорий процессов управления и организации.

Категория процесса управления состоит из процессов, которые содержат методы общего характера и могут использоваться любым, кто управляет любым типом проекта или процессом внутри жизненного цикла программного обеспечения.

Категория процесса организации состоит из процессов, которые устанавливают бизнес-цели организации и разрабатывают (развивают) процесс, продукт и активные ресурсы, которые, когда используют проекты в организации, помогут ей достичь бизнес-целей.

Каждый процесс в эталонной модели описан в терминах утверждения цели. Эти утверждения включают в себя уникальные функциональные цели процесса, которые подтверждаются в конкретной среде. Утверждение цели включает дополнительный материал, определяющий выходы успешной реализации процесса. Соответствие цели процесса представляет первый шаг в формировании возможности процесса.

Эталонная модель не определяет как, или в каком порядке, элементы утверждений цели процесса должны быть достигнуты. Цели процесса будут достигнуты в организации через различные действия более низкого уровня, задачи и методики, выполняемые, чтобы произвести рабочий продукт. Эти выполняемые задачи, действия и методики, а также характеристики произведенных рабочих продуктов, являются показателями, которые демонстрируют, достигнута ли цель конкретного процесса.

Развитие возможности процесса характеризуется в терминах атрибутов процесса, сгруппированных в уровни возможности. Атрибуты процесса являются признаками процесса, которые могут быть оценены по шкале достижения, обеспечивая меру возможности про-

цесса. Атрибуты применимы ко всем процессам. Каждый атрибут процесса описывает аспект полной возможности управления и улучшения эффективности процесса в достижении его целей и содействующего бизнес-целям организации.

Уровень возможности характеризуется набором атрибутов, которые работают вместе. Каждый уровень обеспечивает главное расширение возможности выполнения процесса. Уровни составляют рациональный путь развития через усовершенствование возможности любого процесса.

В эталонной модели определены шесть уровней возможности.

Уровень 0: *Незавершенный*. Общая неудача в достижении цели процесса. Имеются нелегко идентифицированные рабочий продукт или выходы процесса.

Уровень 1: *Выполняемый*. Цель процесса, в общем, достигнута. Достижение не может строго планироваться и отслеживаться. Персонал организации осознает, что процесс должен выполняться, и имеется общее согласие, что этот процесс выполняется, как требуется и когда требуется. Имеются определенные рабочие продукты процесса, и они свидетельствуют в пользу достижения цели.

Уровень 2: *Управляемый*. Процесс вырабатывает рабочие продукты, согласно определенным процедурам, планируется и отслеживается. Рабочие продукты соответствуют конкретным стандартам и требованиям. Основное различие от *Выполняемого уровня* в том, что при выполнении процесса теперь вырабатываются рабочие продукты, которые полностью выполняют требования к качеству в пределах определенного промежутка времени и выделенного ресурса.

Уровень 3: *Установленный*. Процесс выполняется и управляется, используя определенный процесс, основанный на хороших принципах инжиниринга программного обеспечения. Индивидуальные реализации процесса используют документирующие процессы, утвержденные, приспособленные версии стандарта в достижении выходов определенного процесса. Ресурсы, необходимые для установления определения процесса – также на месте. Основное отличие от *Управляемого уровня* в том, что процесс *Установленного уровня* использует определенный процесс, который способен достигнуть своих выходов.

Уровень 4: *Предсказуемый*. На практике определенный процесс последовательно выполняется в пределах определенных ограничений и достигает определенных целей. Подробные меры выполнения процесса собраны и проанализированы. Это ведет к количественному пониманию возможности процесса и улучшенной способности

предсказать выполнение. Выполнение процесса объективно управляется. Качество рабочих продуктов количественно известно. Основное отличие от *Установленного уровня* в том, что определенный процесс теперь выполняется последовательно внутри определенных ограничений, чтобы достичь своих определенных выходов.

Уровень 5: *Оптимизирующий*. Выполнение процесса оптимизируется, чтобы достичь текущие и будущие деловые потребности. Процесс достигает повторяемости при достижении определенных бизнес-целей. Количественная эффективность процесса и цели эффективности для выполнения установлены, базируются на бизнес-целях организации. Непрерывный процесс, контролирующий эти цели, позволяет получать количественную обратную связь, и усовершенствование достигнуто анализом результатов. Основное отличие от *Предсказуемого уровня* в том, что определенные и стандартные процессы теперь динамически изменяются и адаптируются, чтобы эффективно достичь текущих (актуальных) и будущих бизнес-целей.

Естественно, эталонная модель не может использоваться, как основа для проведения надежных и непротиворечивых оценок возможности процесса, так как уровень детализации не достаточен. Описания цели процесса и атрибутов возможности в эталонной модели необходимо поддерживать исчерпывающим набором показателей выполнения процесса и его возможности. Таким образом, будет возможен непротиворечивый рейтинг возможности процесса.

Измерение процесса

Приведем классификацию процессов, принятых в организациях, занимающихся разработкой, эксплуатацией, приобретением, поставкой и поддержкой функционирования ПС. Классификация распознает пять категорий процессов, которые содержат все процессы. Категории и их процессы сопоставимы с теми процессами, которые определены в проекте стандарта ИСО/МЭК 12207-95.

Как было отмечено выше, в эталонной модели процессы объединяются в три группы и пять категорий процессов:

- *начальные процессы жизненного цикла* включают категории процесса инжиниринга (ENG) и поставщик-заказчик (CUS);
- *поддерживающие процессы жизненного цикла* включают категории процесса поддержки (SUP);
- *организационные процессы жизненного цикла* включают категории процесса управления (MAN) и организации (ORG).

Описание каждой категории процессов включает характеристику процессов, которые он содержит, сопровождающуюся списком имен процессов.

Индивидуальные процессы описаны в терминах шести компонентов.

Идентификатор процесса. Идентифицирует категорию и последовательный номер внутри этой категории. Схема нумерации различается между процессами верхнего уровня и процессами второго уровня. Идентификатор состоит из двух частей: сокращения категории (например, ENG для категории процесса инжиниринга) и номер (например, CUS. 1 обозначает Процесс Приобретения и CUS. 1.2 обозначает процесс второго уровня, Процесс Выбора Поставщика, который является составляющим (компонентным) процессом Процесса Приобретения).

Имя процесса. Описательная фраза, которая выделяет принципиальное свойство процесса (например, Выбор Поставщика).

Тип процесса. Имеется 3 типа процессов верхнего уровня (базисный, расширенный, новый) и 2 процесса второго уровня (компонентный, расширенный), которые имеют следующее отношение к процессам ИСО/МЭК 12207-95. Новые процессы дополнительные к тем, что определены в ИСО/МЭК 12207-95. Базисные процессы идентичны в предназначении процессам ИСО/МЭК 12207-95. Расширенные процессы дополняются на существующем процессе ИСО/МЭК 12207-95. Компонентные процессы группируют одно или большее количество действий ИСО/МЭК 12207-95 из того же самого процесса. Расширенные компонентные процессы группируют одно или большее количество действий ИСО/МЭК 12207-95 из того же самого процесса и включают дополнительный материал.

Цель процесса. Материал, который указывает цель процесса, устанавливающего общие цели выполнения процесса на верхнем уровне. Необязательный дополнительный материал может быть включен, чтобы далее определить утверждение цели.

Результаты процесса. Список описаний результатов процесса.

Примечания процесса. Необязательный список информативных примечаний относительно процесса и его отношения к другим процессам.

В качестве примера ниже приводится несколько процессов.

CUS.1. Процесс Приобретения

Базисный процесс

Цель *Процесса Приобретения* состоит в том, чтобы получить продукт и/или услугу, которые удовлетворяют потребности, выражен-

ные заказчиком (клиентом). Процесс начинается с определения потребности заказчика и требуемых результатов с принятием продукта и/или услуги, необходимых заказчику. В результате успешной реализации процесса:

- будет разработан контракт, который ясно выражает ожидания, обязанности и обязательства, как заказчика, так и поставщика;
- будет произведен продукт и/или услуга, что удовлетворит установленную потребность заказчика;
- приобретение будет проверено таким образом, чтобы определенные ограничения как, например, стоимость, план и качество, были выполнены.

ENG.1. Процесс Разработки

Базисный процесс

Цель *Процесса Разработки* состоит в том, чтобы трансформировать согласованный набор требований в функциональный программный продукт или программную систему, которые удовлетворяют установленным потребностям заказчика. В результате успешной реализации процесса:

- будет разработан программный продукт или программная система;
- будут разработаны промежуточные рабочие продукты, что показывает, что конечное изделие основывается на согласованных требованиях;
- будет установлена непротиворечивость между требованиями программного обеспечения и проектами программного обеспечения;
- данные тестирования будут показывать, что конечный продукт встречает согласованные требования;
- конечный продукт будет установлен в целевую среду и принят заказчиками.

ПРИМЕЧАНИЕ: Согласованные требования можно обеспечивать операцией *Процесса Приобретения* (CUS. 1) или *Процессом Установления Требования* (CUS. 3).

ENG.1.1. Процесс Разработки и Анализа Системных Требования

Компонентный процесс ENG.1 – Процесса Разработки

Цель *Процесса Разработки и Анализа Системных Требования* состоит в том, чтобы установить системные требования (функциональные и нефункциональные) и архитектуру, идентифицируя, какие системные требования должны быть распределены к каким элементам системы и в какой версии. В результате успешной реализации процесса:

- будут разработаны системные требования, что соответствует установленным потребностям заказчика;
- будет предложено решение, идентифицирующее основные элементы системы;
- согласованные требования будут распределены каждому из основных элементов системы;
- будет разработана стратегия версии, определяющая приоритет для выполнения системных требований;
- системные требования будут одобрены и модифицированы, как и требуется;
- требования, предложенное решение и их связи будут сообщены всем заинтересованным сторонам.

SUP.1. Процесс Документирования

Расширенный Процесс

Цель *Процесса Разработки Документации* состоит в том, чтобы разработать и поддерживать документы, которые записывают информацию, произведенную процессом или действием. В результате успешной реализации процесса:

- будет разработана стратегия, идентифицирующая документы, сформированные в течение жизненного цикла программного продукта;
- будут определены стандарты, к которым должны обращаться при разработке документов;
- все документы, формируемые процессом или проектом, будут идентифицированы;
- содержание и цель всех документов будет определена, рассмотрена и одобрена;
- все документы будут разрабатываться и издаваться в соответствии с определенными стандартами;
- все документы будут поддерживаться в соответствии с определенными критериями.

ПРИМЕЧАНИЕ: Процесс поддерживает исполнение атрибута процесса 2.2 в тех примерах, где он введен.

MAN.1.1. Процесс Управления Проектом

Компонентный процесс MAN.1 – процесса Управления

Цель *Процесса Управления Проектом* состоит в том, чтобы идентифицировать, устанавливать, координировать и контролировать действия, задачи и ресурсы, необходимые для проекта создания продукта и/или встречи услуги согласованным требованиям. В результате успешной реализации процесса:

- будет определена область работ проекта;

- будет оценена выполнимость достижения целей проекта с доступными ресурсами и ограничениями;
- будут измерены и оценены задачи и ресурсы, необходимые для завершения работы;
- будут идентифицированы и проверяться интерфейсы между элементами проекта, другими проектами и организационными модулями;
- будут разработаны и выполняться планы выполнения проекта;
- будет проверен и будет сообщаться прогресс проекта;
- действия, чтобы исправить отклонения от плана и предотвратить рекуррентное соотношение проблем, идентифицированных в проекте, будут приниматься, когда цели проекта не достигнуты.

Измерение возможности

Измерение возможности эталонной модели определяет шкалу измерения для оценки возможности любого процесса. Возможность процесса определена на шести пунктах шкалы, которая позволяет оценить возможность от незавершенного до оптимизирующего уровня. Шкала определяет повышение возможности выполняемого процесса от эффективности, которая не способна к достижению определенных результатов вплоть до эффективности и является способной к достижению бизнес-цели и поддержке непрерывного улучшения процесса. Следовательно, шкала определяет четкий маршрут улучшения для каждого индивидуального процесса.

Внутри модели мера возможности основана на наборе из девяти атрибутов процесса (АП) (табл. 4.6). Атрибуты процесса используются, чтобы определить, достиг ли процесс данной возможности. Каждый атрибут измеряет специфический аспект возможности процесса. Атрибуты измеряются в процентах и, следовательно, обеспечивают более наглядное понимание специфических аспектов возможности процесса, требуемой, чтобы поддерживать улучшение процесса и определение его возможности. Атрибуты более подробно будут рассмотрены нами ниже.

Рейтинг атрибутов процесса. Атрибут процесса представляет измеримую характеристику любого процесса.

Шкала рейтинга – шкала в процентах, от нуля до ста, которая представляет степень достижения атрибута.

Порядковая шкала рейтинга, определенная ниже, используется для калибровки уровня достижения определенной возможности атрибутов процесса.

