

Глава 3. ПРИНЦИПЫ СТАНДАРТИЗАЦИИ, КАЧЕСТВА РАЗРАБОТКИ И КОММЕРЦИАЛИЗАЦИИ ПРОГРАММНЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ (В.И. Кияев)

3.1. Принципы стандартизации и реализация качества разработки программного продукта

3.1.1. Предпосылки развития требований к качеству программного продукта

Качество и эффективность программного обеспечения, работающего в государственных, научных, социальных и бизнес-структурах, служащего основой всемирной сети и разнообразных информационных систем, являются критически важными факторами. Сегодня деятельность многих организаций, предприятий и, особенно, высокотехнологичных компаний напрямую зависит от качественной обработки информации соответствующими компьютерными системами. Разработка программного обеспечения является сложным и многоаспектным процессом, в котором участвует большое количество специалистов различных профилей и различной квалификации. Кроме того, в процессе разработки ПО сплетаются множество технических, технологических и управленческих проблем. От их адекватного разрешения зависят успех проекта, качество произведенного продукта и, как следствие, рейтинг компании и конкурентоспособность на рынке программных средств.

Некачественное программное обеспечение может привести к серьезным потерям и затратам на восстановление утерянных данных и программной системы в целом. Даже если и удастся избежать чрезвычайных происшествий, это наносит ущерб эффективности работы, что особенно негативно сказывается на коммерческих предприятиях, поскольку снижает их конкурентоспособность. Предприятия наиболее уязвимы в период внедрения и использования новых программных продуктов и проведения реинжиниринга процессов. Именно в это время ущерб от некачественного программного обеспечения наиболее вероятен [1].

По известным данным исследования Тома де Марко (США):

- 15% всех программных проектов так и не достигли своего завершения;

- превышение стоимости проектов на 100–200% является обычным явлением;
- превышение стоимости на 30% в программной индустрии считается настоящим успехом.

Эти исследования проведены в конце прошлого века, однако серьезность проблем к настоящему времени не только не уменьшилась, но по ряду проблем существенно возросла – особенно в России.

Проблемы, выявленные в ходе 151 оценки различных программных проектов в США:

- помехи со стороны правительства – 10%;
- технологические факторы – 12%;
- внешние поступления – 12%;
- оборудование – 15%;
- недостаточность контроля – 35%;
- недостаточность/недостатки планирования производственных процессов – 55%.

Выдержка из статьи «Software's Chronic Crisis», появившейся в журнале «Scientific American», через 12 лет после опубликования этого исследования (автор W. Wayt Gibbs, сентябрь 1994): «Исследования показали, что на каждые шесть крупных систем программного обеспечения, запущенных в действие, приходится две (т.е. 30-35%) таких, разработка которых была прекращена из-за невозможности добиться удовлетворительного функционирования. Средний проект разработки программного обеспечения затягивается на половину первоначально запланированного срока, крупные проекты и того хуже. Три четверти всех больших систем либо выполняют не все функции, которые на них возлагались, либо не используются вовсе».

По данным Департамента по торговле и промышленности Великобритании (DTI), при внедрении проектов информационных технологий на предприятиях потери из-за некачественного программного обеспечения составляют в среднем около 20% от общего объема потерь. По разным оценкам, аналогичный показатель для России достигает величины от 30 до 50%.

Может показаться, что управлять количеством ошибок в коде невозможно и их число зависит только от способностей конкретного разработчика. Это не совсем так. В конце 80-х – начале 90-х годов в программной индустрии был проведен ряд серьезных исследований эффективности труда программистов. Вот дан-

ные американского Института программной инженерии (Software Engineering Institute – SEI) по исследованиям, выполненным под руководством Уоттса Хемффри (Watts Humphrey). Профессиональные программисты со стажем 10 и более лет на 1000 строк кода (СК – строка исходного текста, которая содержит часть программной инструкции) допускают в среднем 131,3 ошибки. Следовательно, большая система размером миллион СК может содержать 100 тысяч ошибок! Если транслятор имеет развитую систему предупреждений, то до 50% из них выявляется на этапе компиляции.

На этапе тестирования отдельных модулей (типичный программный модуль по определению SEI содержит от 50 до 5 тысяч СК) обнаруживается половина оставшихся ошибок. Получается, что перед этапами внедрения и комплексного тестирования в продукте еще скрывается 25 тыс. ошибок. Для их устранения на заключительных этапах тратится от 10 до 40 человеко-часов на ошибку, т. е. *на доведение продукта до идеального состояния потребуется 125 человеко-лет работы!* Это, конечно, сильное преувеличение, но суть именно такова – на исправление ошибок за весь период сопровождения продукта суммарные затраты времени (в человеко-месяцах) могут быть весьма значительными!

Можно привести другой пример. Типичный небольшой продукт имеет объем до 50 тысяч СК. Пусть его создают 5 программистов, делая при этом 100 ошибок на тысячу СК. 50% ошибок выявляется на этапе компиляции с незначительными расходами времени, устранение ошибок на этапе тестирования занимает 90% времени. Стоимость устранения одной ошибки в готовом продукте, по данным Microsoft, оценивается в 4 тыс. долл. (по данным IBM, устранение ошибок в ее продуктах, выпущенных в эксплуатацию, обходится от 2-х до 20 тысяч долларов на ошибку) [2].

Корень зла кроется в неправильных акцентах при управлении качеством ПО! Если на тестирование поступит некачественный продукт, он таким и будет выпущен в продажу. Таким образом, организация производства качественного программного обеспечения является ключевой проблемой управления процессом разработки ПО.

Этой проблемой в индустриально развитых странах стали активно заниматься примерно с середины 70-х годов. К числу бесспорных достижений теории менеджмента качества относятся современное понятие качества и его смысловое

наполнение, а также известные модели систем качества серии ISO 9000:2000 (ISO 9001, ISO 9002 и ISO 9003), построенные на базе процессного подхода.

Согласно современному определению, взятому из международного стандарта, *качество программного обеспечения, как и любого другого продукта, – это его соответствие потребностям и ожиданиям потребителя (пользователя)*. Самый верный путь повысить качество ПО – постоянно улучшать и совершенствовать процесс создания и сопровождения продукта в масштабах компании, включающий процессы взаимодействия с внешней средой!

Анализ нормативных документов показывает, что к числу наиболее значимых показателей качества, основанных на обоснованных претензиях пользователей, можно отнести:

- неадекватность функционирования программного продукта;
- недостаточное взаимодействие продукта с другими программными, аппаратными и телекоммуникационными средствами;
- отказы программного продукта в процессе применения по назначению;
- замедленное время работы программного продукта и задержки представления им промежуточной и выходной информации;
- неполнота отражения информации;
- несоответствие хранимых данных и информации, вводимой оператором;
- потеря актуальности информации, циркулирующей в информационной системе;
- нарушения конфиденциальности информации;
- несоответствие содержанию сопроводительной документации и справочной системе программного продукта.

С точки зрения разработчика в роли показателей могут использоваться технические, технологические и экономические показатели:

- число строк кода в стандартном модуле и количество выявленных ошибок на 1000 строк кода;
- параметры сложности программы;
- вероятность появления стандартных и специфических ошибок;
- стоимость единицы кода и цена «человека-месяца»;
- статистические характеристики процессов (математические ожидания, дисперсии, корреляционные функции и т. д.) и другие оценочные параметры.

Отсюда следует «генеральная» задача разработчика ПО – на основе анализа взаимной корреляции ранжированных потребительских, технологических и технических характеристик создать систему целевых показателей (метрик), которая задает ориентиры разработки и критерии оценки ее качества.

Такая задача трудна даже для искушенного разработчика, однако использование ранее наработанного опыта и знаний, накопленных в межпроектных базах данных и отраженных в корпоративных и международных стандартах, разумная автоматизация процессов разработки и тестирования позволяют существенно уменьшить «размерность» задачи.

Выше было отмечено, что из-за отсутствия адекватных систем управления качеством во многих проектах (особенно крупных) временные и экономические показатели значительно превышаются по отношению к запланированным. При этом, однако, не гарантируется выполнение и технических требований. Ряд крупных IT-проектов, как в Европе, так и в Америке, не достигли заявленных результатов, будучи не в состоянии реализовать требуемые технические и технологические параметры, хотя отпущенные время и средства были значительно превышены.

Система управления качеством является частью системы управления организацией, которая ориентирована на достижение результатов, основанных на целях качества, удовлетворении нужд и ожиданий заказчиков. Цели в сфере реализации качества дополняют основные (стратегические) цели организации. Отсюда следует, что различные части системы управления процессами организации-разработчика могут быть объединены вместе с системой менеджмента качества в единую, унифицированную систему управления с общими элементами. Это способствует эффективному планированию, распределению ресурсов, установлению взаимодополняющих целей и реальной оценке эффективности предприятия.

Инициативы внедрения систем качества в широких масштабах в Японии в начале 50-х годов, поддержанные правительственными программами, обеспечили быстрый рост её конкурентоспособности и выход страны на лидирующие позиции в мире в ряде областей промышленности. Активное вне-

дрение подходов к обеспечению качества в США и Европе началось в начале 60-х годов.

Если говорить о программировании, то идеи реализации качества на базе создания стандартного процесса разработки и сопровождения ПО пришли в эту область из промышленности в ответ на программный кризис конца 80-х годов [3]. Среди стандартов в области разработки систем качества, оценки качества процессов и уровней зрелости компаний, разрабатывающих программное обеспечение, а также совершенствования процессов в настоящее время наиболее популярными являются: ISO 9000 (версии 2000 г.), ISO 12207, TickIT, SW-CMM (SEI), Trilium, ISO 15504 (SPICE), CMMI.

3.1.2. Некоторые аспекты стандартизации процесса разработки программного обеспечения

Первая проблема, с которой приходится сталкиваться при попытке стандартизировать процесс разработки ПО, это интеллектуальная природа процесса. Несмотря на то что программирование более или менее формализуется применением стандартных языков программирования и структурированием данных, сам процесс до сих пор является более искусством, чем ремеслом. Одну и ту же программу разные разработчики напишут по-разному, но очень немногие из полученных вариантов будут совершенными. Именно поэтому капитальный труд известного гуру в области программирования Дональда Кнута назван «Искусство программирования».

