

**Институт системного программирования
Российской академии наук**

B.V. Липаев

**ЧЕЛОВЕЧЕСКИЕ ФАКТОРЫ
В ПРОГРАММНОЙ ИНЖЕНЕРИИ:
РЕКОМЕНДАЦИИ И ТРЕБОВАНИЯ
К ПРОФЕССИОНАЛЬНОЙ КВАЛИФИКАЦИИ
СПЕЦИАЛИСТОВ**

УЧЕБНИК

**СИНТЕГ®
Москва - 2009**

**УДК 004.41(075.8)
ББК 32.973.26-018я73
Л61**

Липаев В.В.

**Человеческие факторы в программной инженерии:
рекомендации и требования к профессиональной квалификации
специалистов. Учебник.** – М.: СИНТЕГ, 2009. - 328 с.

ISBN 978-5-89638-110-5

Для анализа и учета человеческих факторов при производстве программных продуктов в учебнике изложены общие психологические характеристики субъектов, особенности лидеров и коллективов специалистов. В основной группе лекций представлены требования к психологическим и профессиональным характеристикам руководителей и специалистов, их обучению необходимой квалификации, что они должны знать и уметь для успешной деятельности на основных этапах жизненного цикла сложных комплексов программ. Рекомендации и требования для обучения специалистов производству крупных программных продуктов иллюстрированы тремя десятками рисунков, каждый из которых отражает определенные квалификационные уровни и детализирован перечнем необходимых профессиональных знаний и навыков. Значительное внимание сосредоточено на задачах обучения специалистов, которым предстоит разрабатывать требования к программным продуктам, оценивать экономические характеристики производства, планировать деятельность коллективов специалистов при создании таких продуктов. Изложены требования к обеспечению профессиональной квалификации специалистов, управляющих модификаций, сопровождением и конфигурацией крупных программных продуктов, обеспечивающих их качество и безопасность, тестирующих компоненты и комплексы программ, осуществляющих их динамические испытания. Описаны организация и обучение применению стандартов программной инженерии, а также состав и содержание системы международных стандартов.

Учебник целесообразно использовать при обучении студентов старших курсов, аспирантов и менеджеров проектов для повышения квалификации по созданию сложных комплексов программ на всем их жизненном цикле с учетом человеческих факторов. Курс лекций (30 часов) ориентирован также на заказчиков, менеджеров крупных проектов, аналитиков и ведущих специалистов, обеспечивающих все этапы жизненного цикла сложных программных систем, к которым предъявляются высокие требования к качеству и безопасности функционирования и ограничены доступные ресурсы и сроки разработки.

© В.В. Липаев, автор, 2009

ОГЛАВЛЕНИЕ

Предисловие

Лекция 1. Человеческие факторы, влияющие на производство с позиции инженерной профессиональной психологии

Основные понятия инженерной профессиональной психологии. Основы психологии профессиональных субъектов при производстве сложных продуктов. Психологические особенности лидеров производственных коллективов.

Лекция 2. Психологические особенности коллективов специалистов для производства программных продуктов

Психологические особенности групп и небольших команд специалистов для производства программных продуктов. Жизненный цикл развития производственного коллектива программных продуктов.

Лекция 3. Рекомендации по основам организации профессионального обучения специалистов для проектирования и производства крупных программных продуктов

Основные свойства руководителей и специалистов, необходимые при проектировании и производстве крупных программных продуктов. Рекомендации преподавателям и учащимся при обучении проектированию и производству крупных программных продуктов. Модель обучения и обеспечения квалификации руководителей и специалистов для производства крупных программных продуктов.

Лекция 4. Требования к профессиональной квалификации руководителей и специалистов, разрабатывающих требования к крупным программным продуктам

Основы формирования требований к крупным программным продуктам. Особенности подготовки и реализации требований заинтересованных лиц к программному продукту. Проверка корректности требований к крупным

комплексам программ. Требования к профессиональной квалификации руководителей для эффективного управления производством крупных программных продуктов.