N – не достигнутый: 0–15 % – есть маленькое или нет вообще подтверждения достижения определенного атрибута.

P – частично достигнутый: 16–50 % – есть подтверждение надежного систематического метода к достижению определенного атрибута, некоторые аспекты достижения могут быть непредсказуемы.

L – в значительной степени достигнутый: 51–85 % – есть подтверждение надежного систематического метода к значительному достижению определенного атрибута, выполнение процесса может измениться в некоторых областях.

F – полностью достигнутый: 86–100 % – есть подтверждение полного и систематического метода к полному достижению определенного атрибута, никаких значительных недостатков не существует в пределах определенной части организации.

Таблица 4.6

Номер	Название
Уровень 1	Выполняемый процесс
АП 1.1	Атрибут выполнения процесса
Уровень 2	Управляемый процесс
АП 2.1	Атрибут управления выполнением
АП 2.2	Атрибут управления рабочим продуктом
Уровень 3	Установленный процесс
АП 3.1	Атрибут определения и преобразования процесса
АП 3.2	Атрибут ресурса процесса
Уровень 4	Предсказуемый процесс
АП 4.1	Атрибут измерения процесса
АП 4.2	Атрибут контроля процесса
Уровень 5	Оптимизирующий процесс
АП 5.1	Атрибут изменения (верификации) процесса
АП 5.2	Атрибут возможности дальнейшего улучшения

Каждый атрибут процесса, оцененный в любой части организации, включая самый высокий уровень возможности, определенный в сфере оценки, должен быть согласован с рейтингом, который использует шкалу атрибута, определенную выше. Набор рейтингов атрибута для процесса формирует профиль для этого процесса. Выход оценки включает набор профилей для всех оцененных процессов.

Каждый рейтинг атрибута процесса должен иметь ссылку, которая записывает имя процесса и оцененный атрибут процесса.

Используемый идентификатор должен давать объективное подтверждение использованию, чтобы определить рейтинг, который должен извлекаться. Рейтинги могут представляться в любом формате, как, например, матрицы или, как часть базы данных, при условии, что представление допустит идентификацию индивидуальных рейтингов, согласно схеме.

Модель уровня возможности процесса. Уровень возможности, достигнутый процессом, должен быть получен из рейтинга атрибута для этого процесса, согласно модели уровня возможности процесса, определенной в табл. 4.7. Цель этого требования состоит в том, чтобы гарантировать единообразие значений, когда уровень возможности процесса ссылается для процесса.

Таблица 4.7

Шкала	Атрибуты процесса	Оценка
Уровень 1	Выполнение процесса	В основном или полностью
Уровень 2	Выполнение процесса Управление выполнением Управление рабочим продуктом	Полностью В основном или полностью То же
Уровень 3	Выполнение процесса Управление выполнением Управление рабочим продуктом Определение и преобразование процесса Ресурс процесса	Полностью То же — " В основном или полностью То же
Уровень 4	Выполнение процесса Управление выполнением Управление рабочим продуктом Определение и преобразование процесса Ресурс процесса Измерение процесса Контроль процесса	Полностью То же — " — " — " В основном или полностью То же
Уровень 5	Выполнение процесса Управление выполнением Управление рабочим продуктом Определение и преобразование процесса Ресурс процесса Измерение процесса Контроль процесса Изменение процесса Возможность дальнейшего улучшения	Полностью То же — " — " — " — " — " В основном или полностью То же

Ниже приведена таблица, содержащая итоговые списки процессов, которые включены в эталонную модель, и соответствие процессов эталонной модели процессам, определенным в стандарте ИСО/МЭК 12207-95 (табл. 4.8).

Таблица 4.8

12207-95	Действия и процессы 12207-95	15504-97	Процессы 15504-97	Тип
1	2	3	4	5
5.	Начальные процессы ЖЦ			
5.1	Процесс приобретение	CUS.1	Приобретение	Базисный
5.1.1	Инициализация	CUS.1.1	Подготовка приобретения	Компонента
5.1.2	Подготовка Заявки-для-Предложения	CUS.1.2	Выбор поставщика	Компонента
5.1.3	Подготовка контракта и корректировка	CUS.1.2	Выбор поставщика	Компонента
5.1.4	Проверка поставщика	CUS.1.3	Проверка поставщика	Компонента
5.1.5	Принятие и завершение	CUS.1.4	Одобрение заказчиком	Компонента
5.2	Процесс поставки	CUS.2	Поставка	Базисный
5.2.1	Инициализация	CUS.2	Поставка	Базисный
5.2.2	Подготовка ответа	CUS.2	Поставка	Базисный
5.2.3	Контракт	CUS.2	Поставка	Базисный
5.2.4	Планирование	CUS.2	Поставка	Базисный
5.2.5	Выполнение и управление	CUS.2	Поставка	Базисный
5.2.6	Обзор и оценка	CUS.2	Поставка	Базисный
5.2.7	Поставка и завершение	CUS.2	Поставка	Базисный
		CUS.3	Установление требований	Новый
5.3	Процесс разработки	ENG.1	Разработка	Базисный
5.3.1	Реализация процесса	ENG.1	Разработка	Базисный
5.3.2	Анализ системных требований	ENG.1.1	Разработка и анализ системных требований	Компонента
5.3.3	Разработка архитектуры системы	ENG.1.1	Разработка и анализ системных требований	Компонента
5.3.4	Анализ требований ПС	ENG.1.2	Анализ программных требований	Компонента

Продолжение табл. 4.8

1	2	3	4	5
5.3.5	Разработка архитектуры ПС	ENG.1.3	Разработка ПС	Компонента
5.3.6	Рабочий проект ПС	ENG.1.3	Разработка ПС	Компонента
5.3.7	Кодирование и тестирование ПС	ENG.1.4	онструкция ПС	Компонента
5.3.8	Интеграция ПС	ENG.1.5	Интеграция ПС	Компонента
5.3.9	Тестирование квалификации ПО	ENG.1.6	Тестирование ПО	Компонента
5.3.10	Интеграция системы	ENG.1.7	Тестирование и интеграция системы	Компонента
5.3.11	Тестирование квалификации системы	ENG.1.7	Тестирование и интеграция системы	Компонента
5.3.12	Инсталляция ПС	CUS.2	Поставка	Базисный
5.3.13	Поддержка программ	CUS.2	Поставка	Базисный
5.4	Процесс функционирования	CUS.4	Функциональное использование	Базисный
5.4.1	Реализация процесса	CUS.4.1	Функциональное использование	Расширенная компонента
5.4.2	Функциональное тестирование	CUS.4.1	Функциональное использование	Расширенная компонента
5.4.3	Функционирование системы	CUS.4.1	Функциональное использование	Расширенная компонента
5.4.4	Поддержка пользователя	CUS.4.2	Поддержка пользователя	Расширенная компонента
5.5	Процесс эксплуатации	ENG.2	Эксплуатация ПС и системы	Базисный
5.5.1	Реализация процесса	ENG.2	Эксплуатация ПС и системы	Базисный
5.5.2	Анализ проблем и модификаций	ENG.2	Эксплуатация ПС и системы	Базисный
5.5.3	Реализация модификации	ENG.2	Эксплуатация ПС и системы	Базисный
5.5.4	Принятие в эксплуатацию	ENG.2	Эксплуатация ПС и системы	Базисный
5.5.5	Миграция	ENG.2	Эксплуатация ПС и системы	Базисный
5.5.6	Утилизация ПС	ENG.2	Эксплуатация ПС и системы	Базисный

Продолжение табл. 4.8

1	2	3	4	5
6.	Вспомогательные процессы ЖЦ			
6.1	Процесс документирования	SUP.1	Документирование	Расширенный
6.1.1	Реализация процесса	SUP.1	Документирование	Расширенный
6.1.2	Разработка и развитие	SUP.1	Документирование	Расширенный
6.1.3	Продукция	SUP.1	Документирование	Расширенный
6.1.4	Эксплуатация	SUP.1	Документирование	Расширенный
6.2	Управление конфигурацией	SUP.2	Управление конфигурацией	Базисный
6.2.1	Реализация процесса	SUP.2	Управление конфигурацией	Базисный
6.2.2	Идентификация конфигурации	SUP.2	Управление конфигурацией	Базисный
6.2.3	Контроль конфигурации	SUP.2	Управление конфигурацией	Базисный
6.2.4	Учет статуса конфигурации	SUP.2	Управление конфигурацией	Базисный
6.2.5	Оценка конфигурации	SUP.2	Управление конфигурацией	Базисный
6.2.6	Управление выпуском и поставкой	SUP.2	Управление конфигурацией	Базисный
6.3	Процесс гарантии качества	SUP.3	Гарантия качества	Базисный
6.3.1	Реализация процесса	SUP.3	Гарантия качества	Базисный
6.3.2	Гарантия продукта	SUP.3	Гарантия качества	Базисный
6.3.3	Гарантия процесса	SUP.3	Гарантия качества	Базисный
6.3.4	Системы гарантии качества	SUP.3	Гарантия качества	Базисный
6.4	Процесс верификации	SUP.4	Верификация	Базисный
6.4.1	Реализация процесса	SUP.4	Верификация	Базисный
6.4.2	Верификация	SUP.4	Верификация	Базисный
6.5	Процесс проверки достоверности	SUP.5	Проверка достоверности	Базисный
6.5.1	Реализация процесса	SUP.5	Проверка достоверности	Базисный
6.5.2	Проверка достоверности	SUP.5	Проверка достоверности	Базисный
6.6	Процесс совместного обзора	SUP.6	Совместный обзор	Базисный

1	2	3	4	5
6.6.1	Реализация процесса	SUP.6	Совместный обзор	Базисный
6.6.2	Обзоры управления проектом	SUP.6	Совместный обзор	Базисный
	Технические обзоры	SUP.6	Совместный обзор	Базисный
6.7	Процесс проверки	SUP.7	Проверка	Базисный
6.7.1	Реализация процесса	SUP.7	Проверка	Базисный
6.7.2	Аудит	SUP.7	Проверка	Базисный
6.8	Процесс решения проблем	SUP.8	Решение проблем	Базисный
6.8.1	Реализация процесса	SUP.8	Решение проблем	Базисный
6.8.2	Решение проблем	SUP.8	Решение проблем	Базисный
		SUP.9	Измерение	Новый
		SUP.10	Повторное использование	Новый
7.	Организационные процессы ЖЦ			
7.1	Процесс управления	MAN.1	Управление	Базисный
7.1.1	Инициализация и определение области	MAN.1.1	Управление проектом	Компонента
7.1.2	Планирование	MAN.1.1	Управление проектом	Компонента
7.1.3	Выполнение и контроль	MAN.1.1	Управление проектом	Компонента
7.1.4	Обзор и оценка	MAN.1.1	Управление проектом	Компонента
7.1.5	Закрытие	MAN.1.1	Управление проектом	Компонента
		MAN.2	Управление качеством	Новый
		MAN.3	Управление риском	Новый
		ORG.1	Организационное выравнивание	Новый
7.2	Процесс инфраструктуры	ORG.4	Инфраструктуры	Базисный
7.2.1	Реализация процесса	ORG.4	Инфраструктуры	Базисный
7.2.2	Создание инфраструктуры	ORG.4	Инфраструктуры	Базисный
7.2.3	Эксплуатация инфраструктуры	ORG.4	Инфраструктуры	Базисный

1	2	3	4	5
7.3	Процесс усовершенствования	ORG.2	Усовершенствование	Базисный
7.3.1	Создание процесса	ORG.2.1	Создание процесса	Компонента
7.3.2	Оценка процесса	ORG.2.2	Оценка процесса	Компонента
7.3.3	Усовершенствование процесса	ORG.2.3	Усовершенствование	Компонента
7.4	Подготовка процесса	ORG.3	Управления людскими ресурсами	Расширенный
7.4.1	Реализация процесса	ORG.3	Управление людскими ресурсами	Расширенный
7.4.2	Подготовка существенной разработки	ORG.3	Управление людскими ресурсами	Расширенный
7.4.3	Подготовка реализации плана	ORG.3	Управление людскими ресурсами	Расширенный

Выполнение оценки

Ниже определяется набор минимальных требований для выполнения оценки. Данные требования увеличат вероятность того, что результаты оценки процессов являются целевыми, беспристрастными, непротиворечивыми, повторимыми и представительными, и определяют обстоятельства, при которых результаты оценки сравнимы.

Вход должен быть определен до оценки и одобрен заказчиком оценки. Вход оценки, как минимум, должен определять следующее.

1. Личность заказчика оценки и его отношение к оцениваемой организации.

2. Цель оценки.

3. Сферу оценки, включая:

а) процессы, которые будут исследованы внутри организации;
б) самый верхний уровень возможности, который должен исследоваться;

в) часть организации, которая разворачивает эти процессы;

г) контекст процесса, который, как минимум, включает размер организации, демографию организации, предметную область продукта или услуг организации, размер, уязвимость и сложность изделий или услуг, качественные характеристики изделий (см., например, ИСО/МЭК 9126-91).