Компьютерная программа является материальным объектом, но она строится на абстрактных идеях и состоит из сложнейших виртуальных конструкций. Это в корне отличает ее от физических объектов, с которыми имеет дело обычное производство. Выше уже говорилось об этом, но ввиду важности обстоятельства повторим ещё раз – невозможно создать качественное ПО, не решив ряд типичных проблем [4]:

- команда разработчиков, как правило, состоит из творческих личностей и часто бывает очень трудно привести их к «общему знаменателю»;
- каждый программный продукт неизбежно содержит ошибки, отражающие квалификацию и индивидуальный стиль разработчика – разнообразие

зие и нестандартность ошибок сильно усложняют процесс достижения стандартных целей в области качества;

- каждый успешный проект по-своему уникален и индивидуален, он подобен мозаике со сложным рисунком, и поэтому очень трудно выделить из него некий базовый процесс-клише, который можно было применить в дальнейших разработках;
- по сходной причине трудно поставить производство сложного и уникального ПО на поток, так как часто для его разработки требуется создание сопутствующего специфического программного «инструментария» для разработки, оптимизации и тестирования;
- одна из специфических проблем программирования состоит в том, что его продуктивность растет очень медленно, если растет вообще: по некоторым оценкам, средний программист способен создать 10-50 строк операторов в день. Кроме того, эти оценки должны быть уменьшены для больших систем, так как увеличенная нагрузка требует увеличения затрат (в относительных единицах). Это обстоятельство резко отделяет программирование от другого вида деятельности, где поточное производство радикально уменьшает цену единицы продукта. Вся экономика строится на этом далеко не новом принципе – *но только не производство программного обеспечения!*

Таким образом, в большинстве случаев справедливо заключить, что разработка ПО, особенно нестандартного, является в большой степени искусством, нежели ремеслом. Вследствие этого попытки привести процесс программирования в жесткие рамки системы качества в этой области являются более трудными, чем в других видах производственной деятельности.

На рисунке 3.1 показано содержание процесса, превращающего компьютерную программу в программный продукт.

Совершенно очевидно, что эти *ключевые области* жизненного цикла программных средств обязательно должны быть «погружены» в среду организационных и вспомогательных процессов создания инфраструктуры, управления конфигурацией, распределения ответственности, производственного и административного контроля, внутреннего аудита, обучения персонала, регулирования взаимоотношений «поставщик-покупатель» и т.д.

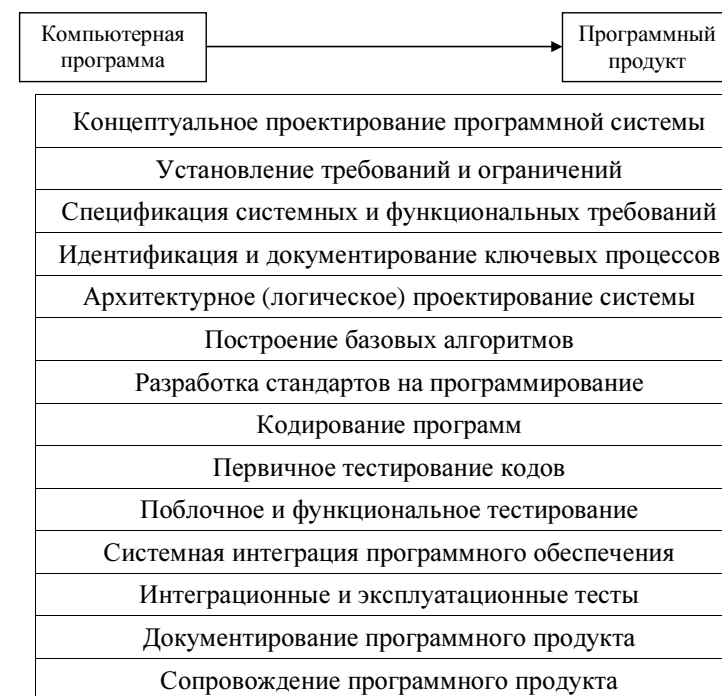


Рис. 3.1. Структура базового процесса для реализации жизненного цикла программного продукта

На рисунке 3.2 представлена концептуальная модель различных аспектов создания программного средства, отображающая внешнее и внутреннее информационное пространство, в котором действует разработчик ПО, а также наборы сущностей, атрибутов и состояний, образующих внутреннюю среду разработки.

Из рисунка видно, что процессы стандартизации и реализации качества не могут быть изолированы от общей системы знаний и опыта, образующей *культурный контекст* разработки ПО. Это достаточно широкая сфера, охватывающая практику решения задач в конкретной прикладной области, состояние программной среды и наличие необходимых инструментов, нормы языков программирования, применение новейших технологий и т.д.

Конкретные условия разработки, организация и структура коллектива разработчиков, принципы его работы – это еще один компонент контекста. Принимаемые решения, эффективность деятельности, конечный результат часто существенным образом зависят от того, организованы программисты в виде «хирургической

бригады» Брукса или как группа равноправных соисполнителей. В любом случае указанные аспекты составляют предметную область процесса реализации и сопровождения качества разработки программного обеспечения. Развитие этих идей, постепенное их воплощение, накопление опыта компаний в создании эффективных методик и технических средств разработки и контроля качества процессов привели к тому, что в течение 30 лет постепенно сложилась определенная культура, появились традиции и стиль работы по созданию программных продуктов.

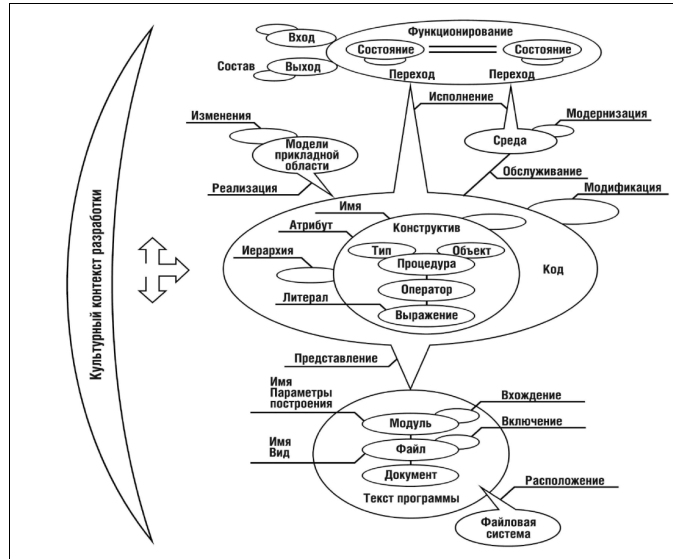


Рис. 3.2. Внутренняя понятийная среда процесса разработки ПО

Наилучшие практики, взятые на «вооружение» ведущими компаниями, образовали так называемые «Тридцать четыре необходимые компетенции для управления программными проектами, стандартизации процесса разработки ПО и реализации его качества» [5]. Эти компетенции нашли отражение в большом количестве национальных и корпоративных стандартов разных стран, которые послужили основой для создания взаимосвязанной сети базовых и вспомогательных международных стандартов, широко применяемых в настоящее время (рис. 3.3). Кроме общих международных стандартов на системы менеджмента качества, оценку и аудит процессов, существует ряд отечественных нормативных документов, конкретно посвященных качеству программных продуктов, упоминание которых интересно в ретроспективном плане.

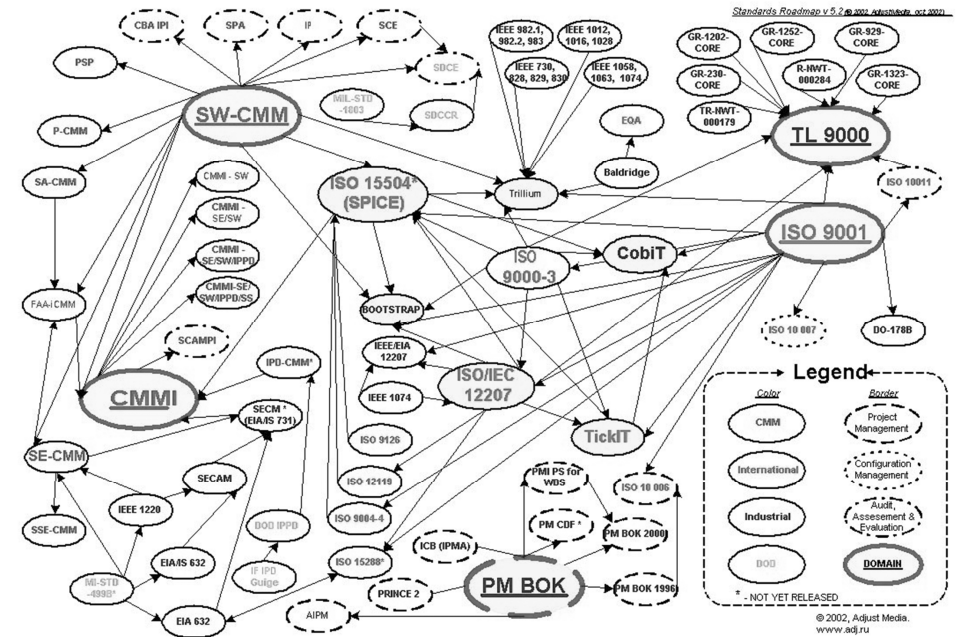


Рис. 3.3. Система взаимосвязанных корпоративных и международных стандартов

В таблице 3.1 приведен список организаций, поддерживающих процесс разработки программного обеспечения на основе международных стандартов.