Лекция 5. Требования к профессиональной квалификации руководителей и специалистов, оценивающих экономические характеристики производства крупных программных продуктов

Основные требования к профессиональной квалификации руководителей для оценивания экономических характеристик производства программных продуктов. Оценивание требуемых экономических характеристик производства программного продукта с учетом профессиональной квалификации специалистов. Оценивание стоимости привлечения квалифицированных специалистов для производства программных продуктов.

Лекция 6. Требования к профессиональной квалификации руководителей, планирующих деятельность специалистов при производстве крупных программных продуктов

Требования к профессиональной квалификации руководителей по планированию деятельности коллективов специалистов при производстве крупных программных продуктов. Требования к профессиональной квалификации руководителей, разрабатывающих графики производства программных продуктов.

Лекция 7. Требования к профессиональной квалификации специалистов, управляющих модификацией, сопровождением и конфигурацией крупных программных продуктов

Требования к профессиональной квалификации специалистов, управляющих сопровождением и модификацией крупных комплексов программ. Требования к профессиональной квалификации специалистов, управляющих конфигурацией крупных комплексов программ.

Лекция 8. Требования к профессиональной квалификации специалистов, обеспечивающих качество программных продуктов

Требования к качеству программных продуктов, определяющие необходимую профессиональную квалификацию

специалистов при их производстве. Понятия и общие свойства дефектов и ошибок в крупных комплексах программ. Требования к профессиональной квалификации специалистов для учета причин и типов дефектов и ошибок в крупных комплексах программ.

Лекция 9. Требования к профессиональной квалификации специалистов, тестирующих компоненты и крупные комплексы программ

Требования к профессиональной квалификации специалистов для успешного тестирования компонентов и комплексов программ. Требования к профессиональной квалификации специалистов по организации, планированию, оценке и учету затрат на тестирование компонентов и крупных комплексов программ.

Лекция 10. Требования к профессиональной квалификации специалистов, осуществляющих динамические испытания крупных программных продуктов

Требования к профессиональной квалификации специалистов для подготовки динамических испытаний крупных программных продуктов. Требования к профессиональной квалификации руководителей и специалистов по организации динамических испытаний крупных программных продуктов. Требования к профессиональной квалификации специалистов для выполнения динамических испытаний крупных программных продуктов. Требования к профессиональной квалификации специалистов для прогнозирования характеристик ошибок в крупных динамических комплексах программ.

Лекция 11. Требования к профессиональной квалификации руководителей и специалистов по стандартизации проектирования и производства крупных программных продуктов

Требования к профессиональной квалификации руководителей и специалистов по организации, освоению и применению стандартов программной инженерии. Требования к составу и содержанию системы стандартов для производства крупных программных продуктов.

Литература

ПРЕДИСЛОВИЕ

Человеческий фактор в создании сложных комплексов программ может определять качество реализации их функций, надежности и безопасности путем применения *двух методологий и стилей* использования специалистов – разработчиков. Им соответствуют: большое число независимых, различных по квалификации, дислокации и ответственности за результаты добровольной деятельности – **«демократический стиль»** использования специалистов и организация профессиональных, сплоченных коллективов высокой квалификации под централизованным управлением – **«авторитарный стиль»**. Эти методологии различаются сферами применения, методами достижения высокого качества комплексов программ, психологическими характеристиками участвующих специалистов и организацией их деятельности в реальном времени. Различие стилей использования человеческого фактора при создании крупных комплексов программ привело к формированию двух стратегий, позволяющих получать сложные программные продукты, характеризующиеся высоким качеством, надежностью и безопасностью применения. Результатами использования этих методологий обычно называют [7, 15]:

- *свободное программное обеспечение* (СПО), ориентированное на высокие тиражи распространения, применения и модификации благодаря доступности для многих пользователей исходных текстов программ на языке программирования и/или в двоичном коде, а также первичной технологии, процессов разработки и сопровождения комплексов программ;

- *закрытые тексты* и технологические документы комплексов программ, создаваемые для конкретных заказчиков определенных систем управления и обработки информации с ограниченным тиражом, производство, распространение и модификация которых регламентируется и контролируется заказчиком.