4. Ограничения целостности оценки, которые могут включать:

а) доступность ключевых ресурсов;

б) максимальное количество времени, которое должно использоваться для оценки;

в) специфические процессы, которые должны быть исключены из оценки;

г) минимальный, максимальный или специфический типовой размер, или покрытие, которые желательны для оценки;

д) собственность на выполнение оценки и любые ограничения на их использование;

е) средства управления на информацию, которая следует из соглашения конфиденциальности.

5. Подлинность модели, используемой в пределах оценки, которая должна быть совместима с хорошей моделью инжиниринга ПС.

6. Личности экспертов-консультантов, включая компетентного эксперта-консультанта, ответственного за оценку.

7. Личности экспертов-консультантов и персонала поддержки со специфическими обязанностями для оценки.

8. Любая дополнительная информация, которая будет собрана в течение оценки, чтобы поддерживать улучшение процесса или определение возможности процесса.

Любые изменения во входе оценки должны быть согласованы с заказчиком и зарегистрированы в записи оценки. Заказчик оценки должен проверять, что эксперт-консультант, который берет ответственность за оценку и наблюдение за ней (компетентный эксперт-консультант), имеет необходимую компетентность и умения. Эксперт-консультант должен подтвердить требования заказчика, прежде чем приступить к оценке.

Компетентный эксперт-консультант должен гарантировать, что оценка проводится в соответствии с определенными требованиями и принимает во внимание релевантные руководства в других частях ИСО/МЭК 15504-97.

Эксперт-консультант должен гарантировать, что все другие участники оценки проинформированы о целях, сфере и методе оценки. По завершении оценки, компетентный эксперт-консультант должен подтвердить, что требования были выполнены.

Оценка должна проводиться, согласно зарегистрированному процессу, который является способным удовлетворить цель оценки.

Оценка процесса должна содержать следующие действия.

1. Планирование. План оценки должен быть разработан и зарегистрирован, определяя, как минимум: требуемые входы; действия, которые нужно выполнить для проведения оценки; ресурсы и план, назначенный этим действиям; выбор и определенные обязанности

экспертов-консультантов и участников организации оценки; критерии для проверки исполнения требований.

2. Сбор данных. Данные, требуемые для оценки процессов в пределах области оценки, должны быть собраны систематическим и упорядоченным способом, применяя следующие принципы:

- должны быть явно идентифицированы стратегия и методы сбора и анализа данных;

- должно быть установлено соответствие между процессами организации, определенное в сфере оценки через совместимую модель, используемую для оценки, и процессами, определенными в эталонной модели;

- каждый процесс, идентифицированный в области оценки, должен быть оценен на основе целевого доказательства;

- целевое доказательство, собранное для каждого атрибута оцененного процесса, должно быть достаточным для удовлетворения цели оценки и ее сферы;

- цель – утвержденное доказательство, основанное на показателях оценок атрибута процесса, которые поддерживают решения эксперта-консультанта, должна быть зарегистрирована, чтобы обеспечить основу для проверки оценок. Эти записи должны включать:

- а) методы и исследованные рабочие продукты;

- б) имена личностей, обеспечивающих информацию;

- в) обсуждение оценок.

3. Проверка правильности данных. Собранные данные должны быть утверждены, гарантируя, что утвержденные данные достаточно покрывают область оценки.

4. Ранжирование процесса. Оценка должна быть назначена и утверждаться для каждого атрибута процесса, до и включая, самый высокий уровень возможности, определенный в области оценки, подразделения.

Набор рейтингов атрибута процесса должен быть зарегистрирован как профиль процесса подразделения.

Чтобы обеспечивать основу для повторяемости оценки, определенный набор показателей оценки в совместимой модели должен использоваться в течение оценки, чтобы поддерживать суждения эксперта-консультанта в рейтинге атрибутов процесса.

5. Результаты. Результаты оценки, включая, как минимум, выходы, определенные ниже (запись оценки), должны быть зарегистрированы и сообщаться заказчику оценки.

Критерии для квалификации компетентного эксперта-консультанта должны быть зарегистрированы.

Эксперты-консультанты, участвующие в оценке, должны иметь доступ к соответствующим материалам, описывающим как выполнить оценки, и необходимую компетентность, использовать любые приборы или инструментальные средства, выбранные, чтобы поддерживать оценку.

Запись оценки, как минимум, должна содержать:

- дату оценки;
- вход оценки (исходные данные);
- используемый метод оценки и инструментальные средства;
- имена экспертов-консультантов, которые провели оценку, включая компетентного эксперта-консультанта, ответственного за оценку;
- набор профилей процесса, следующих из рейтингов (т. е. один профиль для каждого оцененного процесса);
- расположение записей объективного доказательства, обеспечивающего оценки атрибута процесса в профилях процесса, и эксперт-консультант(ов), ответственных за решение оценки,
- любую дополнительную информацию, собранную в течение оценки, которая была идентифицирована в начале оценки, чтобы поддерживать улучшение процесса или определение возможности процесса;
- время, потраченное экспертом-консультантом, и уровень возможности процесса или уровни, как определено в эталонной модели.

Запись должна содержать следующее утверждение “..., эта оценка проведена в соответствии с требованиями ИСО 15504-97 “Оценка процесса разработки программных средств””.

4.4. Модель оценки

Эталонная модель, определенная выше, обеспечивает общую основу для выполнения оценок возможности процесса разработки ПС, принимая во внимание результаты, использующие общую шкалу рейтинга. Эталонная модель определяет двухмерную модель возможности процесса. Одно измерение, измерение процесса, определяет и классифицирует процессы разработки ПС в пять категорий процесса. Второе измерение, измерение возможности, определяет серию атрибутов процесса, сгруппированных в уровни возможности. Атрибуты процесса представляют собой измеряемые характеристики возможности процесса.

Эталонная модель, как указывалось выше, не может быть использована как основа для проведения надежной и последователь-

ной оценки возможности процесса, так как предусмотренный уровень детализации в ней не достаточный. Описания цели и атрибутов процесса в эталонной модели должны поддерживаться исчерпывающим набором показателей выполнения процесса и его возможности. В этом подразделе представлена примерная модель оценки, включающая эти показатели.

Полная модель оценки основывается на эталонной модели и является совместимой с ней, и может использоваться в качестве основы для проведения оценки возможности процесса разработки ПС (рис. 4.11). Чтобы выполнить оценку, нужна дополнительная информация о методах. Описание примерного метода не входит в цели данного учебного пособия.

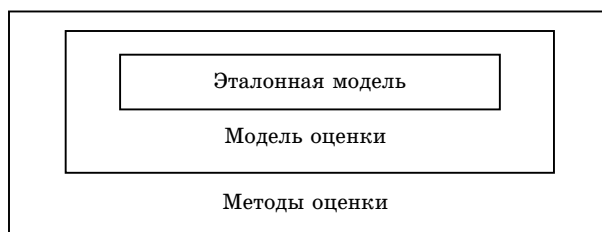


Рис. 4.11. Отношения между эталонной моделью, моделью оценки и методами оценки

Структура и принципы модели оценки

Основная структура модели оценки идентична эталонной модели, определенной выше. Существует связь один к одному между категориями процессов, процессами, целями процесса, уровнями возможностей процессов и атрибутами процессов эталонной модели и моделью оценки.

Модель оценки расширяет эталонная модель, добавляя определение и используя показатели оценки (рис. 4.12). Показатели оценки – объективные атрибуты или характеристики действия, или рабочего продукта, которые поддерживают решения эксперта-консультанта о выполнении и возможности оцениваемого процесса.

Базовое действие и характеристики рабочего продукта касаются процессов, определенных в измерении процессов эталонной модели и выбираются, чтобы явно направлять достижение целей для определенных процессов. Базовые действия и рабочие продукты являются показателями выполнения процесса. Базовые действия являются функциональной декомпозицией процесса и демонстрируют выполнение процессов. Они определены для поддерж-



Рис. 4.12. Связь между эталонной моделью и моделью оценки

ки решения о степени достижения цели процесса и его результатов (выходов). Процессы также используют и формируют (входы и выходы) рабочие продукты со специфическими характеристиками.

Действия управления касаются атрибутов процесса, определенных в измерении возможности процесса эталонной модели. Доказательство их эффективного выполнения поддерживает решение о степени достижения атрибута. Действие управления со связанными показателями атрибута — показатели возможности процесса. Действия управления являются средствами достижения возможностей, указанных атрибутами процесса. Данные действия связываются с показателями атрибутов, такими как:

- характеристики выполнения действия, которые описывают руководство в его реализации;
- характеристики ресурсов и инфраструктуры, которые обеспечивают механизмы для помощи в управлении процессом;
- процессы, связанные с измерением процесса, которые поддерживают действие управления.

Конкретные действия управления связываются с каждым атрибутом процесса. Показатели атрибута помогают устанавливать

объективные подтверждения, что действия управления выполняются.

Принципы модели оценки. Модель оценки основана на следующем принципе: возможность процесса может быть оценена демонстрацией достижения атрибутов процесса. Каждый процесс в измерении процесса имеет набор связанных базовых действий, выполнение которых обеспечивает демонстрацию степени достижения цели процесса. Аналогично, каждый атрибут процесса в измерении возможности имеет набор связанных действий управления, выполнение которых обеспечивает демонстрацию степени достижения предписанного атрибута в процессе.

Показатели, определенные в модели оценки – не требуемые характеристики процесса. Они представляют доказательство, которое могло бы быть найдено в экземпляре процесса и, следовательно, может использоваться для суждения (решения) о достижении возможности. Набор показателей обеспечивает набор подробных отличий, которые могут использоваться для оценки, способствует ли конкретный экземпляр действия цели процесса или достижению его атрибута.

Представление выполнения процесса. Модель оценки группирует процессы при измерении процесса в пять категорий, согласно типу деятельности, к которому они обращены. Группировки идентичны тем, что определены в эталонной модели.

Базовое действие – деятельность по инжинирингу ПС или управлению проектом, которое направляет цель конкретного процесса. Последовательное выполнение базовых действий, связанных с процессом, поможет достигнуть цели процесса. Согласованный набор базовых действий связан с каждым процессом в измерении процесса.

Базовые действия описаны на абстрактном уровне, идентифицируя, “что” должно быть выполнено без определения “как”. Они характеризуют выполнение процесса. Выполнение только базовых действий процесса представляет первый шаг в формировании возможности процесса, но они представляют уникальные, функциональные действия процесса, если даже это выполнение не систематическое. Выполнение базовых действий может быть непредсказуемым, противоречивым, плохо спланированным, и/или результат проявляется в рабочих продуктах низкого качества. Выполнение процесса, однако, производит рабочие продукты, которые являются, по крайней мере, незначительно пригодными для использования в достижении цели процесса. В модели оценки каж-

дый рабочий продукт имеет определенный набор характеристик, которые могут использоваться, чтобы оценить эффективную реализацию процесса.

Представление возможности процесса. Развитие возможности процесса выражено в модели оценки в терминах атрибутов процесса, сгруппированных в уровни возможности. Атрибуты и уровни возможности идентичны тем, которые определены в эталонной модели.

Атрибуты процесса – характеристики, которые могут быть оценены по шкале достижения, обеспечивая меру возможности процесса. Они применимы ко всем процессам. Каждый атрибут процесса описывает аспект полной возможности управления и улучшения эффективности процесса в достижении цели и содействующего бизнес-целям организации.

Уровень возможности – набор атрибутов, работающих вместе, обеспечивающих основное увеличение возможности выполнить процесс. Уровни составляют рациональный путь развития через улучшение возможности любого процесса.

В пределах модели оценки измерение потенциальных возможностей основано на наборе из девяти атрибутов процесса эталонной модели. Атрибуты процесса использованы для определения, достиг ли процесс данной возможности. Каждый атрибут измеряет конкретный аспект возможности процесса. Атрибуты эволюционируют по четырехточечной порядковой шкале рейтингов. Следовательно, они обеспечивают проникновение в суть специфических аспектов возможности процесса, требуемое для поддержки улучшения процесса и определения возможности.

Следуя определению атрибута каждого процесса, модель оценки обеспечивает такой набор действий управления, который, если адекватно выполненный в процессе создания экземпляров, достигнет характеристик атрибута. Действия управления – действия управления родового типа и применимы ко всем процессам. Они разрабатываются для достижения принципиальных функций управления по планированию, организации, распределению ресурсов и контролю. Обычно, но не обязательно, иметь четыре действия для каждого атрибута.

Связанные с каждым действием управления, наборы показателей предоставляют типовое подтверждение, доказывающее степень, в которой действие управления выполняется. Эти показатели включают характеристики выполнения действия, связанные характеристики ресурсов и инфраструктуры, связанные процессы.