Таблица 3.1

Организация	Название	Основная цель деятельности	Продукт деятельности в области разработки ПО
Международная организация по стандартизации (International Organization for Standardization)	ISO	Разработка, актуализация, утверждение стандартов	Стандарты семейства ISO 9000, задающие модели систем качества и стандарт жизненного цикла разработки ПО ISO 122207
Институт инженерии по электротехнике и электронике (Institute of Electrical and Electronics Engineers)	IEEE	Разработка стандартов по инжинирингу в области электротехники и электроники	Собрание стандартов в области программного инжиниринга
Американский национальный институт стандартов (American National Standards Institute)	ANSI	Разработка американских национальных стандартов	Руководства, определяющие применение стандартов ISO и ISO 12207 к проектам в области программного инжиниринга
Национальный институт стандартов и технологий США (National Institute of Standards and Technology)	NIST	Разработка технологий, измерений и стандартов в американской промышленности	Стандарты и Национальная премия Малкольма Балдриджа за качество производства и продукта
Институт программного инжиниринга (Software Engineering Institute, USA)	SEI	Разработки в области программного инжиниринга	Модель зрелости возможностей компании, разрабатывающей ПО – модели CMM и CMMI
Американское общество обеспечения качества (American Society of Quality)	ASQ	Улучшение качества производимых продуктов	Систематизация знаний в области инжиниринга качества программного ПО

Окончание табл. 3.1

Организация	Название	Основная цель деятельности	Продукт деятельности в области разработки ПО
Институт управления проектами (Project Management Institute)	PMI	Общие принципы управления проектами, в т.ч. проектами разработки ПО	Руководство «Основы знаний в области управления проектами» (PM BoK – Project Management Body of Knowledge)
Британский институт стандартов (British Standards Institute)	BSI	Разработка и актуализация британских национальных стандартов	Система британских национальных стандартов BS

3.1.3 Отечественные стандарты оценки качества программных продуктов

Рассмотрим ретроспективно некоторые российские стандарты оценки качества ПО. Это интересно потому, что это был, пожалуй, единственный случай в индустрии разработки ПО, когда российский ГОСТ был взят за основу при создании аналогичного стандарта ISO.

Прежде всего, это ГОСТ 28195-89 «Оценка качества программных средств. Общие положения», который устанавливает общие положения по оценке качества программных средств, поставляемых через фонды алгоритмов и программ, номенклатуру и применяемость показателей качества [6]. В стандарте отмечается, что оценка качества должна осуществляться на всех этапах жизненного цикла программных средств: при планировании показателей качества, его контроле на отдельных этапах разработки, в процессе производства, при проверке эффективности модификации на этапе сопровождения. Установлено, что оценку качества проводят специалисты организаций:

- разработчика – на этапах разработки;
- фондодержателя – на этапах приемки программного средства в Фонд алгоритмов и программ;

- испытательных и сертификационных центров – на этапах испытаний и внедрения; изготовителя – на этапах тиражирования;
- пользователя – на этапах внедрения, сопровождения и эксплуатации.

К основным задачам, решаемым при оценке качества программных средств, отнесены:

- планирование уровня качества;
- контроль значений показателей качества в процессе разработки и испытаний;
- эксплуатационный контроль заданного уровня качества;
- методическое руководство разработкой нормативно-технических документов по оценке качества.

С момента вступления в силу ГОСТ 28195-89 произошли существенные изменения во многих аспектах общественной жизни, в том числе значительно изменились экономико-правовые отношения в сферах разработки и эксплуатации программных средств. Например, в области коммерческих программных продуктов исчез фондодержатель, а разработчик и изготовитель чаще всего представляют собой одно и то же юридическое лицо. В рыночных условиях именно разработчик заинтересован в обеспечении качества своих продуктов в течение всего их жизненного цикла. Кроме того, изменился порядок сертификации продукции. Типичный порядок оценки качества программных продуктов, сложившийся в современных условиях, приведен в таблице 3.2.

Таблица 3.2

Заинтересованные стороны в оценке качества	Этапы жизненного цикла программного средства					
	Разработка	Испытания	Тиражирование	Внедрение	Сопровождение	Эксплуатация
Разработчик	Да	Да	Да	Да	Да	Да
Испытательные и сертификационные центры	-	Да	-	Да	-	Да
Пользователь	-	-	-	-	-	Да

Стандарт ГОСТ 28195-89 определяет иерархическую структуру, номенклатуру и содержание понятий качества программных средств. На верхнем

уровне выделены шесть характеристик – *надежность, корректность, удобство применения, эффективность, универсальность и сопровождаемость*, которые детализируются на втором уровне 19 комплексными показателями. На третьем уровне дальнейшая детализация содержит более чем 200 оценочных элементов. Состав используемых показателей рекомендуется выбирать в зависимости от назначения, функций и этапов жизненного цикла программного средства.

В следующем по времени отечественном стандарте ГОСТ 28806-90 установлены основные термины и определения понятий в области качества программных средств. В ГОСТ 28806-90 к общим характеристикам качества программного средства отнесены: *функциональность, надежность, удобство использования, эффективность и сопровождаемость*. В справочном приложении стандарта приведены примеры 20 подхарактеристик качества.

Сравнительный анализ стандартов показывает, что в различных нормативных документах номенклатура показателей качества программных средств заметно отличается друг от друга. Стандарты практически не содержат рекомендаций по выбору, применению и упорядочению необходимой совокупности показателей качества. Кроме того, для большинства показателей отсутствуют методики их оценки и сравнения с установленными требованиями (критериями). Практическое использование нормативных документов еще в большей степени осложняется тем, что ни один из них не содержит перечней факторов, влияющих на качество программных средств, и методик их количественной и качественной оценки.

3.1.4. Стандарты семейства ISO

В 1947 году в Лондоне представители 25 стран решили создать международную организацию, основной задачей которой стала бы координация разработок и унификация международных стандартов. Новая организация получила название Международная организация по стандартизации (International Organization for Standardization – ISO). На греческом языке «*iso*» означает «*равный*» – тем самым подчеркивалось, что все страны, входящие в эту организацию, равны и каждая страна может на равных участвовать в формировании новых стандартов. В настоящее время ее членами являются более 160 стран. Глав-

ной причиной рождения стандартов явилось желание инициаторов создания этой организации устранить технические барьеры в торговле и в оценке качества продуктов, которые возникли вследствие того, что в разных странах для одних и тех же технологий и товаров действовали разнородные стандарты. Сегодня стандартами ISO и других полномочных стандартизирующих организаций (табл. 3.1. и рис. 3.3) «перекрыты» многие технологические отрасли – от промышленного производства, безопасности, программирования и телекоммуникаций до банковской и финансовой сферы. По мнению экспертов, в ближайшие годы стандарты по-прежнему будут распространяться ударными темпами, что обусловлено целым рядом факторов:

- бурным прогрессом в области интеграции мировой торговли и промышленности, взаимопроникновением рынков. С технологической точки зрения свободная конкуренция немыслима без четко сформулированных, понятных и общепринятых стандартов. В современных условиях ни один рынок не может обойтись без компонентов, производимых на других рынках, а цифровая обработка данных требуется повсеместно;
- глобальным распространением компьютерных коммуникационных средств. Вычислительная индустрия – яркий пример технологии, существование которой абсолютно немыслимо без общепринятых стандартов. Полная совместимость открытых систем способствует развитию здоровой конкуренции среди производителей;
- потребностью в наличии общих стандартов в процессе становления новейших технологий. На ранних этапах их развития возникает необходимость в стандартизации терминологии и способах представления и хранения количественной информации.

По уставу ISO членом этой организации может стать «самая авторитетная в стране организация, занимающаяся выработкой стандартов». Таким образом, интересы какой-либо страны в ISO может представлять только одна организация. Помимо основных членом, в ISO входят так называемые члены-корреспонденты (как правило, ими становятся относительно крупные «стандартообразующие» организации отдельных стран, однако еще недостаточно мощные, чтобы распространить свое влияние на стандартизацию технологий по всей стране). Члены-корреспонденты не принимают активного участия в разработке международных стандартов, но имеют полный доступ к интересующей их

информации. Наконец, есть еще и члены-подписчики – они представлены организациями развивающихся стран. В настоящее время Россия имеет статус члена-корреспондента ISO.

Выполнение технической работы в ISO возложено на более чем 2000 технических комитетов, подкомитетов и рабочих групп, в состав которых входят представители правительственных, промышленных, научно-исследовательских и юридических кругов (на сегодняшний день – около 500 организаций). Каждая организация – член ISO – имеет право включить своего представителя в любой комитет, если она особо заинтересована в деятельности этого комитета.

Работу комитетов курируют крупнейшие организации – члены ISO – AFNOR, ANSI, BSI, CSBTS, DIN, SIS и др. Принятие стандарта возможно только после достижения консенсуса между квалифицированным большинством членом комитета.

Центральный секретариат ISO, штаб-квартира которого находится в Женеве, публикует исходные версии стандартов, подготовленные комитетами, и рассылает их членам ISO для ознакомления и голосования. Регулярно издаваемый бюллетень «ISO Momento» содержит подробную информацию о структуре и деятельности всех технических комитетов. Круг интересов и сфера деятельности ISO не ограничиваются какой-либо конкретной технологической областью. Организацию интересует все и вся, за исключением электротехники и электроники (эти направления контролирует International Electrotechnical Commission – IEC). Работы в области информационных технологий ведутся ISO и IEC (Рабочий комитет JTC 1) совместно, поэтому соответствующие стандарты имеют маркировку ISO/IEC.

Новые стандарты утверждаются в соответствии с тремя принципами.

Во-первых, стандарты являются результатом *консенсуса всех заинтересованных сторон* – производителей, поставщиков, потребителей, профессиональных разработчиков, правительственных и исследовательских организаций.

Во-вторых, стандарты имеют действительно *мировое распространение* и удовлетворяют *как производителей, так и потребителей*.

В-третьих, появление новых стандартов диктуется исключительно *научными достижениями, появлением новых технологий, а также требованиями свободного рынка*, а не чьей-то злой или доброй волей. Если рынок созрел для нового стандарта, то такой стандарт появляется.