Первый класс сложных программных комплексов первоначально обычно создается группами энтузиастов в университетах, корпорациях или сообществах разработчиков и пользователей и распространяется бесплатно. Функции таких комплексов программ ориен-

тированы на решение новых, оригинальных задач массового использования и конкурентоспособных на рынке программных продуктов, куда со временем они поступают. Их разработка, изменение и совершенствование не регламентируется в реальном времени и реализуется по инициативе пользователей без определенных, контролируемых планов и сроков. По мере безвозмездного совершенствования функций и качества программного продукта различными заинтересованными, неуправляемыми пользователями – разработчиками (зачастую сотни и тысячи) расширяется сфера его применения, повышается качество, надежность и безопасность использования, что приводит к активному проникновению в бизнес и систему образования.

Второй класс программных продуктов обычно имеет конкретного заказчика, относительно узкую сферу применения, предназначен для конкретных систем или пользователей, жесткую регламентированную технологию производства, модификаций и документирования, что сближает их создание с обычным промышленным производством сложных изделий. Он управляет планами и ограниченными сроками поставки готовых испытанных продуктов заказчику и пользователям, которые допускается эпизодически изменять только с санкции заказчика. Современные крупные программные продукты для систем управления и обработки информации в реальном времени являются одними из наиболее сложных интеллектуальных объектов, создаваемых человеком, и требованиям к их **необходимой профессиональной квалификации посвящено основное содержание учебника.**

В процессе разработки таких продуктов **творчество руководителей и специалистов** как поиск методов, альтернативных решений и способов осуществления заданных требований, а также формирование и декомпозиция этих требований, составляют значительную часть всех человеческих затрат. Индустриализация разработки программных продуктов позволяет автоматизировать многие нетворческие, технические и рутинные операции и этапы, а также облегчает творческие процессы за счет селекции, обработки и отображения информации, необходимой для принятия творческих решений. Следствием этого является возможность значительного сокращение доли затрат на творческий труд в непосредственных затратах специалистов на проектирование и производство комплексов программ.

Большинство **вузов страны пока не учит регламентированному производству крупных программных продуктов** и до сих пор

ограничивается обучением элементам традиционного программирования только небольших простейших программ. Они *не готовят* остро необходимых для современной программной индустрии профессионалов: системных аналитиков, архитекторов и менеджеров проектов, специалистов по комплексированию, испытаниям, обеспечению качества, надежности и безопасности применения крупных программных продуктов. Выпускники таких вузов, как правило, *не знают* современных промышленных методов, технологий и международных стандартов программной инженерии, поддерживающих и регламентирующих жизненный цикл комплексов программ, инструментальных систем обеспечения качества, верификацию, тестирование и сертификацию сложных программных продуктов. Это определяет низкую системотехническую и профессиональную квалификацию программистских коллективов ряда отечественных предприятий, берущихся за создание крупных программных систем. В результате многие сложные программные продукты оказываются не конкурентоспособными, недостаточного качества и безопасности, требуют длительной доработки, тестирования и испытаний для устранения системных и технических дефектов и ошибок.

Очень быстро увеличивается сложность и ответственность задач систем обработки информации и управления, возлагаемых на ЭВМ, что вызывает *рост требований к качеству*, надежности функционирования и *безопасности применения* программных продуктов. Неуклонно повышаются размеры многих комплексов программ, что вызывает возрастание затрат творческого труда на единицу размера новых программ. Каждое предприятие вынуждено тратить большие средства на поиск, обучение и переподготовку специалистов. Все больше определяющей для экономических показателей создания крупных программных продуктов становится доля затрат на творческий труд, соответственно *возрастают требования к профессиональной квалификации и творческим способностям при отборе и обучении специалистов*.