Использование показателей в рейтинге процессов. Выход из оценки процесса – набор профилей процесса, один для каждого образца процесса внутри области применения оценки. Каждый профиль процесса состоит из набора девяти рейтингов атрибута для оцененного процесса. Каждый рейтинг атрибута представляет суждение эксперта-консультанта о степени достижения атрибута.

Модель оценки представляет набор показателей, формирующих основу набора зарегистрированного доказательства, позволяющего максимизировать объективность суждений эксперта. Показатель определен как целевой атрибут или характеристика действия (рабочего продукта), который поддерживает решение о выполнении или возможности процесса. Из этого определения могут быть идентифицированы два различных класса показателей: показатели выполнения процесса и показатели возможности процесса. Внутри контекста этой модели эти типы показателей касаются, соответственно, базовых действий, определенных при измерении процесса, и действий управления при измерении возможности. Классы и типы показателей, их отношение к выходу оценки показаны на рис. 4. 13.



Рис. 4.13. Связь между показателями и оценками атрибута процесса

Показатели — атрибуты или характеристики, существование которых подтверждает, что определенные действия выполняются и для которых возможно собрать объективное доказательство в ходе оценки. Каждое такое доказательство получается или из исследования рабочих продуктов оцениваемых процессов, или из утверждений, сделанных исполнителями и администраторами процессов. Существование рабочих продуктов и их характеристик обеспечивает доказательство выполнения действий, связанных с ними. Аналогично утверждение, полученное от исполнителей процесса, обеспечивает доказательство относительно выполнения действий и способа, которым они выполняются.

Показатели модели включают списки предложенного доказательства, которое эксперт-консультант мог получить, или наблюдать, в ходе оценки. Они не предназначены, чтобы быть расцененными как обязательный набор контрольных списков, которые должны сопровождаться, но довольны как руководство для эксперта-консультанта в накоплении необходимого объективного доказательства, чтобы поддерживать их суждения о возможности. Полученное подтверждение должно быть зарегистрировано в форме, которая явно определяет типы показателей и классы, так, чтобы поддержка для суждений эксперта-консультанта могла быть легко подтверждена или проверена.

Измерение процесса в модели оценки

В модели оценки при измерении процесса полное описание каждого процесса включает:

- сферу цели процесса, принимаемую из эталонной модели;
- примечания, усиливающие эту сферу, где необходимо;
- набор базовых действий для процесса, необходимых для достижения цели процесса;
- число входных и выходных рабочих продуктов, связанных с каждым процессом;
- характеристики, связанные с каждым рабочим продуктом.

Для примера приведем описание по одному процессу из каждой категории. Полный список базовых действий каждого процесса приведен в Прил. 2.

CUS.1. Приобретение ПС

Целью процесса *Приобретения ПС* является получение продукта и/или услуги, которые удовлетворяют потребности, выраженные клиентом. Процесс начинается с идентификации потребности клиента и результатов с принятием продукта и/или услуги, необходимых клиенту. В результате успешной реализации процесса:

- будет разработан контракт, в котором ясно выражено ожидание, обязанности и ответственность, как поставщика, так и клиента;
- продукт и/или услуга, которые будут произведены, удовлетворят потребности клиента;
- процесс приобретения будет управляемым, чтобы выполнить установленные ограничения (например, такие как стоимость, график и качество).

ENG.2. Разработка требований к ПС

Цель процесса *Разработки программных требований* – формулировка требований к программному обеспечению системы. Как результат успешной реализации этого процесса:

- будут определены требования, предъявляемые к различным программным компонентам системы и их интерфейсам, соответствующим настоящим и предполагаемым потребностям клиента;
- должны быть разработаны проанализированные, корректные и поддающиеся проверке требования к программному обеспечению;
- должно быть понято влияние программных требований на операционную среду;
- должна быть разработана соответствующая стратегия выпуска, определяющая приоритет в реализации требований к системе;
- программные требования должны быть проверены и откорректированы, если необходимо;
- требования к программному средству должны быть переданы всем задействованным сторонам.

SUP.3. Выполнение гарантии качества

Целью процесса *Гарантия качества* является обеспечение гарантии того, что рабочие продукты и действия процесса или проекта согласуются со всеми применяемыми стандартами, процедурами и требованиями. Результаты успешной реализации процесса:

- должны быть определены, спланированы и расписаны действия по гарантии качества процессов и проекта;
- должны быть определены стандарты качества, методологии, процедуры и средства для выполнения действий по обеспечению гарантии качества;
- должны быть установлены ресурсы и ответственность для выполнения действий по обеспечению гарантии качества;
- должна быть установлена и гарантироваться независимость ответственных лиц за выполнение действий по гарантии качества;
- должны быть выполнены определенные действия по гарантии качества в соответствии с планами и графиками.

MAN.2. Управление качеством

Целью процесса *Управления качеством* является управление качеством продукта и услуг проекта, а также гарантия того, что они удовлетворяют клиента. Процесс включает установление фокуса на управление качеством продукта и процесса, как проекта, так и организационного уровня. В результате успешной реализации процесса:

- будут установлены цели качества, основанные на требованиях качества заказчика, для различных контрольных точек внутри жизненного цикла ПС;
- будут определены и использованы метрики, которые измеряют результаты действий проекта в контрольных точках жизненного цикла проекта, и оценивают, были ли цели качества достигнуты;
- будут идентифицированы и интегрированы в модель жизненного цикла ПС действия, которые помогут достичь целей качества;
- будут выполнены идентифицированные действия качества;
- будут приниматься корректирующие действия, когда цели качества не достигнуты.

ORG.2. Определение процесса

Цель *Определения процесса* – сформировать библиотеку многократно используемых определений процесса (включая стандарты, процедуры и модели), которая будет поддерживать исполнение устойчивых и повторяющихся процессов управления и инжиниринга ПС (все процессы, включенные в это руководство). В результате успешной реализации процесса:

- будет существовать хорошо определенный и поддерживающий стандарты комплект процессов, вместе с индикацией применимости каждого процесса;
- будут определены подробные задачи, действия и связанные рабочие продукты для каждого стандартного процесса, вместе с ожидаемыми характеристиками выполнения;
- будет существовать развернутый конкретный процесс для каждого проекта, приспособленный из стандартного процесса в соответствии с потребностями проекта;
- будет существовать и поддерживаться библиотека информации и данных, связанная с использованием стандартного процесса для конкретных процессов.

Связанные рабочие продукты могут использоваться при рассмотрении потенциальных входов и выходов реализации процесса

организации и обеспечивают руководство для поиска и обеспечения объективного доказательства, обеспечивающее оценку конкретного процесса. Приведенные ниже некоторые рабочие продукты (табл. 4.9–4.14) не должны рассматриваться как контрольный список того, что каждая организация должна иметь. Они должны использоваться как пример и отправная точка для рассмотрения, являются ли рабочие продукты необходимыми и способствуют ли достижению предполагаемой цели процесса.

Таблица 4.9

CUS.1. Приобретение ПС

Вход	Выход
83) Просьба клиента	44) Оценка потребности в продукте
52) Внутренние требования	52) Требования на продукт / услугу
48) Ответ на предложения поставщика	45) Стратегии / план приобретения
49) Запись истории поставщика	47) Просьба о предложении
29) Записи оценки / проверки	21) Результаты анализа
51) Контракт	31) Записи обзора
21) Анализ результатов	51) Контракт
45) Стратегия приобретения	68) Стратегия приемосдаточного испытания

Таблица 4.10

CUS.2. Управление потребностями клиента

Вход	Выход
83) Просьба клиента	83) Просьба клиента
52) Требования клиента	46) Анализ рынка
21) Результаты анализа	87) Механизм связи
22) Запись анализа риска	31) Записи обзора
51) Контракт	44) Оценка потребностей в продукте
96) История изменения	82) Процедуры поддержки клиента
6) Структура прерывания работы	50) Обязательство / соглашение
	52) Требования клиента
	51) Контракт
	95) Контроль изменений
	96) История изменения
	17) План проекта
	87) Механизм связи
	58) Трассируемость записи / распределения
	97) Поправочные действия

Таблица 4.11

ENG.1. Разработка системных требований и проектирование

Вход	Выход
52) Требования клиента	52) Системные требования
52) Эксплуатационные требования	53) Проект системы / архитектура
44) Оценка потребностей продукта	58) Трассируемость записи /
83) Просьба клиента	распределения
94) Просьба изменения	69) Стратегия / план версии
46) Маркетинговый анализ	

Таблица 4.12

ENG.2. Разработка программных требований

Вход	Выход
52) Требования клиента	52) Программные требования
52) Эксплуатационные требования	21) Результаты анализа
44) Оценка потребностей продукта	
83) Просьба клиента	
94) Просьба изменения	
53) Проект системы / архитектуры	
84) Отчет по проблемам	
87) Механизм связи	
101) Проект БД	

Таблица 4.13

SUP.1. Разработка документации

Вход	Выход
52) Требования клиента	52) Требования документации
83) Просьба клиента	16) План
53) Проект системы / архитектуры	18) Данные исполнения процесса
44) Оценка потребностей продукта	105) Документация клиента
3) Описание процесса	31) Записи обзора
30) План обзора	79) Запись поставки
77) Дистрибутивный список	81) Акцептованная запись
78) Инструкции поставки	95) Управление изменением
84) Отчеты по проблемам	96) История изменения
94) Просьба изменения	7) Рабочий продукт
9) Стандарты	
27) критерии качества	

Поля в таблице характеристик рабочего продукта (табл. 4.14) содержат следующую информацию.

Идентифика-
тор рабочего
продукта #

Номер идентификатора для рабочего продукта, кото-
рый используется, чтобы сделать ссылку на рабочий
продукт.

Тип рабоче-
го продукта

Обеспечивает пример типичного имени, связанного с
характеристиками рабочего продукта. Это имя предус-
мотрено как идентификатор типа рабочего продукта,
метода или процесса. Организации могут назвать эти
рабочие продукты другими именами. Имя рабочего
продукта в организации не значимо. Аналогично, орга-
низации могут иметь несколько эквивалентных рабо-
чих продуктов, которые содержат характеристики, оп-
ределенные в одном типе рабочего продукта. Форматы
для рабочих продуктов могут измениться.

Характерис-
тики рабо-
чего продук-
та

Обеспечивает примеры потенциальных характеристик,
связанных с типом рабочего продукта. Эксперт может
найти их в образцах, предусмотренных организаци-
онным модулем.

Таблица 4.14

№ пп.	Тип рабочего продукта	Характеристики рабочего продукта
1	2	3
1	Методоло- гия разра- ботки ПС	Определение подхода / метода, использованного в разработке ПС Идентификация модели жизненного цикла (водопад, спираль, последовательное построение и т. п.), использованного в разработке ПС Обеспечивает описание процесса высокого уровня, деятельности и управления
2	Модель жизненного цикла раз- работки ПС	Описание деятельности высокого уровня выполнялось в каждой фазе жизненного цикла Упорядочение фаз жизненного цикла Идентификация критических зависимостей фаз жизненного цикла Идентификация необходимых входов, выходов каждой фазы жизненного цикла Идентификация ключевых точек решения (этапов) модели Идентификация точек контроля качества в модели

1	2	3
3	Описание процесса	<p>Детальное описание процесса включает:</p> <ul style="list-style-type: none"> – цель процесса; – задачи и действия, которые должны выполняться и упорядочение задач; – критические зависимости между действиями задачи; – ожидаемое время, требуемое для выполнения задачи; – входы/выходы рабочих продуктов <p>Идентифицирует вход процесса и критерии выхода</p> <p>Идентифицирует внутренние и внешние интерфейсы процесса</p> <p>Идентифицирует меры процесса</p> <p>Идентифицирует ожидания качества</p> <p>Идентифицирует функциональные роли и обязанности</p>
4	Процедуры работы, методы	<p>Идентифицирует каждые задания, которые должны выполняться однозначно</p> <p>Каждое задание, упорядоченное порядком выполнения</p> <p>Охват информации поддержки (т. е. команды и установочные параметры и т. п.), которые необходимы для задач</p> <p>Устанавливает правила, которыми персонал будет руководствоваться</p>
5	Спецификация	<p>Идентифицирует задания, которые должны выполняться</p> <p>Идентифицирует начало и необходимую дату завершения заданий</p> <p>Учитывает идентификацию задач и зависимости задач</p> <p>Идентифицирует состояние завершения задачи против запланированной даты</p> <p>Имеет отображение к планируемым данным ресурса</p>
6	Структура прерывания работы	<p>Определяет задачи, которые нужно выполнить</p> <p>Собственные документы задач</p> <p>Документы критических зависимостей между задачами</p> <p>Документы входов и выходов рабочих продуктов</p> <p>Документы критических зависимостей между определенными рабочими продуктами</p>
7	Рабочий продукт	<p>Определяет атрибуты, связываемые с экспонатом выполнения процесса:</p> <ul style="list-style-type: none"> – ключевые элементы, которые должны быть представлены в рабочем продукте; – ожидаемая форма, стиль; – ожидаемый носитель (бумага, электронные) и атрибуты хранения (память)