Процесс создания нового стандарта включает три этапа. Обычно инициатива его разработки исходит от производителей, которые доводят базовые предложения стандарта до своего представителя в ISO. Если эта организация признает целесообразность появления нового стандарта, то соответствующая рабочая группа определяет техническую область, на которую предполагаемый стандарт будет распространяться. На втором этапе идет выработка технических спецификаций, в ходе которой представители различных стран стремятся к достижению консенсуса. На заключительном этапе первая версия стандарта утверждается (за стандарт должно проголосовать 75% кворума) и публикуется. *С этого момента стандарт становится официальным International Standard of ISO!*

По мере совершенствования технологий, появления новых материалов, методов обработки, повышения требований к качеству и надежности изделий возникает необходимость в пересмотре стандартов. В ISO существует правило: все стандарты должны пересматриваться не реже одного раза в пять лет. Сегодня «перу» ISO принадлежит около 9300 различных стандартов, описание которых занимает 171000 страниц текста на английском языке.

Специальная группа планирования работ подкомитета № 7 ISO в 1995 году разработала перспективную Программу развития стандартизации (ПРС) в области программной инженерии (Software Engineering Programme), где даются обзор общих проблем и требований к программной инженерии, а также пути их решения в пределах системы международных стандартов. Определено, что стандарты должны:

- ориентироваться на процессы, продукты, ресурсы и терминологию в области разработки программных средств (ПС);
- способствовать распространению продуктов с предсказуемым качеством;
- способствовать повышению производительности и качества труда программистов;
- содействовать воспроизводимости процессов для ПС;
- поддерживать оценку соответствия ПС заданным требованиям объективными методами;
- инициировать или мотивировать создание необходимой инструментальной поддержки;
- производить экономический эффект при применении;

- защищать общественную безопасность, здоровье и благосостояние людей;
- основываться на реалиях, а не на непроверенных теориях.

Перечень стандартов ISO для построения системы качества и управления качеством

1. ISO 9000-1 (1994 г.). Управление качеством и гарантии качества. Часть 1. Руководство по выбору и использованию.
2. ISO 9000-2 (1993 г.). Управление качеством и гарантии качества. Часть 2. Общее руководство по применению стандартов ISO 9001, ISO 9002 и ISO 9003.
3. ISO 9000-3 (1991 г.). Управление качеством и гарантии качества. Часть 3. Руководство по применению стандарта ISO 9001 при разработке, установке и сопровождении ПО.
4. ISO 9000-4 (1993 г.). Управление качеством и гарантии качества. Часть 4. Руководство по управлению надежностью программ. ISO 9001 (1994 г.).
5. Системная модель качества для процессов проектирования, разработки, производства, установки и обслуживания. ISO 9002 (1994 г.).
6. Системная модель качества для процессов проверки качества проектирования, установки и обслуживания. ISO 9003 (1993 г.).
7. Системная модель качества для процессов проверки качества при окончательном тестировании.
8. ISO 10011-1 (1990 г.). Руководство по аудиту качества систем. Часть 1. Аудит.
9. ISO 10011-2 (1991 г.). Руководство по аудиту качества систем. Часть 2. Квалификационные требования, предъявляемые к аудиторам качества.
10. ISO 10011-3 (1991 г.). Руководство по аудиту качества систем. Часть 3. Менеджмент программ аудита.
11. ISO 10012-1 (1992 г.). Требования по качеству, предъявляемые к измерительной аппаратуре. Часть 1.
12. ISO 10013. Руководство по созданию качественной документации.
13. ISO/TR 13425. Руководство по выбору статистических методов, используемых при разработке стандартов и спецификаций.
14. ISO 8402 (1994 г.). Управление качеством и гарантии качества. Словарь терминов.
15. ISO 9000:2000. Системы менеджмента качества, в 4-х частях.

Отметим, что стандарты семейства ISO 9000, относящиеся к 90-м годам, до сих пор не потеряли актуальности. Часть из них заменена более поздними версиями, отражающими современные реалии (рис. 3.9) Для прохождения процедуры сертификации в ISO организации должны представить комплект документации, оформленной стандартным способом (Quality System Documentation). Задача эта достаточно сложная и отнимает много времени – чтобы упростить ее, можно воспользоваться программой ISOplus, выпускаемой Software Productivity Center. Это комплект более чем из 100 файлов, представленных в формате Word и выполняющих роль шаблонов документов, процедур, анкет и руководств, которые пригодятся в ходе подготовки к сертификации.

3.1.5. Стандарт ISO/IEC 9126:1993

В совместном стандарте ISO и Международной комиссии по электротехнике (IEC) ISO/IEC 9126:1991 «Оценивание программного продукта. Характеристики качества и руководящие указания по их применению» определены шесть групповых характеристик верхнего уровня: *функциональность* (Functionality), *надежность* (Reliability), *удобство использования* (Usability), *эффективность* (Efficiency), *сопровожаемость* (Maintainability), *переносимость* (Portability) – и дан предварительный перечень групповых характеристик второго уровня иерархии (подхарактеристик) [7, 8]. Стандарт дал реальную возможность для развития работ по установлению и стандартизации полной номенклатуры показателей качества вплоть до единичных измеряемых показателей (метрик).

В более поздней версии 1993 года ISO/IEC 9126:1993 выделены несколько видоизмененные характеристики (показатели) качества с позиций пользователя, разработчика и управляющего проектом. Документом рекомендуется шесть основных характеристик верхнего уровня: *функциональная пригодность*, *надежность*, *применимость*, *эффективность*, *сопровожаемость* и *переносимость*, детализированные 21 показателем второго уровня. В более поздней версии эти показатели несколько видоизменены (рис. 3.4).



Рис. 3.4. Соотношение показателей качества программного продукта первого и второго уровней

Ниже приводится смысловое содержание некоторых важных показателей первого и второго уровней [9]:

- *Функциональная возможность* – способность программного средства обеспечивать решение задач, удовлетворяющих сформулированные потребности заказчиков и пользователей при применении комплекса программ в заданных условиях.
- *Функциональная пригодность* – набор и описания субхарактеристики и ее атрибутов, определяющие назначение, номенклатуру, основные, необходимые и достаточные функции программного средства, соответствующие техническому заданию и спецификациям требований заказчика или потенциального пользователя.
- *Правильность (корректность)* – способность программного средства обеспечивать правильные или приемлемые для пользователя результаты и внешние эффекты.
- *Способность к взаимодействию* – свойство программных средств и их компонентов взаимодействовать с одним или большим числом компонентов внутренней и внешней среды.

- *Защищенность* – способность компонентов программного средства защищать программы и информацию от любых негативных воздействий.
- *Надежность* – обеспечение комплексом программ достаточно низкой вероятности отказа в процессе функционирования программного средства в реальном времени.
- *Эффективность* – свойства программного средства, обеспечивающие требуемую производительность решения функциональных задач, с учетом количества используемых вычислительных ресурсов в установленных условиях.
- *Практичность (применимость)* – свойства программного средства, обуславливающие сложность его понимания, изучения и использования, а также привлекательность для квалифицированных пользователей при применении в указанных условиях.
- *Сопровождаемость* – приспособленность программного средства к модификации и изменению конфигурации и функций.
- *Мобильность (переносимость)* – подготовленность программного средства к переносу из одной аппаратно-операционной среды в другую.

В настоящее время завершена разработка и формализован последний проект состоящего из четырех частей стандарта ISO 9126-(1-4) для замены редакций 1991 и 1993 годов. Проект состоит из следующих частей под общим заголовком «Информационная технология – характеристики и метрики качества программного обеспечения»:

- Часть 1. Характеристики и субхарактеристики качества.
- Часть 2. Внешние метрики качества.

Часть 3. Внутренние метрики качества.

- Часть 4. Метрики качества в использовании.

Первая часть стандарта ISO 9126-1 – распределяет атрибуты качества программных средств по шести характеристикам, используемым в остальных частях стандарта. Исходя из принципиальных возможностей их измерения, все характеристики могут быть объединены в три группы, к которым применимы разные категории метрик:

- *категорийные (описательные, номинальные) метрики* – предназначены для «измерения» функциональных возможностей программных средств;
- *количественные метрики* – применимы для измерения надежности и эффективности сложных комплексов программ;

- *качественные метрики* – в наибольшей степени соответствуют практичности, сопровождаемости и мобильности программных средств.

В этой части стандарта ISO 9126-1 даются также определения с уточнениями из остальных его частей для каждой характеристики программного средства, а также для субхарактеристик качества.

Вторая и третья части стандарта ISO 9126-2 и ISO 9126-3 посвящены формализации соответственно внешних и внутренних метрик характеристик качества сложных программных средств. Все таблицы содержат унифицированную рубрикацию, где отражены имя и назначение метрики; метод ее применения; способ измерения, тип шкалы метрики; тип измеряемой величины; исходные данные для измерения и сравнения; а также этапы жизненного цикла программного средства (по ISO 12207), к которым применима метрика.

Четвертая часть стандарта ISO 9126-4 предназначена для покупателей, поставщиков, разработчиков, сопровождающих, пользователей и менеджеров качества программных средств. В ней обосновываются и комментируются выделенные показатели сферы использования (контекста) программных средств и группы выбранных метрик для пользователей.

3.1.6. Методика выбора показателей качества

В начале этого параграфа уместно привести две цитаты, которые исчерпывающе «закрывают» вопрос о необходимости формирования систем количественных и качественных показателей для адекватного измерения процессов разработки и качества продукта: «Когда вы можете измерить то, о чем вы говорите, и выразить это в числах – вы знаете кое-что об этом; но если вы не можете измерить это и выразить в числах – ваше знание скудно и неудовлетворительно» (Лорд Кельвин) и «Мы не можем управлять тем, что мы не можем измерить» (Том де Марко).

Действительно, реально и эффективно управлять процессом разработки можно лишь в том случае, если в каждый момент времени мы можем судить о том, насколько правильно он выполняется. А это возможно только в том случае, если мы имеем систему показателей (метрик) для измерения параметров процесса и можем соотносить их величину и колебания с запланированными или модельными величинами [10].