Специалистов необходимо обучать ряду важных *для программной индустрии профессиям*, связанным с созданием крупных программных продуктов [33, 39, 42]:

- организации и регламентированной работе больших профессиональных коллективов специалистов над целостным продуктом;
- распределению сотрудников разной профессиональной спе-

циализации по производственным этапам, компонентам и видам работ в жизненном цикле комплексов программ;

- планированию и методам работы в условиях ограниченных ресурсов, по графикам в реальном времени с заданными сроками, контролем качества и документированием результатов;
- тестированию, испытаниям и гарантии качества, надежности компонентов и программного продукта в целом.

Эти профессиональные факторы способствуют формированию, совершенствованию и применению *современной методологии и регламентированной инженерной дисциплины* обеспечения процессов производства сложных программных продуктов для различных областей. Обычно такие системы существуют и применяются долгие годы, развиваясь от версии к версии, претерпевая на своем жизненном пути множество изменений, улучшение существующих функций, добавление новых или удаление устаревших возможностей, адаптацию для работы в новой среде, устранение дефектов и ошибок.

Программная инженерия – это область компьютерной науки и технологии, которая занимается проектированием и производством крупных программных систем, настолько больших и сложных, что для этого требуется участие *слаженных профессиональных коллективов* инженеров различных специальностей и квалификаций. *Суть методологии программной инженерии* состоит в применении систематизированного, научного и предсказуемого в реальном времени процесса проектирования и производства крупных программных комплексов. Это обеспечивает организацию, освоение и применение апробированных, высококачественных методов и процессов проектирования, программирования, верификации, тестирования и сопровождения программных комплексов и их компонентов. Тем самым такие процессы позволяют получать стабильные, *предсказуемые результаты* и программные продукты требуемого качества.

Важнейшим человеческим фактором, определяющим производство и качество программных продуктов, являются *люди – специалисты* с их уровнем профессиональной квалификации, а также с многообразием психологических характеров, знаний, опыта, стимулов и потребностей. Быстрый рост сложности и повышение *ответственности за качество* программных продуктов привели к появлению *новых требований* к квалификации и дифференциации обучения специалистов в области программной инженерии. Им недостаточно навыков процедурного программирования небольших компо-

нентов, а необходимы глубокие знания системотехники, технологии и стандартов проектирования, производства программных продуктов в определенной тематической области их применения. Эти специалисты должны владеть ***новыми интеллектуальными профессиями***, обеспечивающими высокое качество программных продуктов, а также контроль, испытания и удостоверение реального достигнутого качества на каждом этапе разработки и совершенствования комплексов программ. Накопленный опыт создания крупных комплексов программ и острый дефицит соответствующих специалистов отразились на необходимости ***принципиального изменения и расширения методов и программ их обучения и воспитания***. Крупномасштабное производство программных продуктов различных классов, разделение труда специалистов по профессиональной квалификации при разработке программ, структура и организация больших коллективов, а также экономика таких производств, стали важнейшей частью ***проблемы выбора, обучения и стимулирования специалистов*** для фрагментов всего жизненного цикла сложных программных продуктов.

Человеческие факторы – личная профессиональная квалификация, психологические характеристики специалистов и организация крупного коллектива в ***основном определяют*** качество и трудоемкость при производстве программных продуктов. Разработка сложных программных комплексов, особенно на начальных и завершающих этапах, характеризуется наиболее высокой долей творческого труда. Дефекты, трудоемкость и длительность отдельных операций и частных работ существенно зависят от индивидуальных психологических особенностей их исполнителей, функционального назначения и характеристик конкретного проекта. Даже при сокращении суммарных затрат на разработку программных компонентов за счет автоматизации нетворческого труда, все более определяющей для производства становится доля затрат на творческий труд и ***возрастают требования к творческим профессиональным способностям конкретных специалистов***. Отсюда принципиальной особенностью планирования производства комплексов программ является необходимость активного участия руководителей – менеджеров при ***отборе и подготовке профессиональных специалистов – создателей программных продуктов***, а также при составлении планов их производства на

базе исследованных характеристик прототипов завершенных разработок и личного опыта.