1	2	3
8	Интерфейс	<p>Определяет связи между двумя продуктами, процессами или задачами процесса</p> <p>Определяет критерии и формат для общих продуктов</p> <p>Определяет критерий критических зависимостей синхронизации или упорядочение последовательности</p>
9	Стандарты	<p>Идентификация к кому/чему они относятся</p> <p>Идентифицируются ожидаемые соответствия</p> <p>Соответствие требованиям может быть продемонстрировано</p> <p>Условия для подгонки или исключения требований</p>
10	Стандарты кодирования	<p>Покрытие для ПС включает:</p> <ul style="list-style-type: none"> – оглашения присваивания имен данным – определяет необходимые языки, компиляторы, системы управления базой данных, и т. п. – формат кода, структуру, необходимые комментарии – стандартные структуры данных, типы, классы – наилучшие методы (практики) – необходимое использование инструментальных средств: словари данных, связанные CASE средства – требование совместимости для существующего программного обеспечения и / или аппаратных средств – соображения безопасности – соображения исполнения – стандартные сообщения об ошибке, коды – стандарты интерфейса: человеко-машинные интерфейсы, внешние интерфейсы системы – периферийное оборудование, аппаратные средства – хранение и поиск исходной программы и объектных модулей – стандарты качества и надежности
11	Оценки	<p>Покрытие для элементов типа: размер, усилия, стоимость, спецификация, ресурсы</p> <p>Оценки реалистичны и достижимы: в соответствии с распределенными ресурсами и в соответствии с историческими записями (где они существуют)</p> <p>Исходные данные нужно делать оценками как доступными и полными</p> <p>Утвержденные исходные данные</p>

1	2	3
12	Цели бизнеса, качества, организационные, обучения, исполнения (эффективности), процесса	Идентифицирует цель, которая будет достигнута Идентифицирует кто ожидается, что достигнет цели Идентифицирует любые пошаговые цели поддержки Идентифицирует любые условия / ограничения Идентифицирует интервал времени для достижения Являются приемлемыми и достижимыми в пределах распределенных ресурсов Являются текущими, установленными для текущего проекта, организации Используются для контроля прогресса Оптимизированы, чтобы поддерживать известные критерии эффективности, планы

Измерение возможности процесса в модели оценки

Действия управления, со связанными показателями атрибута, являются показателями возможности процесса и средствами достижения возможностей, направляемых атрибутами процесса. Подтверждение выполнения действия управления поддерживает решение о степени достижения атрибута процесса. Действия управления связываются с их показателями атрибута, которые являются:

- характеристиками выполнения действия, которые обеспечивают руководство в реализации действия;
- характеристиками ресурса и инфраструктуры, которые обеспечивают механизмы для помощи в управлении процессом;
- связанными процессами в измерения процесса, которые поддерживают действие управления.

Конкретные действия управления связаны с каждым атрибутом процесса. Набор действий управления предназначен для применения всеми процессами в измерении процесса модели. Показатели атрибута помогают устанавливать объективное доказательство того, что действия управления, связанные с атрибутом процесса, выполняются.

Размерность возможности модели оценки состоит из шести уровней возможностей, соответствующих уровням возможности эталонной модели.

Уровень 0: Незавершенный процесс. Процесс не осуществляется или отказывается достигать своей цели. На этом уровне нет атрибутов.

Уровень 1: Выполняемый процесс. Осуществленный процесс достигает своей определенной цели. Следующий атрибут процесса демонстрирует достижение этого уровня – *АП 1.1. Атрибут выполнения процесса.*

Уровень 2: Управляемый процесс. Выполняемый процесс способен получить рабочие продукты определенного качества в определенный промежуток времени и с определенными затратами ресурсов. Следующие атрибуты процесса демонстрируют достижение этого уровня: *АП 2.1. Атрибут управления выполнением* и *АП 2.2. Атрибут управления рабочим продуктом*.

Уровень 3: Установленный процесс. Управляемый процесс выполняется, используя установленное определение процесса (определенный процесс) для всей организации. Следующие атрибуты процесса демонстрируют достижение этого уровня: *АП 3.1. Атрибут определения и преобразования процесса* и *АП 3.2. Атрибут ресурса процесса*.

Уровень 4: Предсказуемый процесс. Установленный процесс может выполняться, следуя прогнозу (внутри определенных ограничений по управлению), в соответствии с бизнес-целями организации. Следующие атрибуты процесса демонстрируют достижение этого уровня: *АП 4.1. Атрибут измерения процесса* и *АП 4.2. Атрибут контроля процесса*.

Уровень 5: Оптимизирующий процесс. Возможна оптимизация выполнения предсказуемого процесса для соответствия текущим и будущим потребностям бизнеса. Следующие атрибуты процесса демонстрируют достижение этого уровня: *АП 5.1. Атрибут изменения (верификация) процесса* и *АП 5.2. Атрибут возможности дальнейшего улучшения*.

Характеристики выполнения действия, характеристики ресурса и инфраструктуры и связанные процессы могут использоваться, когда оцениваются действия управления для конкретной реализации процесса. Эти характеристики и связанные процессы обеспечивают руководство для нахождения объективного доказательства, обеспечивающего эффективную реализацию действия управления.

Таблицы, приведенные в Прил. 2 (без характеристик выполнения действия управления), не должны рассматриваться как контрольные списки того, что каждая организация должна делать или иметь. Эти таблицы являются ориентиром для рассмотрения того, что действия управления действительно выполняются, тем самым способствуя достижению атрибута процесса.

В заключение необходимо отметить, что:

- ПС становятся критическим элементом обеспечения производства как товаров, так и услуг;
- производство ПС становится высокотехнологичной индустрией;
- все большая ориентация производителей на “серийные” ПС.

Ввиду вышеизложенного очевидной становится необходимость стандартного подхода к решению проблемы управления качеством и

оптимизации производственных процессов. Также достаточно очевидным представляется тот факт, что ИСО 9001-94 для решения этой задачи в программной индустрии недостаточно. Для обеспечения потребителей в инструменте для решения вопросов управления качеством ИСО разработан новый стандарт, получивший обозначение ИСО 15504-97 – SPICE.

Проект SPICE был начат ИСО в июле 1991 г. и к настоящему времени объединил лучшие из существующих в мире практик. Архитектура SPICE двухмерна и состоит из:

- категорий процессов – 5, типовых процессов – 29, а также базовых действий – 200;
- уровней возможностей – 6, атрибутов процессов – 9, действий управления – 32.

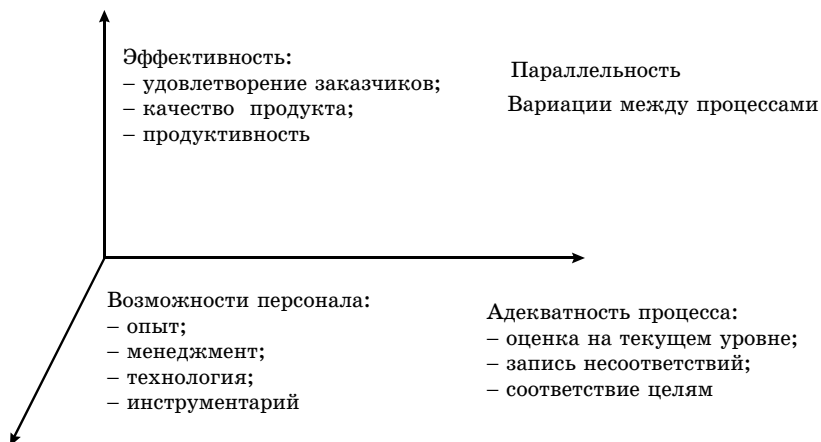


Рис. 4.13. Трехмерная схема улучшения процесса

SPICE предлагает достаточно законченную и подробную модель, предоставляющую пользователям достаточную свободу в выборе путей улучшения процессов разработки ПС. Модель улучшения процессов в SPICE трехмерна (рис. 4.13), где по одной оси откладывается эффективность, по другой – возможности персонала, и по третьей – адекватность процесса. Таким образом, пользователь может выбирать траекторию улучшения процесса в трехмерном пространстве, где улучшения по каждой из осей идут параллельно с улучшениями по другой. Собственно, параллельность не является требованием, это, скорее, рекомендация, позволяющая избежать серьезных перекосов в процессе производства.

Основные термины и определения

В данном приложении приведены основные определения и значения важнейших терминов, установленных в отечественных и международных стандартах. Для этих целей используются ГОСТ 19.004-80, ГОСТ 19.101-77, ГОСТ 19.102-77, ГОСТ 34.003-90, ИСО 9000-3-91, ИСО 2382-1, ИСО 8402, ИСО 15504-97.

Объект – то, что может быть индивидуально описано и рассмотрено. ПРИМЕЧАНИЕ: Объектом может быть, например, деятельность или процесс, продукция, организация, система или отдельное лицо, или любая комбинация из них. [ИСО 8402]

Качество – совокупность характеристик объекта, относящихся к его способности удовлетворить установленные и предполагаемые потребности [ИСО 8402].

ПРИМЕЧАНИЯ: 1. Во многих случаях потребности могут меняться со временем; это предполагает проведение периодического анализа требований к качеству. 2. Обычно потребности переводятся в характеристики на основе установленных критериев. Потребности могут включать, например, такие аспекты, как эксплуатационные характеристики, функциональная пригодность, надежность (готовность, безотказность, ремонтпригодность), безопасность, окружающая среда, экономические и эстетические требования. 3. Для выражения превосходной степени в сравнительном или количественном смысле при проведении технических оценок термин качество не используется изолированно. Чтобы выразить эти значения, должно применяться качественное прилагательное. Например, могут использоваться следующие термины:

– *относительное качество*, когда объекты классифицируются в зависимости от их степени превосходства или в сравнительном смысле;

– *уровень качества* в количественном смысле (применяется при статистическом приемочном контроле) и *мера качества*, когда проводятся точные технические оценки. 4. В некоторых справочных источниках качество обозначается как “пригодность для использования” или “соответствие цели”, или “удовлетворение нужд потребителя”, или “соответствие требованиям”. Все это представляет собой только некоторые стороны качества.

Требования к качеству – выражение определенных потребностей или их перевод в набор количественно или качественно установленных требований к характеристикам объекта, чтобы дать возможность их реализации и проверки [ИСО 8402].

ПРИМЕЧАНИЯ: 1. Существенно, чтобы требования к качеству полностью отражали установленные и предполагаемые потребности потребителя. 2. Термин требования охватывает рыночные и контрактные требования, а также внутренние требования организации. Они могут быть разработаны, детализированы и актуализированы на различных этапах планирования. 3. Заданные количественные требования к характеристикам включают, например, номинальные значения, относительные значения, предельные отклонения и допуски. 4. Требования к качеству должны быть выражены на начальной стадии в функциональных терминах и документально оформлены.

Надежность – собирательный термин, используемый для описания характеристики готовности и влияющих на нее факторов: безотказности, ремонтпригодности и обеспеченности технического обслуживания и ремонта [ИСО 8402].

ПРИМЕЧАНИЯ: 1. Надежность используется только для общих описаний, когда не применяются количественные термины. 2. Надежность является одним из зависящих от времени аспектов качества.

Общее руководство качеством, административное управление качеством – те аспекты общей функции управления, которые определяют политику в области качества, цели и ответственность, а также осуществляют их с помощью таких средств, как планирование качества, управление качеством, обеспечение качества, улучшение качества в рамках системы качества [ИСО 8402].

ПРИМЕЧАНИЯ: 1. Обязанности по общему руководству качеством лежат на всех уровнях управления, но управлять ими должно высшее руководство. В общее руководство качеством вовлекаются все члены организации. 2. При общем руководстве качеством акцент делается на экономические аспекты.

Управление качеством – методы и виды деятельности оперативного характера, используемые для выполнения требований к качеству [ИСО 8402].

Система качества – совокупность организационной структуры, методик, процессов и ресурсов, необходимых для осуществления общего руководства качеством [ИСО 8402].

Программа качества – документ, регламентирующий конкретные меры в области качества, ресурсы и последовательность деятельности, относящейся к специфической продукции, проекту или контракту [ИСО 8402].

Технические условия – документ, устанавливающий требования [ИСО 8402].

Алгоритм — точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату [ГОСТ 19.004-80].

Программа — алгоритм, записанный в форме, воспринимаемой вычислительной машиной [ГОСТ 19.004-80].

Программное обеспечение — совокупность программ и документов на них для реализации целей и задач цифровых электронных вычислительных машин [ГОСТ 19.004-80].