Проблема формирования адекватных систем показателей действительно является сложной проблемой. Конечно, имеется достаточно много простых и очевидных внешних метрик, единицами измерения которых являются время, деньги, процентные соотношения. Однако, как только речь заходит об измерении внутренних показателей большого проекта или комплексного процесса разработки программного продукта, мы вступаем в область «ноу-хау», так как и проекты, и процессы, с которыми имеют дело в различных компаниях, часто являются уникальными, и компании неохотно делятся своими секретами. Именно поэтому достаточно известно о стандартных метриках (ISO 9126) и так мало публикаций о корпоративных достижениях в области метрологии процессов разработки программного обеспечения.

Исходными данными и высшим приоритетом при выборе показателей качества в большинстве случаев являются *назначение, функциональная пригодность и безопасность* соответствующего программного средства. Достаточно полное и корректное описание этих свойств должно служить базой для определения значений большинства остальных характеристик и атрибутов качества. Принципиальные и технические возможности и точность измерения значений характеристик качества всегда ограничены в соответствии с их содержанием. Это определяет рациональные диапазоны значений каждого атрибута, которые могут быть выбраны на основе здравого смысла, а также путем анализа прецедентов в спецификациях системных и функциональных требований реальных проектов.

Критерии качества программного продукта отражают следующие его характеристики: *обоснованность, надёжность, модульность, тестируемость, переносимость, гибкость, модифицируемость, документированность, ясность, точность, эффективность, экономичность, легкость сопровождения* и т.д. Такой критерий должен:

- численно характеризовать набор основных целевых функций программного средства;
- обеспечивать возможность определения общих и конкретных затрат, необходимых для достижения требуемого качества;
- оценивать степень влияния на показатель качества различных внешних факторов;
- быть по возможности простым, хорошо измеримым и иметь малую дисперсию на широком диапазоне измерений.

Указанные выше составляющие качества не появляются сами собой – они складываются из многих факторов. На рисунке 3.5 показаны слагаемые качества программного продукта – и каждое из этих слагаемых должно быть тщательно измерено.

Определим понятие «метрика». Под метрикой мы понимаем *текстовые, символьные, формульные, табличные формализованные оценки, факторы и критерии, а также правила их применения, образующие базовую систему для измерений различных параметров как самих компьютерных программ, процессов, сервисов, инструментов и средств разработки, ресурсов, так и их качества.*

Эти измерения могут проводиться на уровне общих критериев качества или на уровне отдельных характеристик стандартного процесса, текущего проекта, разрабатываемого продукта.

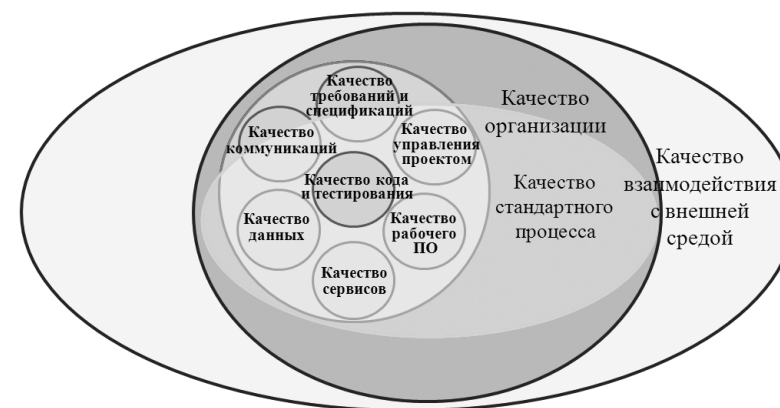


Рис. 3.5. Слагаемые качества программного продукта

В настоящее время в мировой практике используется несколько сотен метрик компьютерных программ [11]. Существующие качественные и количественные оценки можно сгруппировать по шести основным направлениям:

- 1) оценки топологической и информационной сложности программ;
- 2) оценки уровня языковых средств и их применения;
- 3) оценки трудности восприятия и понимания программных текстов, ориентированные на психологические факторы, существенные для сопровождения и модификации программ;

- 4) оценки надежности программных систем, позволяющие прогнозировать нештатные и «отказовые» ситуации;
- 5) оценки производительности ПО и повышения его эффективности путем выявления и коррекции ошибок концептуального и системного проектирования;
- 6) оценки производительности труда программистов для прогнозирования сроков разработки программ и планирования работ по созданию программных комплексов.

Структура метрики может быть достаточно сложной – в общем случае элементами метрики являются следующие базовые составляющие:

- используемая модель технологии производственного процесса (Production Technology Model – PTM);
- используемая модель внешней среды, в которой протекает деятельность рассматриваемой процессной или проектной единицы (Environment Model – EM);
- цели, преследуемые в ходе производственной деятельности (Activity Goals – AG);
- используемая модель функционирования процессно-проектной единицы в рассматриваемой среде – модель реализации имеющихся технологических возможностей производства (Performance Model – PM);
- выбранный метод измерения эффективности (Measurement Model – MM);
- выбранная шкала измерения эффективности (Units of Measure – UM).

Процессы выбора и установления метрик и шкал для описания характеристик качества программных средств можно разделить на два общих этапа:

- выбор и обоснование набора исходных данных, отражающих общие особенности и этапы жизненного цикла проекта программного средства и его потребителей, каждый из которых влияет на определенные характеристики качества комплекса программ;
- выбор, установление и утверждение конкретных метрик и шкал измерения характеристик и атрибутов качества проекта для их последующей оценки и сопоставления с требованиями спецификаций в процессе квалификационных испытаний или сертификации на определенных этапах жизненного цикла программного средства.

На первом этапе за основу следует брать всю базовую номенклатуру характеристик, субхарактеристик и атрибутов, стандартизированных в ISO 9126. Их описания желательно предварительно упорядочить по приоритетам с учетом назначения и сферы применения конкретного проекта программного средства. Далее необходимо выделить и ранжировать по приоритетам потребителей, которым необходимы определенные показатели качества проекта программного средства с учетом их специализации и профессиональных интересов. Подготовка исходных данных завершается выделением номенклатуры базовых, приоритетных показателей качества, определяющих функциональную пригодность программного средства для конкретных потребителей и общую систему количественных показателей, которые будет использовать разработчик.

На втором этапе, после фиксации исходных данных, которое должен выполнить потребитель оценок качества, процесс выбора метрик начинается с ранжирования характеристик и субхарактеристик для конкретного проекта, процесса, продукта и их потребителя. Далее этими специалистами для каждого из отобранных показателей должны быть установлены и согласованы метрика и шкала оценок субхарактеристик и их атрибутов для проекта и потребителя результатов анализа. Определяются измеряемые зоны процессов, устанавливаются перекрытия зон, оцениваются примерные величины корреляций измеряемых величин.

На основании этого можно сформировать конкретный алгоритм формирования метрики:

1. Определение множества характеристик, которые, являясь важными для программного обеспечения, допускают несложное измерение и не перекрываются.
2. Выделение кандидатов в метрики, которые измеряют степень удовлетворения указанным характеристикам.
3. Исследование характеристик и связанных метрик для определения корреляции, значимости, степени автоматизируемости измерений.
4. Анализ и нахождение корреляции между метриками, степени перекрытия, зависимости и недостатки.
5. Рафинирование множества метрик в целом – во множество метрик, которые в совокупности адекватно отражают качество программного обеспечения.
6. Корректировка каждой метрики в итоговом множестве в контексте зафиксированных множеств характеристик и метрик.

Для показателей, представляемых качественными признаками, желательно определить и зафиксировать в спецификациях описания условий, при которых следует считать, что данная характеристика реализуется в программном средстве. Выбранные значения характеристик качества и их атрибутов должны быть предварительно проверены разработчиками на их реализуемость с учетом доступных ресурсов конкретного проекта и при необходимости откорректированы.

3.1.7. Оценка качества программного продукта в соответствии с требованиями международного стандарта ISO 14598

Методологии и стандартизации оценки характеристик качества готовых программных средств и их компонентов (программного продукта) на различных этапах жизненного цикла посвящен международный стандарт ISO 14598, состоящий из шести частей. Стандарт рекомендует следующую общую схему оценки характеристик качества программ:

- установка исходных требований для оценки – определение целей испытаний, идентификация типа метрик программного средства, выделение адекватных показателей и требуемых значений атрибутов качества;
- селекция метрик качества, установление рейтингов и уровней приоритета метрик субхарактеристик и атрибутов, выделение критериев для проведения экспертиз и измерений;
- планирование и проектирование процессов оценки характеристик и атрибутов качества в жизненном цикле программного средства;
- выполнение измерений для оценки, сравнение результатов с критериями и требованиями, обобщение и оценка результатов.

Для каждой характеристики качества рекомендуется формировать меры и шкалу измерений с выделением требуемых, допустимых и неудовлетворительных значений. Реализация процессов оценки должна коррелировать с этапами жизненного цикла конкретного проекта программного средства в соответствии с адаптированной версией стандарта ISO 12207.

Стандарт ISO 14598 рассматривает и оценивает следующие характеристики качества ПС:

Оценка функциональной пригодности – наиболее неопределенная и объективно трудно оцениваемая субхарактеристика программного средства. Облас-

ти применения, номенклатура и функции комплексов программ охватывают столь разнообразные сферы деятельности человека, что невозможно выделить и унифицировать небольшое число атрибутов для оценки и сравнения этой субхарактеристики в различных комплексах программ. Тем не менее, стандарт выделяет некоторые обобщенные субхарактеристики.

Оценка корректности программных средств состоит в формальном определении степени соответствия комплекса реализованных программ исходным требованиям контракта, технического задания и спецификаций на программное средство и его компоненты. Путем верификации должно быть определено соответствие исходным требованиям всей совокупности компонентов комплекса программ, вплоть до модулей и текстов программ и описаний данных.

Оценка способности к взаимодействию состоит в определении качества совместной работы компонентов программных средств и баз данных с другими прикладными системами и компонентами на различных вычислительных платформах, а также взаимодействия с пользователями в стиле, удобном для перехода от одной вычислительной системы к другой с подобными функциями.