Анализ и учет **человеческих факторов** в программной инженерии *связаны с большими трудностями*, типичными для новых разделов науки и техники, появляющихся *на стыке различных профессий и областей знаний*. В данном случае особенности состоят в том, что **менеджеры и разработчики** комплексов программ, как правило, не знают даже основ психологии специалистов, необходимой для производства сложной интеллектуальной продукции, а **психологи** не представляют сущность и свойства объектов разработки - программных продуктов, а также особенностей технологических процессов, их производства и применения. Создание программных средств как производственной продукции существенно повысило **актуальность обоснования, прогнозирования и оценивания роли человеческих факторов** для характеристики качества процессов производства. Объективно положение осложнено трудностью оценки и учета психологических характеристик и профессиональной квалификации руководителей и специалистов, участвующих при создании таких объектов. Широкий спектр количественных и качественных показателей, которые с различных сторон характеризуют содержание компонентов и комплексов программ, и не высокая достоверность оценки их свойств и значений, определяют значительную неопределенность при попытке описать и измерить процессы производства сложных программных продуктов *для психологической и профессиональной оценки их создателей*.

Технология регламентированного проектирования и производства крупных программных продуктов большими коллективами специалистов *принципиально и психологически отличается от индивидуальной разработки небольших программ или комплексов программ свободным методом* [1, 37, 43]:

- руководители таких больших коллективов специалистов должны быть способны выполнять **роль лидеров**, объединяющих и координирующих знания, навыки и труд над программным продуктом специалистов с разной профессиональной квалификацией и психологическими характеристиками;
- взаимодействие специалистов, **творческая и психологическая совместимость** в коллективе должны обеспечивать планируемое производство целостного программного продукта в реальном времени в заданные сроки и требуемого качества;

- при формировании коллектива и выполнении совместных групповых работ необходимо учитывать и использовать **особенности каждого специалиста** в коллективе, которые отличаются профессиональной квалификацией и психологическими характеристиками;
- обычно невозможно выделить **персональное авторство и ответственность** за отдельные функции и фрагменты, определяющие характеристики, качество, дефекты и риски программного продукта;
- качество поставляемого программного продукта зависит от **качества труда почти каждого специалиста** и его персональной квалификации, но не всегда можно выделить конкретного специалиста, **ответственного** за выявленные критические дефекты, ошибки и риски программного продукта.

Приступая к разработке сложного программного продукта, заказчики и руководители крупных проектов, прежде всего, должны понять и прогнозировать **целесообразно ли его создание** и оценить какова будет возможная эффективность применения готового продукта, оправдаются ли затраты на его разработку и использование. Поэтому в промышленности сложные технические проекты традиционно должны начинаться с анализа, планирования и разработки **экономического обоснования** заказчиками и руководителями проектов предстоящего жизненного цикла и применения предполагаемого продукта. В программных продуктах практически отсутствуют затраты на физические материалы и комплектующие компоненты, все их достоинства и недостатки практически **полностью определяются интеллектуальным трудом специалистов, их квалификацией**, психологическими характеристиками и умением слаженно, ответственно работать в большом коллективе.

В программной инженерии особое значение имеет параметр – **реальное время** как ограниченный, не пополняемый **человеческий ресурс**, который проявляется в трех базовых группах процессов создания комплекса программ:

- значения реального времени входит в реализацию жизненного цикла, во все **производственные процессы** и определяет планы и сроки создания программных продуктов, тем самым с позиций заказчика и пользователей этот человеческий фактор определяет принципиальную необходимость и возможность создания продукта к определенному сроку;

- реальное время определяет внутренние ***процессы исполнения комплекса программ***, качество, надежность и безопасность функционирования программного продукта в реальном времени в соответствии с его назначением, в зависимости от поступающей информации из внешней среды;
- реальное время определяет ***процессы во внешней среде***, в системе управления и обработки информации или у пользователей, влияет на динамику реализации функций программного продукта при его применении и является потребителем результатов его функционирования.