Программное обеспечение (ПО) — интеллектуальный продукт, состоящий из программ, процедур, правил и любой другой связанной с ними документации, относящихся к функционированию системы обработки данных [ИСО 2382-1].

Продукция программного обеспечения — полный набор компьютерных программ, процедур и связанной с ними документации и информации, предназначенный для поставки пользователю [ИСО 9000-3-91].

Программная продукция — результат процесса разработки ПО, т. е. ПО, выпускаемое для использования [ГОСТ Р ИСО/МЭК 9294-93].

Элемент программного обеспечения — какая-либо идентифицируемая часть программного обеспечения на промежуточном или конечном этапе разработки [ИСО 9000-3-91].

Разработка — все виды деятельности, выполняемые для создания продукции программного обеспечения [ИСО 9000-3-91].

Фаза — определенная часть работы [ИСО 9000-3-91].

Проверка (для программного обеспечения) — процесс оценивания продукции данной фазы в целях обеспечения правильности и согласованности в отношении продукции и стандартов, являющихся входными для данной фазы [ИСО 9000-3-91].

Аттестация (для программного обеспечения) — процесс оценивания программного обеспечения в целях обеспечения соответствия установленным требованиям [ИСО 9000-3-91].

Процесс разработки ПО — процесс или набор процессов, которые используются организацией или проектом для планирования, управления, осуществления, проверки, руководства и совершенствования связанных действий ПО [ИСО 15504-9].

Оценка процесса — корректное определение процессов разработки и эксплуатации ПО организации в сравнении с моделью, совместимой с моделью ссылки [ИСО 15504-9].

Улучшение процесса — действие, направленное на изменение процессов организации с целью удовлетворения деловых потребностей организации и более эффективного достижения деловых целей [ИСО 15504-9].

Определение возможности процесса – систематическая оценка и анализ выбранного процесса разработки и эксплуатации ПО в сравнении с целевой возможностью, выполненной с целью идентификации достоинств, недостатков и рисков, связанных с развертыванием процессов для достижения определенного требования [ИСО 15504-9].

Измерение возможности – набор атрибутов процесса, включающие аспекты возможности модели ссылки процессов и возможности процесса. Атрибуты организуются в уровни возможностей, включающие порядковую шкалу возможности процесса [ИСО 15504 -9].

Измерение процесса – набор процессов, включающих функциональные аспекты процессов модели ссылки и возможности процесса. Процессы группируются в категории связанной деятельности [ИСО 15504-9].

Категория процесса – набор процессов, указывающих общую область деятельности. Категории процессов устанавливают пять областей деятельности: определение требований, инжиниринг (проектирование), поддержка, управление и организация [ИСО 15504 -9].

Процесс – комплекс взаимосвязанных действий, которые трансформируют входы в выходы. ПРИМЕЧАНИЕ: Термин действия используется для покрытия ресурсов [ИСО 8402-94, 1.2], [ИСО/МЭК 12207].

Цель процесса – измеряемые цели высокого уровня выполнения процесса и вероятные результаты эффективной реализации процесса [ИСО 15504-9].

Метод – инжиниринг ПО или деятельность управления, которое способствует созданию выхода процесса или расширяет возможность процесса [ИСО 15504-9].

Определенный процесс – определенный набор действий для достижения цели. Определенный процесс может характеризоваться стандартами, процедурами, обучением, инструментальными средствами и методами [ИСО 15504-9].

Атрибут процесса – измеряемая характеристика возможности процесса, применяемая к любому процессу [ИСО 15504-9].

Уровень возможности (процесса) – точка на порядковой шкале с шестью пунктами (возможности процесса), которая определяет возрастающую возможность выполненного процесса. Каждый уровень строится на возможности нижеуказанного уровня.

Стандартный процесс – действующее определение основного процесса, который ведет к созданию общего процесса в организации. Он описывает фундаментальные элементы процесса, включаемые в любой определенный процесс. Он также описывает связь (например,

порядок и интерфейсы) между этими элементами процесса (по аналогии с *определенным процессом*) [ИСО 15504-9].

Поставщик – организация, которая предоставляет продукт клиенту. ПРИМЕЧАНИЯ: В договорной ситуации, поставщик может быть назван “контрагент”. Поставщиком может быть, например, производитель, дистрибьютор, импортер, монтажник или организация услуг. Поставщик может быть или внешним или внутренним по отношению к организации [ИСО 8402].

Компонент – программа, рассматриваемая как единое целое, выполняющая законченную функцию и применяемая самостоятельно или в составе комплекса.

Комплекс – программа, состоящая из двух или более компонентов и/или комплексов, выполняющих взаимосвязанные функции, и применяемая самостоятельно или в составе другого комплекса.

Документ – уникально обозначенный блок информации для использования человеком, такой как отчет, спецификация, руководство или книга [ГОСТ Р ИСО/МЭК 9294-93].

Документация – набор из одного или более связанных документов [ГОСТ Р ИСО/МЭК 9294-93].

ПРИЛОЖЕНИЕ 2

Таблица П.2.1

Категории процессов и возможности процессов

Категория		Процесс		Базовые действия	
Номер	Имя	Номер	Имя	Номер	Имя
1	2	3	4	5	6
CUS	Категория процесса поставщик – клиент				
		CUS.1	Приобретение ПО		
				CUS.1.1	Определение потребности
				CUS.1.2	Определение требований
				CUS.1.3	Подготовка стратегии приобретения
				CUS.1.4	Подготовка заявки для предложений
				CUS.1.5	Выбор поставщика программного продукта
				CUS.1.6	Определение интерфейсов с независимыми агентам и субподрядчиками
				CUS.1.7	Заключение контракта
				CUS.1.8	Поддержка контракта
				CUS.1.9	Приемка поставленного продукта
		CUS.2	Управление потребностями клиента		
				CUS.2.1	Получение требований и просьб клиента
				CUS.2.2	Согласование требований
				CUS.2.3	Установление базиса требований клиента
				CUS.2.4	Управление изменениями требований клиента
				CUS.2.5	Понимание ожиданий клиента
				CUS.2.6	Информирование клиентов

Продолжение табл. П.2.1

1	2	3	4	5	6
		CUS.3	Поставка ПО		
				CUS.3.1	Обсуждение перед заключением контракта
				CUS.3.2	Заключение контракта
				CUS.3.3	Определение интерфейсов между независимыми агентами и субподрядчиками
				CUS.3.4	Разработка системы или программного обеспечения
				CUS.3.5	Обзор разработки с клиентом
				CUS.3.6	Обеспечение обратной связи с клиентом
				CUS.3.7	Поставка и инсталляция программного обеспечения
		CUS.4	Эксплуатация ПО		
				CUS.4.1	Определение эксплуатационных рисков
				CUS.4.2	Выполнение эксплуатационного тестирования
				CUS.4.3	Эксплуатация программного обеспечения
				CUS.4.4	Разрешение эксплуатационных проблем
				CUS.4.5	Управление запросами пользователей
				CUS.4.6	Регистрация временных работ
				CUS.4.7	Мониторинг производительности системы и обслуживание
		CUS.5	Сервисное обслуживание клиента		
				CUS.5.1	Обучение клиентов
				CUS.5.2	Определение поддержки продукта
				CUS.5.3	Контроль выполнения

1	2	3	4	5	6
				CUS.5.4	Определение уровня удовлетворенности клиента
				CUS.5.5	Сравнение с конкурентами
				CUS.5.6	Сообщение положительных отзывов клиентов
ENG	Категория процесса инжиниринга				
		ENG.1	Разработка и проектирование системных требований		
				ENG.1.1	Определение системных требований
				ENG.1.2	Анализ системных требований
				ENG.1.3	Описание архитектуры системы
				ENG.1.4	Распределение требований
				ENG.1.5	Определение стратегии версии
				ENG.1.6	Передача системных требований
		ENG.2	Разработка требований к ПО		
				ENG.2.1	Выявление требований к ПО
				ENG.2.2	Определение влияния операционной среды
				ENG.2.3	Оценка требований вместе с заказчиком (клиентом)
				ENG.2.4	Определение стратегии выпуска версии ПО
				ENG.2.5	Изменение требований для следующей итерации
				ENG.2.6	Связь с требованиями к ПО
		ENG.3	Проектирование ПО		
				ENG.3.1	Разработка проекта архитектуры ПО
				ENG.3.2	Проектирование интерфейсов
				ENG.3.3	Разработка детального проекта
				ENG.3.4	Установление трассируемости

1	2	3	4	5	6
		ENG.4	Реализация проекта ПО		
				ENG.4.1	Разработка модулей ПО
				ENG.4.2	Разработка процедур верификации модулей
				ENG.4.3	Проверка модулей ПО
		ENG.5	Интеграция и тестирование ПО		
				ENG.5.1	Определение стратегии регрессионного тестирования
				ENG.5.2	Сборка совокупностей модулей ПО
				ENG.5.3	Разработка тестов для совокупностей
				ENG.5.4	Тестирование совокупностей ПО
				ENG.5.5	Интеграция совокупностей ПО
				ENG.5.6	Разработка тестов для ПО
				ENG.5.7	Тестирование интегрированного ПО
		ENG.6	Интеграция и тестирование системы		
				ENG.6.1	Создание агрегатов системных элементов
				ENG.6.2	Разработка тестов для агрегатов
				ENG.6.3	Тестирование агрегатов системы
				ENG.6.4	Разработка тестов для системы
				ENG.6.5	Тестирование интегрированной системы
		ENG.7	Поддержка системы и ПО		
				ENG.7.1	Определение требований по сопровождению
				ENG.7.2	Анализ проблем пользователя и усовершенствования
				ENG.7.3	Определение модификаций для следующего обновления

1	2	3	4	5	6
				ENG.7.4	Реализация и модификация тестов
				ENG.7.5	Обновление системы у пользователей
				ENG.7.6	Удаление системы у пользователей
SUP	Категория процесса поддержки				
		SUP.1	Разработка документации		
				SUP.1.1	Определение требований к документации
				SUP.1.2	Разработка документов
				SUP.1.3	Проверка документов
				SUP.1.4	Распространение документации
				SUP.1.5	Поддержка документации
		SUP.2	Управление конфигурацией		
				SUP.2.1	Установка библиотеки конфигурации
				SUP.2.2	Определение параметров конфигурации
				SUP.2.3	Поддержка описания параметров конфигурации
				SUP.2.4	Управление запросами на изменение
				SUP.2.5	Возможность управления
				SUP.2.6	Сборка конечных продуктов
				SUP.2.7	Поддержка истории параметров конфигурации
				SUP.2.8	Отчет о состоянии конфигурации
		SUP.3	Выполнение гарантии качества		
				SUP.3.1	Выбор критериев качества
				SUP.3.2	Определение записей качества
				SUP.3.3	Гарантия качества действий инжиниринга ПО
				SUP.3.4	Гарантия качества рабочих продуктов

Продолжение табл. П.2.1

1	2	3	4	5	6
				SUP.3.5	Сообщение о результатах
				SUP.3.6	Обработка отклонений
				SUP.3.7	Обеспечение независимости ресурсов и организации по гарантии качества
		SUP.4	Верификации рабочего продукта		
				SUP.4.1	Выбор рабочих продуктов
				SUP.4.2	Определение методов и методик верификации
				SUP.4.3	Определение критериев завершения
				SUP.4.4	Руководство верификацией
				SUP.4.5	Документирование элементов действий
				SUP.4.6	Отслеживание элементов действий
		SUP.5	Проверка правильности рабочего продукта		
				SUP.5.1	Определение рабочих продукты, подлежащих проверке
				SUP.5.2	Определение задачи и методики проверки правильности
				SUP.5.3	Определение критериев проверки
				SUP.5.4	Выполнение действий по аттестации
				SUP.5.5	Документирование проблем (аномалий)
				SUP.5.6	Отслеживание проблем
		SUP.6	Выполнение совместных обзоров		
				SUP.6.1	Определение совместных обзоров
				SUP.6.2	Подготовка обзоров с клиентами
				SUP.6.3	Проведение совместных управленческих обзоров
				SUP.6.4	Проведение совместных технических обзоров

Продолжение табл. П.2.1

1	2	3	4	5	6
				SUP.6.5	Поддержка приемочного обзора клиента
				SUP.6.6	Выполнение совместной оценки процесса
		SUP.7	Выполнение аудита		
				SUP.7.1	Определение требований к аудиту
				SUP.7.2	Подготовка к аудиту
				SUP.7.3	Проведение аудита
				SUP.7.4	Выполнение корректировочных действий
		SUP.8	Разрешение проблем		
				SUP.8.1	Сообщение деталей проблемы
				SUP.8.2	Ранжирование проблем
				SUP.8.3	Определение решений
				SUP.8.4	Сообщение путей решения проблемы
				SUP.8.5	Исправление дефекта
				SUP.8.6	Распространение исправлений
				SUP.8.7	Анализ тенденций проблем
MAN	Категория процесса управления				
		MAN.1	Управление проектом		
				MAN.1.1	Определение сферы работ
				MAN.1.2	Определение стратегии разработки
				MAN.1.3	Выбор модели жизненного цикла ПО
				MAN.1.4	Разработка оценок проекта
				MAN.1.5	Разработка структуры распределения работ
				MAN.1.6	Определение требований к инфраструктуре