Оценка защищенности программных средств включает определение полноты использования доступных методов и средств защиты программного средства от потенциальных угроз и достигнутой при этом безопасности функционирования информационной системы. Наиболее широко и детально методологические и системные задачи оценки комплексной защиты информационных систем изложены в трех частях стандарта ISO 15408:1999-(1-3) «Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий».

Оценка надежности – измерение количественных метрик атрибутов субхарактеристик в использовании: завершенности, устойчивости к дефектам, восстанавливаемости и доступности/готовности.

Оценка потребности в ресурсах памяти и производительности компьютера в процессе решения задач значительно изменяется в зависимости от состава и объема исходных данных. Для корректного определения предельной пропускной способности информационной системы с данным программным средством нужно измерить экстремальные и средние значения длительностей исполнения функциональных групп программ и маршруты, на которых они достигаются. Если предварительно в процессе проектирования производитель-

ность компьютера не оценивалась, то, скорее всего, понадобится большая доработка или даже замена компьютера на более быстродействующий.

Оценка практичности программных средств проводится экспертами и включает определение понятности, простоты использования, изучаемости и привлекательности программного средства. В основном это качественная (и субъективная) оценка в баллах, однако некоторые атрибуты можно оценить количественно по трудоемкости и длительности выполнения операций при использовании программного средства, а также по объему документации, необходимой для их изучения.

Оценка сопровождаемости показывает полноту и достоверность документации о состояниях программного средства и его компонентов, все предполагаемые и выполненные изменения, позволяющее установить текущее состояние версий программ в любой момент времени и историю их развития. Она должна определять стратегию, стандарты, процедуры, распределение ресурсов и планы создания, изменения и применения документов на программы и данные.

Оценка мобильности – качественное определение экспертами адаптируемости, простоты установки, совместимости и замещаемости программ, выражаемое в баллах. Количественно эту характеристику программного средства и совокупность ее атрибутов можно (и целесообразно) оценить в экономических показателях: стоимости, трудоемкости и длительности реализации процедур переноса на иные платформы определенной совокупности программ и данных.

3.1.8. Процессный подход и формирование системы качества при разработке программного обеспечения

Выбор характеристик и оценка качества программных средств – лишь одна из задач в области обеспечения качества продукции, выпускаемой компаниями-разработчиками ПО. Комплексное решение задач обеспечения качества программных средств предполагает разработку и внедрение той или иной системы управления качеством.

В мировой практике наибольшее распространение получила система, основанная на международных стандартах серии ISO 9000, включающей десяток с лишним документов, в том числе стандарт, регламентирующий обеспечение качества ПО (ISO 9000-3) [12,13]. Эти стандарты, а также специализированные

стандарты ISO 12207, CMM, SPICE должны служить руководством для ведущих специалистов компаний, разрабатывающих ПО на заказ.

Стандарт ISO/IEC 12207:1995 «Информационные технологии. Процессы жизненного цикла программного обеспечения»

Первая редакция ISO 12207 подготовлена в 1995 году объединенным техническим комитетом ISO/IEC JTC1 «Информационные технологии, подкомитет SC7, проектирование программного обеспечения». По определению, ISO 12207 – базовый стандарт процессов жизненного цикла программного обеспечения (ЖЦ ПО), ориентированный на различные (любые!) виды ПО и типы проектов разработки информационных систем, куда ПО входит как часть. Стандарт определяет стратегию и общий порядок в создании и эксплуатации ПО, он охватывает ЖЦ ПО от концептуализации идей до завершения ЖЦ [14].

Отметим, что процессы, используемые во время ЖЦ разработки ПО, должны быть совместимы с процессами, используемыми во время ЖЦ создания информационной системы. Отсюда понятна целесообразность совместного использования стандартов на ИС и на ПО. В связи с этим в стандарте дано следующее определение: *система – это объединение одного или более процессов, программного обеспечения, аппаратных средств, телекоммуникационного и другого оборудования и персонала для обеспечения возможности удовлетворения определенных потребностей или целей.*

Стандарт ISO 12207 равно ориентирован на организацию действий каждой из двух участвующих сторон: поставщик (разработчик) и покупатель (пользователь). Стандарт может быть в равной степени применен и в том случае, когда обе стороны – из одной организации.

Общая структура стандарта. Процессы ЖЦ. По сравнению с известными стандартами ISO 12207 состоит из гораздо более крупных обобщенных процессов: «приобретение», «поставка», «разработка» и т. п. (рис. 3.6). Каждый процесс разделен на набор действий, каждое действие – на набор задач (рис. 3.7). Очень важное отличие ISO 12207 от стандартов серии ISO 9000: каждый процесс, действие или задача инициируется и выполняется другим процессом по мере необходимости, причем нет заранее определенных последовательностей (естественно, при сохранении логики связей по исходным сведениям задач и т. п.) – всё определяется конкретным проектом разработки.

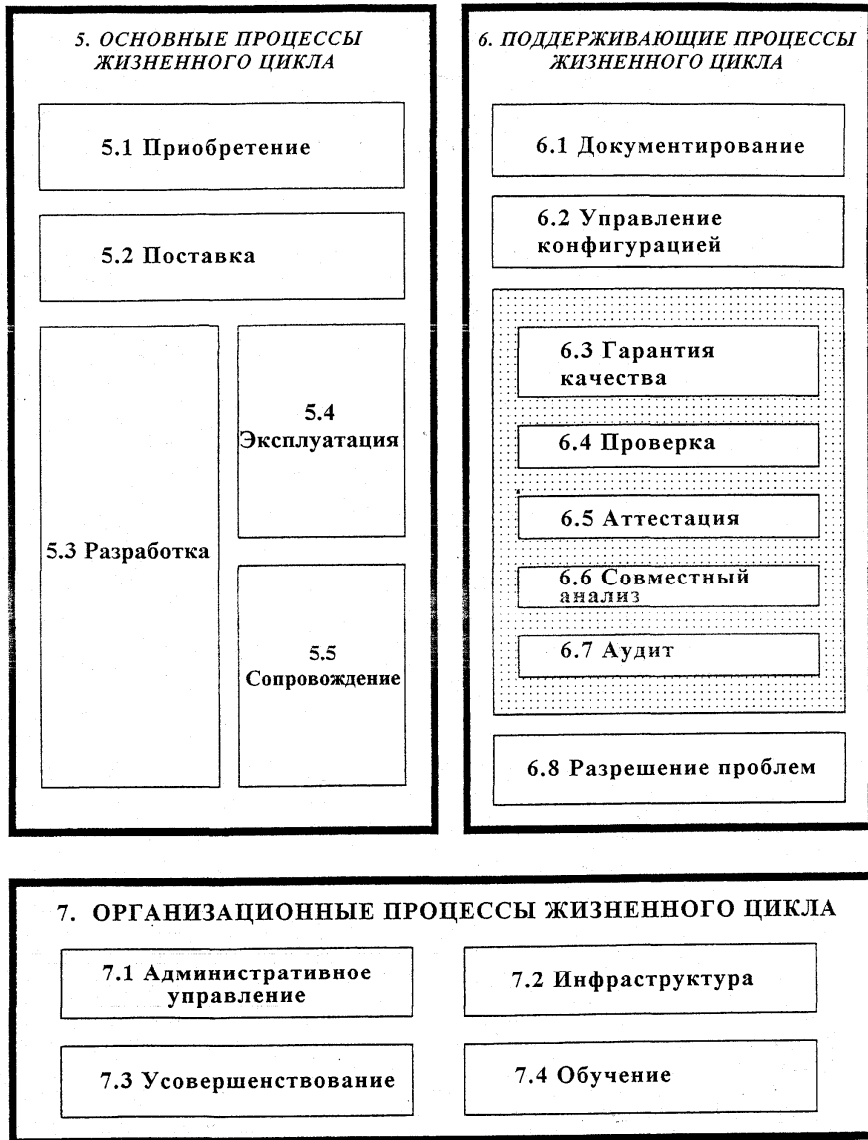


Рис. 3.6. Процессная область стандарта ISO 12207

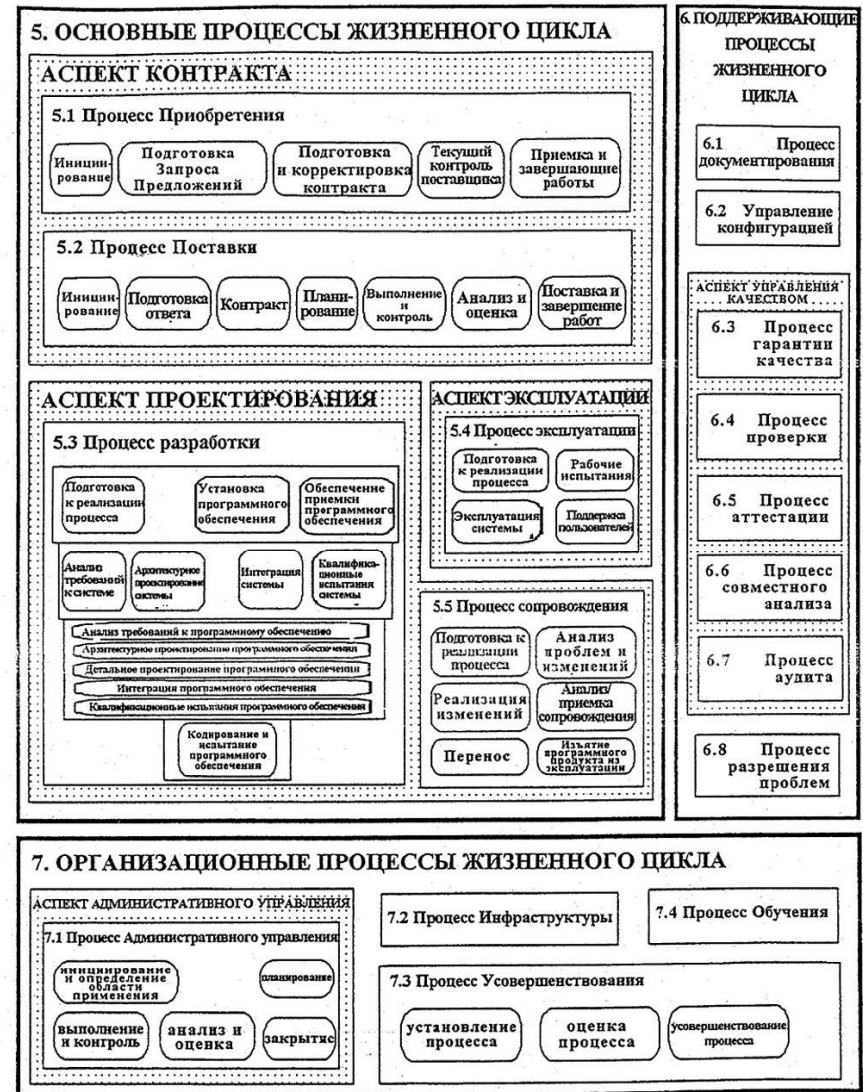


Рис. 3.7. Аспекты и действия по реализации программного продукта

Структура процессов. В стандарте выделено 5 основных процессов жизненного цикла разработки ПО:

1. *Процесс приобретения.* Определяет действия предприятия-покупателя, которое приобретает АС, программный продукт или сервис ПО.