При планировании размеров, сложности и трудоемкости производства конкретных программных продуктов ***интуитивные оценки*** заказчиков, руководителей и специалистов, как правило, отличаются существенными ошибками. Они обычно психологически оптимистичны, и комплекс программ им кажется меньше по размеру, что приводит к нереальным оценкам количества функций, размера и сложности продукта и его компонентов. Следствием этого являются ***большие ошибки при планировании затрат времени***, качества и стоимости создания программных продуктов. Типичны ситуации, когда отставание сроков внедрения промышленных систем управления и обработки информации полностью ***зависит от неготовности и/или недостаточного качества для них программных продуктов***.

Заказчику продукта необходимо оценивать реальную потребность в его создании и возможную конкурентоспособность, а потенциальному разработчику предполагаемого программного продукта проводить оценку реализуемости проекта в условиях, ресурсах и сроки, предлагаемые заказчиком. ***Руководители*** конкретных проектов зачастую не в состоянии достаточно обоснованно определять, сколько времени и затрат интеллектуального труда специалистов потребуется на каждый этап программной части проекта системы, и не могут оценивать, насколько успешно выполняется в реальном времени текущий план производства. Это, как правило, означает, что програмная часть проекта системы с самого начала ***выходит из-под экономического контроля*** и возможна катастрофа с реализацией и завершением проекта всей системы в требуемый срок с заданным качеством и безопасностью применения.

Вследствие роста сфер применения и ответственности функций, выполняемых программами, резко возросла необходимость ***гаранти-***

рования высокого качества программных продуктов, регламентирования и корректного формирования требований к характеристикам реальных комплексов программ, их реализации и достоверного определения. Сложность анализируемых объектов – комплексов программ и **психологическая самоуверенность ряда руководителей и специалистов** в собственной «непогрешимости», часто приводят к тому, что реальные характеристики качества функционирования и безопасности применения программных продуктов остаются неизвестными не только для заказчиков и пользователей, но также для самих разработчиков. Отсутствие четкого декларирования в документах понятий и требуемых значений характеристик качества продуктов зачастую вызывает конфликты между заказчиками-пользователями и разработчиками-поставщиками из-за разной трактовки одних и тех же характеристик.

Лидеры – руководители производства крупных программных продуктов играют особую роль при формировании полноценных, корректных требований, которые должны осуществляться совместно заказчиком и другими заинтересованными лицами, с участием экспертов по тематической функциональной области назначения и применения продукта. Необходим подбор и организация специалистов для проверки корректности исходных требований при производстве крупных программных продуктов.

При формировании крупных производственных коллективов очень важно уметь **учитывать психологические особенности и профессиональную компетентность их лидеров и специалистов**. Для этого руководителям и специалистам – участникам проектирования и производства крупных программных продуктов полезно знать и использовать **основы психологии и характеристик людей**, участвующих в конкретном производстве сложных продуктов [14, 43]. Они должны уметь различать психологические характеристики и типы личностей в коллективе специалистов, использовать их общие и профессиональные характеристики и стимулировать производственные особенности творческих личностей, осуществлять правильный подбор специалистов с определенными психологическими и профессиональными характеристиками при формировании коллектива. Жизненный цикл развития коллективов специалистов в программной инженерии существенно зависит от психологических особенностей индивидуумов, размеров групп и коллективов специалистов. Для га-

рантии качества и безопасности применения программных продуктов руководителям и специалистам целесообразно знать и учитывать статистические *свойства дефектов и ошибок* в комплексах программ, их типы и источники, факторы, влияющие на их появление и обнаружение, а также возможные негативные последствия.