Продолжение табл. П.2.1

1	2	3	4	5	6
				MAN.1.7	Определение графика проекта
				MAN.1.8	Распределение ответственности
				MAN.1.9	Определение плана проекта
				MAN.1.10	Отслеживание исполнения
				MAN.1.11	Оценка выполнения проекта
		MAN.2	Управление качеством		
				MAN.2.1	Установление целей качества
				MAN.2.2	Определение метрик качества
				MAN.2.3	Определение действий качества
				MAN.2.4	Выполнение действий качества
				MAN.2.5	Оценка качества
				MAN.2.6	Применение корректировочного действия
		MAN.3	Управление риском		
				MAN.3.1	Установление сферы управления риска
				MAN.3.2	Определение рисков
				MAN.3.3	Анализ и расположение рисков по приоритетам
				MAN.3.4	Разработка стратегии управления рисками
				MAN.3.5	Определение метрик риска
				MAN.3.6	Выполнение стратегии управления рисками
				MAN.3.7	Оценка результатов стратегии управления рисками
				MAN.3.8	Применение корректирующего действия

1	2	3	4	5	6
		MAN.4	Управление субподрядчиками		
				MAN.4.1	Определение утверждения работы
				MAN.4.2	Квалификация потенциальных субподрядчиков
				MAN.4.3	Выбор субподрядчика
				MAN.4.4	Поддержка связи
				MAN.4.5	Поддержка связи
				MAN.4.6	Оценка согласие
				MAN.4.7	Оценка качества субподрядчика
ORG	Категория процесса организации				
		ORG.1	Бизнес-инжиниринг		
				ORG.1.1	Установление стратегической точки зрения
				ORG.1.2	Развертывание точки зрения
				ORG.1.3	Установление культуры качества
				ORG.1.4	Формирование интегрированных команд
				ORG.1.5	Обеспечение стимулирования
				ORG.1.6	Определение планов карьеры
		ORG.2	Определение процесса		
				ORG.2.1	Определение целей
				ORG.2.2	Определение текущих действий, роли, прав и обязанностей
				ORG.2.3	Определение входов и выходов
				ORG.2.4	Определение критериев входа и выхода
				ORG.2.5	Определение критериев входа и выхода
				ORG.2.6	Идентификация внешних интерфейсов
				ORG.2.7	Идентификация внутренних зависимостей

1	2	3	4	5	6
				ORG.2.8	Определение мер процесса
				ORG.2.9	Документирование стандартного процесса
				ORG.2.10	Установление стратегии
				ORG.2.11	Установление ожиданий выполнения
				ORG.2.12	Развертывание процесса
		ORG.3	Усовершенствование процесса		
				ORG.3.1	Определение возможностей усовершенствования
				ORG.3.2	Определение сферы действий усовершенствования
				ORG.3.3	Понимание процесса
				ORG.3.4	Определение усовершенствований
				ORG.3.5	Распределение усовершенствований по приоритетам
				ORG.3.6	Определение мер воздействия
				ORG.3.7	Изменение процесса
				ORG.3.8	Подтверждение усовершенствования
				ORG.3.9	Разворачивание усовершенствований
		ORG.4	Обеспечение квалифицированными человеческими ресурсами		
				ORG.4.1	Определение потребностей в людских ресурсах
				ORG.4.2	Разработка или приобретение средств обучения
				ORG.4.3	Тренировка персонала
				ORG.4.4	Вербовка квалифицированного штата
				ORG.4.5	Оценка эффективности штата
				ORG.4.6	Обеспечение обратной связи выполнения
				ORG.4.7	Поддержка записей штата

Продолжение табл. П.2.1

1	2	3	4	5	6
				ORG.4.8	Определение команд проекта
				ORG.4.9	Наделение полномочиями команд проекта
				ORG.4.10	Поддержка взаимодействия команд проекта
		ORG.5	Обеспечение инфраструктуры инжиниринга ПО		
				ORG.5.1	Определение требований среды инжиниринга ПО
				ORG.5.2	Обеспечение среды инжиниринга ПО
				ORG.5.3	Обеспечение поддержки разработчиков
				ORG.5.4	Сопровождение среды инжиниринга ПО
				ORG.5.5	Предоставление рабочей обстановки, благоприятной продуктивному исполнению
				ORG.5.6	Гарантия целостности данных и безопасности
				ORG.5.7	Обеспечение средствами удаленного доступа
				ORG.5.8	Определение организационной стратегии повторного использования
				ORG.5.9	Установление библиотеки повторных использований
				ORG.5.10	Интеграция повторных использований в жизненный цикл

Таблица П.2.2

Выполнение процесса – до какой степени сводится выполнение процесса, использующего набор действий, которые инициализированы и сопровождаются, используя определенные входные рабочие продукты, которые являются адекватными в удовлетворении цели процесса	
Действия управления	Гарантировать, что базовые действия выполняются для удовлетворения цели процесса
Характеристики ресурсов и инфраструктуры для ДУ	См. Изменение процесса
Связанные процессы для ДУ	Это действие относится к каждому процессу в пределах сферы оценки

Таблица П.2.3

Управление выполнением – до какой степени сводится управление планированием и выполнением процесса, чтобы получить рабочие продукты в установленное время и с требуемыми затратами				
Действия управления	Определить требования к ресурсу, чтобы дать возможность планировать и отслеживать процесс	Распланировать выполнение процесса, определяя действия и размещения ресурсов согласно требованиям	Выполнить определенные действия для достижения цели процесса	Управлять выполнением действий, чтобы произвести рабочие продукты в установленное время с требуемыми затратами ресурсов
Характеристики ресурсов и инфраструктуры для ДУ	Методы и инструментальные средства (ИС) оценки производительности Модели риска, средства идентификации и знание предотвращения ИС планирования проекта (которые поддерживают выбранную технологию) ИС контроля/управления проектом Статистические данные выполнения процесса	Технологии PERT/CPM ИС планирования проекта (которые поддерживают выбранные технологии) ИС контроля/управления проектом Средства, процедуры и ИС связи	Планы периодического обзора проекта ИС проверки процесса/плана ИС управление последовательностью операций Механизмы связи	ИС связи Отчет проблем и база данных поправочных действий Собрание механизмов Аудит команд/ресурсов
Связанные процессы для ДУ	CUS.2, MAN.1, MAN.2, MAN.3	MAN.1, MAN.2, MAN.3, MAN.4, ORG.4, ORG. 5	MAN.1, SUP.3	MAN.1, MAN.2, MAN.3, SUP.7, SUP.8

Таблица П.2.4

Управление рабочим продуктом – до какой степени сводится управление выполнением процесса, чтобы произвести рабочие продукты, которые отвечают функциональным и нефункциональным требованиям, соответствуют входным данным и задачам обеспечения качества, и процесс создания которых проконтролирован и задокументирован

Действия управления	Определить требования по целостности и качеству рабочих продуктов	Определить действия, необходимые для достижения требований по целостности и качеству рабочих продуктов	Управлять конфигурацией рабочих продуктов, чтобы убедиться в их целостности	Управлять качеством рабочих продуктов для соответствия его функциональным и нефункциональным требованиям
Характеристики ресурсов и инфраструктуры для ДУ	Цели и политика качества Элементы системы качества Система управления конфигурацией	Библиотека или другие источники методов качества и принципов управления конфигурации Системы / инструментальные средства управления конфигурации Инструментальные средства документирования	БД отчетов о проблемах Инструментальные средства управления конфигурацией Хранилище рабочих продуктов	Трассируемость требований клиента Контрольный лист Инструментальные средства, формы БД отчетов о проблемах
Связанные процессы для ДУ	CUS.2, SUP.2, SUP.3, MAN.2	SUP.1, SUP.2, SUP.3	SUP.2	MAN.2, SUP.3, SUP.4, SUP.5, SUP.7

Определение и преобразование процесса – до какой степени ведется выполнение процесса в виде преобразованного экземпляра стандартного определения процесса. Стандартный процесс отвечает определенным бизнес-целям организации

Действия управления	Получить определение стандартного процесса из множества возможных в организации, которое соответствует целям процесса и бизнес-целям организации	Преобразовать стандартный процесс для получения определенного процесса, соответствующего контексту процесса	Выполнить несколько раз определенный процесс, чтобы последовательно достичь цели процесса и поддерживать бизнес-цели организации	Обеспечить обратную связь со стандартным процессом исходя из опыта использования определенного процесса
Характеристики ресурсов и инфраструктуры для ДУ	Библиотека процесса ПО Инструментальные средства моделирования процессов БД метрик и инструментальные средства Международные стандарты Инструментальные средства управления конфигурацией и документирования	Инструментальные средства моделирования процессов Инструментальные средства документирования Инструментальные средства управления конфигурацией	Инструментальные средства соответствующего процесса, в зависимости от его конкретной цели	Библиотека информации и данных, связанная с использованием стандартного процесса Механизмы сбора данных Инструментальные средства управления экспериментальными данными Инструментальные средства и методы анализа
Связанные процессы для ДУ	ORG.2, MAN.1	CUS.2, ORG.2, MAN.1	MAN.1, ORG.2	MAN.1, ORG.2, ORG.3, SUP.8

Таблица П.2.6

Ресурс процесса – до какой степени сводится определение ресурса и инфраструктуры процесса, которые поддерживают выполнение определенного процесса для соответствия установленным целям				
Действия управления	Определить компетентность людских ресурсов, требуемых для поддержки выполнения определенного процесса	Определить требования к инфраструктуре процесса для поддержки выполнения определенного процесса	Обеспечить необходимую квалификацию людских ресурсов	Обеспечить соответствующую инфраструктуру процесса согласно определенным потребностям процесса
Характеристики ресурсов и инфраструктуры для ДУ	Схема классификации типовой компетенции, связанной со стандартным процессом БД способностей людских ресурсов	Организационная среда инжиниринга ПО	Процедуры выбора людских ресурсов Материалы формального курса Написанные материалы проекта Средства работы в режиме On-line БД способностей Программа подготовки Записи сверхурочного времени	Библиотеки повторного использования Продукт / инструментальное средство эталонного тестирования Тенденции технологии
Связанные процессы для ДУ	ORG.2, ORG.4	ORG.2, ORG.5	ORG.4	ORG.5

Измерение процесса — хорошо ли производится определение набора данных и измерений, которые поддерживают возможность выполнения любого процесса, чтобы реально достичь определенных бизнес-целей организации				
Действия управления	Определить цели процесса и связать с ними измерения, которые соответствуют бизнес-целям организации	Обеспечить соответствие ресурсов и инфраструктуры для сбора данных	Выбрать соответствующие данные измерений из реализации определенного процесса	Оценить достижимость целей процесса на основании сравнения записанных измерений
Характеристики ресурсов и инфраструктуры для ДУ	Показатели управления верхнего уровня (стоимость, время, надежность, прибыль, риск) Библиотека или другие источники методов измерения Статистический метод контроля процесса в определении возможности процесса	Записи персонала и распределения ресурсов Хранилище Подготовка Исполнители процесса подготовлены в статистическом контроле процесса	Записи качества Инструментальные средства накопления и обработки данных Отчеты статуса Инструментальные средства управления последовательностью операций Система записи / трассировки дефектов	Инструментальные средства управления процессом, измерения продукта и статистического анализа Отчеты статуса и отклонений Показатели управления верхнего уровня Системы трассировки с вложенными средствами контроля статистического процесса
Связанные процессы для ДУ	MAN.1, MAN.2, MAN. 3, ORG.1	MAN.1, ORG.4, ORG. 5	MAN.1, MAN.2	MAN.1, MAN.2, MAN. 3, ORG.5, SUP.3, SUP.8

Таблица П.2.8

Контроль процесса – до какой степени сводится использование измерений для управления выполнением процесса, чтобы любой экземпляр процесса мог реально достичь определенной цели

Действия управления	Определить метод анализа и контроля в соответствии с контекстом процесса	Обеспечить соответствие ресурсов и инфраструктуру для анализа и контроля процесса	Проанализировать достаточные меры, чтобы определить параметры управления процесса	Определить отклонения и произвести необходимые действия по контролю для поддержания контроля процессом
Характеристики ресурсов и инфраструктуры для ДУ	Библиотека методов контроля процесса или другие средства Инструментальные средства контроля процесса Инструментальные средства оценки процесса Инструментальные средства статистического анализа	Записи персонала и распределения ресурсов Хранилище Программа обучения Урсы обучения Инструментальные средства статистического анализа	Методы статистического анализа Каркас оценки процесса разработки ПО Инструментальные средства контроля процесса Инструментальные средства оценки возможности процесса	Хранилище результатов выполнения / возможности процесса Каркас оценки процесса разработки ПО Инструментальные средства управления процессом Отчеты статуса и отклонений Обучающие системы
Связанные процессы для ДУ	MAN.2, ORG.3	ORG. 4, ORG. 5	MAN.1, MAN.2	ORG.1, ORG.3