2. *Процесс поставки.* Определяет действия предприятия-поставщика, которое снабжает покупателя системой, программным продуктом или сервисом ПО.

3. *Процесс разработки.* Определяет действия предприятия-разработчика, которое разрабатывает принцип построения программного изделия и программный продукт.

4. *Процесс функционирования (эксплуатации).* Определяет действия предприятия-оператора, которое обеспечивает обслуживание системы (а не только ПО) в процессе ее функционирования в интересах пользователей. В отличие от действий, которые определяются разработчиком в инструкциях по эксплуатации (эта деятельность разработчика предусмотрена во всех трех рассматриваемых стандартах), определяются действия оператора по консультированию пользователей, получению обратной связи и др., которые он планирует сам и берет на себя соответствующие обязанности.

5. *Процесс сопровождения.* Определяет действия персонала сопровождения, который обеспечивает сопровождение программного продукта, что представляет собой управление модификациями программного продукта, поддержку его текущего состояния и функциональной пригодности, включает в себя установку и удаление программного изделия на вычислительной системе.

В стандарте обозначены 8 вспомогательных (поддерживающих) процессов, которые сопровождают реализацию другого процесса, будучи неотъемлемой частью всего ЖЦ программного изделия, и обеспечивают должное качество проекта ПО. Вспомогательные процессы – это процессы решения проблем, документирования, управления конфигурацией, гарантирования качества, последний из которых использует результаты остальных процессов группы обеспечения качества, в которую входят: процесс верификации, процесс аттестации, процесс совместной оценки, процесс аудита.

Число организационных процессов – четыре: это процессы управления, создания инфраструктуры, усовершенствования, обучения. К ним примыкает особый *процесс адаптации*, который определяет основные действия, необходимые для адаптации стандарта к условиям конкретного проекта.

Под процессом усовершенствования здесь понимается не усовершенствование ИС или ПО, а улучшение самих процессов приобретения, разработки, гарантирования качества и т. п., реально осуществляемых в организации.

Особенности стандарта ISO 12207. «Динамический» характер стандарта выявляется способом определения последовательности выполнения процессов

и задач, при котором один процесс при необходимости вызывает другой или его часть. Примеры:

- выполнение Процесса приобретения в части анализа и фиксации требований к системе или ПО может вызывать выполнение соответствующих задач Процесса разработки;
- в Процессе поставки поставщик должен управлять субподрядчиками согласно Процессу приобретения и выполнять верификацию и аттестацию по соответствующим процессам;
- сопровождение может требовать развития системы и ПО, что выполняется по Процессу разработки.
- Такой характер взаимосвязи процессов позволяет реализовывать любую модель ЖЦ.

Таким образом, для всех этапов жизненного цикла программных продуктов разработчик должен самостоятельно разрабатывать комплексы показателей качества, которые в совокупности образуют систему показателей (метрику). Он также самостоятельно должен выявлять факторы, влияющие на качество. Только такой структурированный индивидуальный подход к выбору и обоснованию показателей и факторов обеспечивает эффективный контроль и управление качеством.

Стандарт TickIT

Достаточно широкую известность в среде разработчиков ПО получил британский стандарт TickIT (tick – обложка, чехол) [15]. Как уже было сказано выше, для построения системы качества такой компании абсолютно недостаточно использовать только модель ISO 9001, а необходимо воспользоваться группой специализированных и рекомендательных стандартов. Попытка представить консолидированную модель и дать соответствующие рекомендации, составленные из необходимых стандартов для разработчиков программного обеспечения, реализована в стандарте TickIT.

В отличие от модели ISO 9001, которая регламентирует «*что необходимо сделать?*», разработчики данного стандарта попытались ответить на вопрос «*что необходимо сделать для становления системы качества в организации, занимающейся разработкой программного обеспечения?*».

TickIT – это ещё и схема сертификации систем качества для программного обеспечения, предложенная группой ведущих фирм и некоммерческих организаций Великобритании, работающих в области информатики. TickIT базиру-

ется на стандарте ISO 9001:1994. Таким образом, предметом TickIT является менеджмент предприятий, разрабатывающих программное обеспечение.

Согласно схеме TickIT, могут быть сертифицированы системы качества предприятий, занимающихся следующими видами деятельности:

- разработка программного продукта или услуги как для внешнего заказчика, так для внутреннего использования, включая встроенное (Embedded) программное обеспечение;
- системная интеграция, поддержка, администрирование программных комплексов и систем;
- копирование, архивирование, хранение данных и программного обеспечения.

Помимо своей основы – стандарта ISO 9001, TickIT содержит следующие дополнительные компоненты:

- руководство по TickIT, базирующееся на указаниях ISO 9000-3 (руководство по системам качества для программного обеспечения);
- схему регистрации аудиторов через специальный Комитет по TickIT IRCA (Международный регистр сертифицированных аудиторов);
- систему проверок аудиторов Британским компьютерным обществом (BCS) и Институтом по обеспечению качества (IQA);
- систему аккредитации сертификационных обществ UKAS (Великобритания), SWEDAC (Швеция);
- программы, направленные на расширение признания схемы;
- трехлетний цикл пересмотра схемы;
- систему специальных премий за достижения.

Стандарт TickIT, кроме требований модели ISO 9001, включает набор требований и практик из стандартов ISO 12207 «Жизненный цикл ПО» и ISO 9000-3 «Руководство для внедрения ISO 9001:94 для разработки, установки и поддержки ПО». Преимуществом данной модели является в первую очередь то, что данный стандарт представляет собой *достаточную модель*, по которой можно не просто проверять, а, самое главное, разрабатывать систему качества. Следующее важное преимущество: разрабатывая систему качества по модели TickIT, мы также строим «ISO 9000-совместимую» систему качества компании.

Стандарт «Использование ISO 9001:2000 для построения систем менеджмента качества программных продуктов, сертификации и непрерывного улучшения» (The TickIT Guide) разработан профессионалами отрасли из Европы и

США, приглашёнными для работы над стандартом в составе специального комитета BRD/3/1 Британским институтом стандартизации (BSI).

К недостаткам модели TickIT можно отнести недостаточное распространение данного стандарта и, как следствие, недостаточное его признание в мире. В самом деле – если рассмотреть стандарты, которые были взяты в основу TickIT, то невольно задаешься вопросом, почему разработчики TickIT не включили в свою модель другие не менее важные и нужные для организаций разработчиков программного обеспечения стандарты. Такие как ISO 9004-1, ISO 10007 «Управление конфигурацией» и другие?

Здесь нужно учитывать следующее важное обстоятельство – модель TickIT создавалась как отраслевой стандарт для внедрения модели ISO 9000 в организациях-разработчиках ПО, и создатели данного стандарта дали своё, далеко не бесспорное представление о необходимости ограничиться лишь тремя из всего многообразия справочных и рекомендательных стандартов для построения эффективной системы качества.

Отметим всё же, что для первого представления о системе качества для разработчиков ПО гораздо проще изучить TickIT, а не разбираться во всей сложной структуре иерархии рекомендательных стандартов ISO в области обеспечения качества. Гибкая инфраструктура TickIT позволяет схеме следовать изменениям в этой весьма динамичной отрасли, обеспечивая тем самым ее постоянное совершенствование. В последней редакции руководства по TickIT активно используется идеология жизненного цикла программного обеспечения. Основные определения процессов жизненного цикла программного обеспечения даны в другом стандарте – ISO/IEC 12207, который может служить основой при выборе конкретной модели процессов жизненного цикла на предприятии. Таким образом, схема TickIT использует рациональное начало, заложенное в методах совершенствования процессов таких популярных международных стандартов, как CMM (SEI, USA), Trillium (Bell Canada), BOOTSTRAP и других.

Версия международного стандарта ISO 9000:2000

Стандарты ISO являются наиболее известными и распространенными в мире. Они универсальны, их можно применять в качестве моделей независимо от отрасли, в которой функционирует компания. Такой подход, очевидно, имеет свои преимущества и недостатки.

Основным преимуществом международных стандартов ISO серии 9000 является их легитимность, известность, распространенность, признание на ми-

ровом уровне, большое количество экспертов и аудиторов и невысокая стоимость услуг по сертификации. Универсальность же моделей ISO серии 9000 содержит в себе определенные недостатки: они являются достаточно высокоуровневыми, задают абстрактные модели и не содержат конкретных методологических разработок. Стандарты носят рекомендательный характер, выполнение их требований является сугубо добровольным делом разработчика.

По ISO, «качество – это совокупность свойств и характеристик продукта, процесса или услуги, которые обеспечивают способность удовлетворять заявленным или подразумеваемым потребностям». Современные способы обеспечения качества базируются на принципах Всеобщего руководства качеством (Total Quality Management – TQM). TQM – это управление ресурсами и применение количественных методов анализа для улучшения:

- разработок, материалов и услуг, поставляемых в организацию;
- всех процессов внутри организации;
- степени удовлетворенности настоящих и будущих потребностей клиентов.