Основные концепции программной инженерии сконцентрированы и формализованы в *целостном комплексе систематизированных международных стандартов*, охватывающих и регламентирующих процессы жизненного цикла сложных комплексов программ. Несколько десятков современных стандартов допускают целеустремленный отбор необходимых процессов в зависимости от характеристик и особенностей конкретного проекта, а также формирование на их базе проблемно-ориентированных профилей стандартов для определенных типов проектов и/или предприятий. Существуют значительные *психологические трудности изучения и применения стандартов*, обусловленные конфликтами в обучении специалистов, между традиционным индивидуальным «хаотическим» (или *свободным*) программированием небольших компонентов и *регламентированным* коллективным производством крупных программных продуктов. При этом должна сохраняться концептуальная целостность применяемой совокупности стандартов и их эффективное, положительное влияние на процессы и результаты, на качество, надежность и безопасность программных продуктов при реальных ограничениях на использование доступных ресурсов.

Многообразие классов и видов сложных комплексов программ, обусловленное различными функциями и сферами применения систем управления и обработки информации, определяет формальные трудности, связанные с методами и процедурами *доказательства соответствия создаваемых и поставляемых программных продуктов* условиям контрактов, требованиям заказчиков и потребителей. По мере расширения применения и увеличения сложности систем выделились области, в которых дефекты, недостаточное качество комплексов программ *могут наносить катастрофический ущерб*, значительно превышающий положительный эффект от их использования. В таких *критических системах* (например, управления атомными электростанциями, крупными банками или системами вооружения) недопустимы проявления катастрофических рисков функционирования программных продуктов при любых искажениях исходных данных, сбоях, частичных отказах аппаратуры, ошибках пользовате-

лей и в других нештатных ситуациях. Подобные риски комплексов программ могут определять безопасность функционирования объектов, предприятий и даже страны. Вследствие этого резко **повысилась психологическая и юридическая ответственность коллективов, конкретных руководителей и специалистов за качество** результатов их труда и создаваемых программных продуктов.

Цель предлагаемого учебника – представить руководителям, преподавателям, студентам и аспирантам детализированные современные рекомендации и требования для освоения процедур и профессиональной квалификации специалистами для регламентированного проектирования и производства крупных программных продуктов. В основной группе представленных лекций (3 – 11) большое внимание уделяется психологическим и профессиональным характеристикам руководителей и специалистов, их обучению необходимой квалификации, конкретизируется, что они **должны знать и уметь** для успешной профессиональной деятельности на основных этапах жизненного цикла комплексов программ с учетом свойств и особенностей конкретных **человеческих факторов**.

Для повышения квалификации руководителей и специалистов, занятых в проектировании и производстве продуктов, представлены восемь лекций, отражающих конкретные **рекомендации и требования к профессиональной квалификации**, а также некоторые особенности психологических характеров для успешной реализации специалистами основных функций при таком производстве. В каждой лекции для обучения сформулированы систематизированные требования для достижения определенной профессиональной квалификации. Некоторые показатели уровня профессиональной квалификации, где это возможно, отражены в таблицах относительными значениями (рейтингами), которые комментируются описаниями свойств соответствующих уровней квалификации (см. лекцию 5).

Требования и рекомендации детализированы в рисунках (более 30), каждый из которых отражает определенную профессиональную квалификацию и детализирован необходимыми содержаниями знаний и навыков. Эти факторы включает широкий набор психологических и квалификационных характеристик руководителей и специалистов (свыше 200 рекомендаций и требований), которые практически определяют успех при производстве сложных программных продуктов высокого качества. Выборка тем из разделов лекций может быть *ори-*

ентирована на конкретные должности в производственном коллективе.

Для эффективного использования человеческих факторов необходимо *психологическое воспитание и профессиональное обучение* руководителей и специалистов различных категорий и квалификаций методам и процедурам обеспечения и гарантии качества производственных процессов сложных программных продуктов. Для этого необходимо изучать различные тематические сферы и потребности в различных категориях специалистов по программной инженерии, реальные процессы их деятельности, методы обучения, оценки профессиональной квалификации и роли в коллективах. Это требует непрерывного совершенствования обучения и повышения квалификации заказчиков, разработчиков и пользователей в области программной инженерии, освоения ими современных методов, процессов и международных стандартов, а также *высокой корпоративной культуры работы коллективов специалистов*, обеспечивающих жизненный цикл сложных и особенно критических программных продуктов. Для этого руководителям и специалистам полезно знать основы современной психологии и взаимодействия людей в коллективах при производственном производстве, которые изложены в первых двух лекциях с некоторым *акцентом на коллективное создание программных комплексов*.