Изменение (верификация) процесса – до какой степени сводится использование изменений для гарантии того, что содержание, выполнение и управление процесса, или части процесса, изменяются по контролируемому и предсказуемому закону, позволяющему лучше достичь целей процесса

Действия управления	Определить и согласовать изменения в определении стандартного процесса на основе количественного понимания процесса	Обеспечить соответствие ресурсов для эффективной реализации согласованных изменений в ранее определенных процессах	Осуществить указанные изменения в ранее определенных процессах для достижения ожидаемого результата	Подтвердить эффективность изменения процесса на основании сопоставления реального выполнения с целями процесса и бизнес-целями
Характеристики ресурсов и инфраструктуры для ДУ	Обязательство управления Механизмы управления изменением Инструментальные средства моделирования и определения процесса БД метрик процесса Инструментальные средства анализа Планируемые периодические обзоры Организация обучения Опытный и испытанный персонал Механизмы связи для сообщения и обзоров изменений	Организация обучения Обязательство управления Планируемые периодические обзоры Опытный и испытанный персонал БД инструментальных средств Инструментальные средства библиотек и обработки текста	Обязательство управления Механизмы контроля и управления изменением Инструментальные средства моделирования и определения процесса Инструментальные средства связи БД метрик процесса Планируемые периодические обзоры	Инструментальные средства анализа БД метрик процесса Бизнес-цели на подробном уровне для проведения анализа и подтверждение правильности действий Механизм обратной связи всем, кому необходимо знать результаты подтверждения правильности Механизм связи для гарантии входа в определение стандартного процесса (ORG.2) Записи, включающие разделы, указывающие действия по изменению
Связанные процессы для ДУ	MAN.2, ORG.2, ORG.3, ORG.4	ORG.2, ORG.3, ORG.4, ORG.5	ORG.2, ORG.3, ORG.4, ORG.5	ORG.2, ORG.3

Таблица П.2.10

Возможность дальнейшего улучшения (подтверждение) – до какой степени сводится выполнение изменений в процессе, чтобы гарантировать возможность дальнейших улучшений для полного соответствия бизнес-целям организации

Действия управления	Определить возможности улучшения процесса в будущем. Конкретизируйте и систематизируйте предложения по дальнейшему улучшению	Установить стратегию реализации, основанную на выявлении возможностей для улучшения выполнения процесса согласно его бизнес-целям	Реализовать изменения в выбранных областях преобразованного процесса в соответствии со стратегией реализации	Подтвердить эффективность изменений процесса на основании сопоставления действительного выполнения с целями процесса, бизнес-целями и обратной связи с определением стандартного процесса
Характеристики ресурсов и инфраструктуры для ДУ	Участие руководства высшего звена Стадии удовлетворения клиента Использование стандартов и промышленных БД Инновационные механизмы Ранняя идентификация новых технологий R&D – деятельность в связанных областях	Участие и обязательство управления Квалифицированный и опытный персонал Организирующий подготовку	Квалифицированный и опытный персонал Организирующий подготовку Управление участием в действиях по реализации улучшения процесса	Обязательство управления Квалифицированный и опытный персонал Планируемые периодические обзоры Организирующий подготовку
Связанные процессы для ДУ	CUS.5, ORG.1, ORG.3, SUP.8	ORG.1, ORG.3	ORG.1, ORG.3	ORG.1, ORG.2, ORG.3

Руководство по применению характеристик качества

Стандарт ГОСТ Р ИСО/МЭК 9126-93 применяется для установления требований к качеству программного средства и оценивания (измерения, ранжирования и оценки) программных продуктов, включая:

- определение требований к качеству программной продукции;
- оценивание технических требований к ПС при контроле за тем, чтобы требования качества были удовлетворены в процессе разработки;
- описание признаков и свойств (атрибутов) внедренного ПС;
- оценивание разработанного ПС перед его поставкой;
- оценивание разработанного ПС перед приемкой.

Существуют только несколько общепринятых метрик для характеристик, описанных в данном стандарте. Организации, занимающиеся разработкой ПС, могут устанавливать свои собственные модели процесса оценивания, методы формирования и проверки метрик, связанных с этими характеристиками, для охвата различных областей применения и стадий жизненного цикла.

При использовании шести характеристик качества (функциональные возможности, надежность, практичность, эффективность, сопровождаемость, мобильность) в целях описания и оценивания также необходимо установить уровни ранжирования и критерии конкретно для данной организации или для данного применения, или для того и другого.

Важность каждой характеристики меняется в зависимости от класса ПС (ПС для критических систем, ПС для систем реального времени и т.д.) и от принятых точек зрения (представление о качестве с позиции пользователя, с позиции разработчика, с позиции руководителя).

Пользователи заинтересованы в применении ПС, его производительности и результатах использования. Они оценивают ПС без изучения его внутренних аспектов или того, как оно создавалось.

Процесс создания требует от пользователя и разработчика использования одних и тех же характеристик качества ПС, так как они применяются для установления требований и приемки.

Так как разработчики отвечают за создание ПС, которое должно удовлетворять требованиям качества, они заинтересованы в качестве промежуточной продукции так же, как и в качестве конечной продукции. Для того чтобы оценить качество промежу-

точной продукции на каждой фазе жизненного цикла, разработчики должны использовать различные метрики для одних и тех же характеристик, потому что одни и те же метрики неприменимы для всех фаз жизненного цикла.

Схема на рис. П.3.1 отражает основные этапы, требуемые для оценивания качества ПС, начиная с характеристик качества, приведенных в данном стандарте.



Рис. П.3.1. Модель процесса оценивания

Процесс состоит из трех стадий: установления требований к качеству, подготовки к оцениванию и процедуры оценивания. Этот процесс применим к любой фазе жизненного цикла для каждого компонента программной продукции.

Установление требований к качеству. Целью этой стадии является установление требований в терминах характеристик качества и возможных комплексных показателей (подхарактеристик). Требования выражают потребность внешнего окружения для рассматриваемой программной продукции и должны быть определены до начала разработки. Так как программная продукция разделяется на компоненты, требования к продукции в целом отличаются от требований для отдельных компонентов.

Подготовка к оцениванию. Цель – подготовка основы для оценивания.

1. Выбор метрик качества. Способ, которым определялись характеристики качества, не допускает их непосредственного измерения. Существует потребность в установлении метрик, которые соотносятся с характеристиками программной продукции. Каждый количественный признак и каждое количественно оцениваемое взаимодействие программной продукции с его окружением, которые соотносятся с характеристикой, могут быть приняты в качестве метрики.

2. Определение уровней ранжирования. Количественные признаки могут быть измерены, используя метрики качества. Измеренное значение (результат) отображается в масштабе. Данное значение не показывает уровень удовлетворения требований. Для этой цели данные шкалы должны быть разделены на диапазоны, соответствующие различным степеням удовлетворения требований (рис. П.3.2). Так как качество относится к конкретным потребностям, общие уровни ранжирования невозможны. Они должны быть определены для каждого конкретного оценивания.

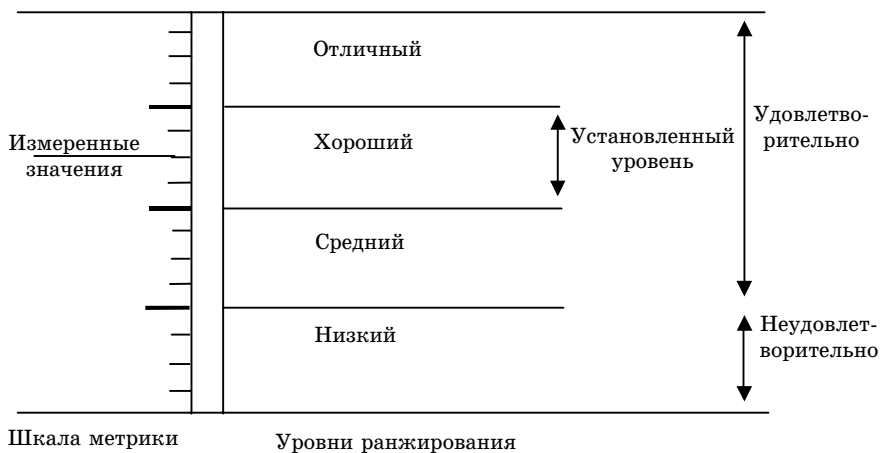


Рис. П.3.2. Измеренное значение и установленный уровень

3. Определение критерия оценки. Для определения качества продукции результаты оценивания различных характеристик должны быть подытожены. Оценщик должен подготовить для этого процедуры, используя, например, таблицы решений или средние взвешенные. Процедуры обычно включают другие аспекты (например, сто-

имость и время), которые способствуют оценке качества программной продукции в конкретных условиях.

Процедура оценивания. Данная стадия модели процесса оценивания уточняется по трем этапам.

1. Измерение. Для измерения выбранные метрики применяются к программной продукции. Результатом являются значения в масштабе метрик.

2. Ранжирование. На этом этапе устанавливаются уровень ранжирования для измеренного значения.

3. Оценка. На этом этапе обобщается множество установленных уровней. Результатом является заключение о качестве программной продукции. Впоследствии обобщенное качество сравнивается с другими факторами. Окончательное решение руководство принимает на основе критерия управляемости. Результатом является решение руководства по приемке или отбраковке, по выпуску или невыпуску программной продукции.

Библиографический список

1. *Богословская Н. В.* и др. Системы автоматизации разработки программного обеспечения: Учеб. пособие / СПб.: СПВУРЭ ПВО, 1996. 86 с.
2. *Буч. Г.* Объектно-ориентированное проектирование с примерами приложения: Пер. с англ. М.: Изд-во Бином, СПб.: Невский диалект, 1998. 560 с.
3. *Гантер Р.* Методы управления проектированием программного обеспечения: Пер. с англ. М.: Мир, 1981. 392 с.
4. Инженерное проектирование программного обеспечения / Пер. с англ. *Б. Боем.* М.: Радио и связь, 1985. 240 с.
5. *Кальянов Г. Н.* CASE структурный системный анализ (автоматизация и применение). М.: ЛОПИ, 1996. 242 с.
6. Software cosiderations in airborne system and equipment certification. RTCA/EROCAE, 1992. 94 p.

Оглавление

Предисловие	3
Глава 1	
Стандарты в области обеспечения качества программных систем	6
1.1. Основные положения стандартов серии ИСО 9000	6
1.2. Руководящие положения по административному управлению качеством и элементы системы качества	14
1.3. Применение ИСО 9001 при разработке программного обеспечения	22
1.4. Показатели качества программных средств в ГОСТ 28195-89 и ГОСТ Р ИСО/МЭК 9126-93	33
1.5. Модели и метрики оценки качества программного средства	41
Глава 2	
Стандарты, определяющие жизненный цикл программных средств	50
2.1. Модели жизненного цикла программных средств	50
2.2. Стадии разработки программных средств, регламентированных ГОСТами	55
2.3. Жизненный цикл разработки ПС с повышенными требованиями к безопасности системы	67
2.4. Процессы жизненного цикла разработки ПС	86
2.5. Сравнительный анализ жизненных циклов программных средств	100
Глава 3	
Документация и ее роль в обеспечении качества жизненного цикла ПС	104
3.1. Документация и ее роль в обеспечении качества	104
3.2. Требования стандартов к программной документации ..	113
Глава 4	
Оценка процесса разработки программного средства	131
4.1. История проекта SPICE	131
4.2. Основные положения	138
4.3. Эталонная модель и выполнение оценки	145
4.4. Модель оценки	162
Приложение 1	179
Приложение 2	184
Приложение 3	204
Библиографический список	208

Учебное издание

Богданов Дмитрий Валерьевич
Фильчаков Владимир Васильевич

**СТАНДАРТИЗАЦИЯ
ЖИЗНЕННОГО ЦИКЛА И КАЧЕСТВА
ПРОГРАММНЫХ СРЕДСТВ**

Учебное пособие

Редактор *А. В. Семенчук*
Компьютерная верстка *А. Н. Колешко*

Лицензия ЛР №020341 от 07. 05. 97. Сдано в набор 27.03.00. Подписано к печати 21.03.00.
Формат 60×84 1/16. Бумага тип. №3. Печать офсетная. Усл. печ. л. 12,2. Усл. кр. -отт. 12,3.
Уч. -изд. л. 13,3. Тираж 100 экз. Заказ №

Редакционно-издательский отдел
Сектор компьютерно-издательских технологий
Отдел оперативной полиграфии
СПбГУАП
190000, Санкт-Петербург, ул. Б. Морская, 67