Задавая модели системы качества, стандарты семейства ISO 9000 лишь формулируют требования, которые должны быть реализованы, но не говорится о том, как это можно сделать. Это связано с тем, что для того, чтобы рекомендовать абстрактному предприятию способы реализации и зафиксировать необходимые рекомендации по выполнению требований, необходимо конкретизировать:

- сферу деятельности организации;
- специфику ее процессов;
- специфику культуры предприятия;
- структуру управления, существующую в компании (матричная, иерархическая проектная) и другие особенности, свойственные компании.

Поэтому для построения полноценной системы качества по ISO необходимо, помимо основной модели ISO 9001:2000, использовать вспомогательные отраслевые и рекомендательные стандарты. Для организации, занимающейся разработкой программного обеспечения, такими стандартами являются: ISO 9000-3:91, ISO 10007:95, ISO 10013:95, ISO 12207:95.

Семейство стандартов ISO 9000 версии 2000 года разработано с тем, чтобы преодолеть недостатки ISO 9000 версии 1994 года и помочь организациям всех специализаций, типов и размеров внедрить и использовать эффективные системы менеджмента качества на базе внедрения и повседневного использования процессного подхода. Оно включает следующие документы:

- ISO 9000:2000, являющийся введением в системы менеджмента качества и содержащий соответствующий словарь;
- ISO 9001:2000, устанавливающий детальные требования для систем менеджмента качества;
- ISO 9004:2000, обеспечивающий руководство по развитию системы менеджмента качества;
- ISO 10011:2000, предназначенный для управления и проведения внутреннего и внешнего аудитов системы менеджмента качества.

Семейство этих стандартов различает требования к системам менеджмента качества и требования к программным продуктам, как, впрочем, и к любой другой продукции. Требования к системам менеджмента качества подробно определены в стандарте ISO 9001:2000.

Подход к системам менеджмента качества является общим и применяется к организациям в любой отрасли экономики, поэтому данный стандарт не устанавливает каких-либо конкретных требований к программным продуктам. Требования к ним могут определяться заказчиками или третьими лицами и содержаться в технических спецификациях, стандартах на продукт, стандартах на процесс, контрактных соглашениях и нормативных документах.

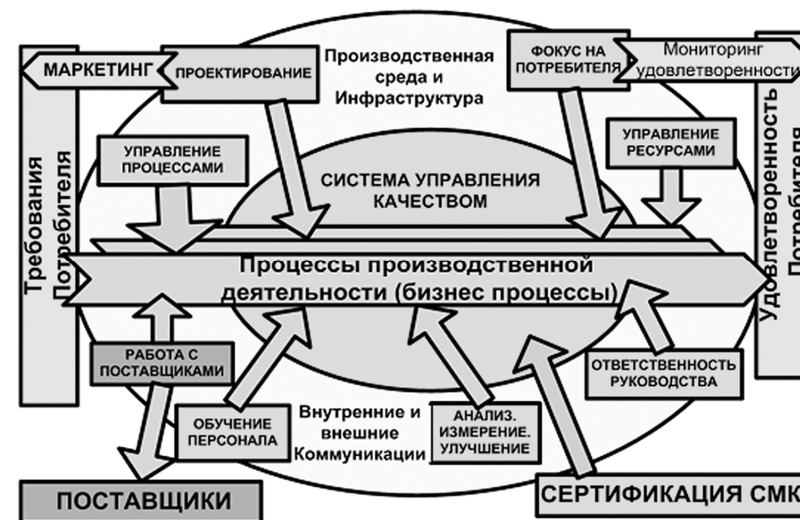


Рис. 3.8. Реализация процессного подхода на базе ISO 9000:2000

В основу построения системы качества в соответствии с моделью ISO 9000:2000 закладываются следующие восемь основных принципов:

- концентрация на потребностях заказчика;
- активная лидирующая роль руководства;
- вовлечение исполнителей в процессы совершенствования;
- реализация процессного подхода;
- системный подход к управлению;
- обеспечение непрерывных улучшений;
- принятие решений на основе фактов;
- взаимовыгодные отношения с поставщиками.

При этом методически в полном соответствии с дисциплиной построения сложных систем в стандарте ISO 9000:2000 предусматривается, с одной стороны, построение организационной системы «сверху – вниз»: от целей предприятия и его политики – к организационной структуре и формированию бизнес процессов, и с другой – итеративное развитие организационной системы через механизмы измерения и улучшения (рис. 3.8).

Внедрение системы менеджмента качества организацией-разработчиком программных продуктов по ISO 9000 версии 2000 года состоит из нескольких этапов.

В их числе можно выделить:

- измерение характеристик продуктов для определения эффективности каждого процесса, направленного на достижение целей качества;
- применение результатов измерений для определения текущей эффективности процессов создания и внедрения продуктов;
- определение способов предотвращения дефектов, снижения изменчивости продукции и минимизации доработок;
- поиск возможностей по снижению рисков и улучшению эффективности и производительности технологических и иных процессов;
- выявление и расстановка в порядке важности тех улучшений, которые могут давать оптимальные результаты с приемлемыми рисками;
- планирование стратегии, процессов и ресурсов для получения идентифицированных улучшений продукции;
- контроль результатов улучшений;
- сравнение полученных результатов с ожидаемыми или запланированными;

- определение подходящих корректирующих действий.

Реализация этих этапов возможна только при наличии в организации системы профилей международных и корпоративных стандартов для оценки стабильности, производительности и качества процессов разработки, критериев, показателей и факторов качества, а также методов их измерения и оценки. На рисунке 3.9 показана совокупность базовых стандартов ISO, регламентирующих разработку и оценку качества программного обеспечения.

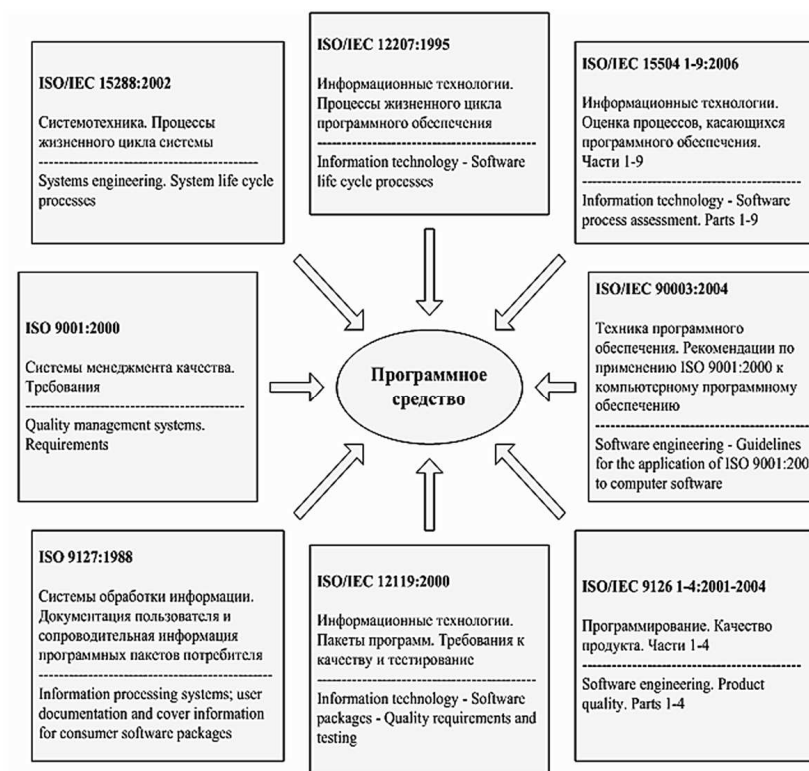


Рис. 3.9. Базовые стандарты ISO, регламентирующие разработку и оценку качества ПО

Система менеджмента качества является частью системы управления, которая ориентирована на достижение результатов, основанных на целях качества, удовлетворении нужд и ожиданий заказчиков. Цели качества дополняют

другие цели организации. Различные части системы управления организации-разработчика могут быть объединены вместе с системой менеджмента качества в единую, унифицированную систему управления с общими элементами. Это способствует планированию, распределению ресурсов, установлению взаимодополняющих целей и оценке эффективности.

3.2. Модели оценки зрелости компаний и технологических процессов.

Совершенствование процесса разработки программного продукта

3.2.1. Концепция Baldrige Award

Объединение международных рынков, повышение требований к качеству и жесткая конкуренция привели к появлению двух параллельных концепций стандартизации качества – ISO 9000:2000, более распространенная в Европе, и Malcolm Baldrige National Quality Award (кратко Baldrige Award), весьма популярная в США. При этом часто ошибочно полагают, что обе концепции содержат примерно одни и те же требования, затрагивают одни и те же критерии качества и, следовательно, почти эквивалентны. Компании в соответствии с национальной принадлежностью или пристрастиями могут выбирать любую [10].

В действительности между двумя упомянутыми концепциями существует принципиальная разница в ориентации, целях и внутреннем содержании (по мнению экспертов, ISO 9000 перекрывает менее 10% критериев, вошедших в Baldrige Award). Главная цель Baldrige Award – научить организации создавать конкурентоспособные продукты. Основная цель ISO 9000 – заставить их жестко следовать описаниям технологических процессов, которые они сами же и составляют. Регистрация в ISO свидетельствует о том, что компания поддерживает «документированную практику, основанную на процессном подходе». Однако качество конечного продукта, требования рынка и технологический уровень самих производственных процессов в компании оказываются вне интересов модели ISO. Отметим, что версия ISO9000:2000 несколько сблизила обе концепции.

Сильная сторона концепции Baldrige Award состоит в том, что она ориентирована на виды деятельности, способствующие повышению конкурентоспособности компаний. Концепция предусматривает различные способы для достижения этого: обращение лицом к рынку и клиентам, нацеленность на конечный результат, постоянное совершенствование деловых процессов, тесная при-