Важнейшим умением, приобретаемым в ходе обучения, должна становиться склонность к *непрерывному самообучению*. Целевая практическая направленность курса приводит к доминированию практики над теорией в силу необходимости использования результатов обучения непосредственно в бизнес-практике. Качество обучения должно обеспечиваться благодаря практической ориентации, умению охватить проблему в целом, не упустив существенных деталей, ограничений ресурсов и сроков производства продукта. При этом важными требованиями к преподавателям являются необходимость адаптации к уровню знаний учащихся, использованию передовых технологий, инструментария, стандартов, а также методологических решений, независимых от конкретного производителя программного продукта. Важными принципами являются ориентация на полный охват жизненного цикла крупных программных продуктов, а также на воспитание должного уровня дисциплины их проектирования и производства. При этом предполагается, что обеспечена возможность применения соответствующих современных технологий и инструмен-

тальных средств автоматизации труда, которые ниже не рассматриваются.

Лекция 1

ЧЕЛОВЕЧЕСКИЕ ФАКТОРЫ ПРОИЗВОДСТВА ПРОДУКТОВ С ПОЗИЦИИ ИНЖЕНЕРНОЙ ПРОФЕССИОНАЛЬНОЙ ПСИХОЛОГИИ

Основные понятия инженерной профессиональной психологии

Для *анализа и учета человеческих факторов* в профессиональной деятельности руководителей и специалистов разных квалификаций целесообразно освоить современные *основы психологии производственных коллективов*. Влияние человеческих факторов на профессиональную трудовую деятельность людей изучается *инженерной и профессиональной психологией* [1, 43, 46]. Руководителю проекта крупной системы, организуя работы необходимо *уметь оценивать* особенности психологии специалистов, с которыми предстоит работать и возможное их влияние на успех проекта. Для этого ему полезно знать основы профессиональной психологии и методы управления человеческими ресурсами. Элементы и понятия этой науки важно использовать для обеспечения профессиональной квалификации специалистов при освоении программной инженерии проектирования и производства *крупных программных продуктов*. Фундаментом этих знаний являются общие свойства личностей – субъектов производственных процессов – рис. 1.1.

Понятие субъекта в инженерной психологии рассматривается на пересечении двух плоскостей: отдельных трудовых функций и конкретного человека. Субъект объединяет, интегрирует, запрещает и разрешает, контролирует выполнение отдельных функций системы. Специалист, получивший сложное непривычное задание, оказывается перед необходимостью найти компромисс между скоростью и точностью (величиной отклонения) решения производственной задачи. Обычные координации усилий, способность быстро собраться или расслабиться, фоновые напряжения и другие возможности субъекта

могут стать препятствием для приспособления к особенностям задания.

Субъект знает свои возможности исполнения разных задач. Он классифицирует задачи на доступные и недоступные для него, так как помнит результаты предыдущих действий и свои ошибки. Соотнеся ситуацию с предполагаемым результатом и оценив условия задачи, субъект может заявить, что не сумеет выполнить задачу в данных условиях.

Руководителям коллективов целесообразно учитывать общие основы современной инженерной профессиональной психологии, знать и уметь использовать:

- особенности профессионального труда субъектов, характеризуемого и учитываемого инженерной психологией;
- понятие субъекта на пересечении двух плоскостей: отдельных трудовых функций и конкретного человека;
- психологическое наполнение трудовой деятельности субъектов;
- понятие профессиональной задачи субъекта и ее социальный психологический характер;
- способы управления поведением субъекта для выявления человеческих ошибок;
- построение психологической модели труда и средства, фиксирующие рабочее пространство и время;
- психологическое содержание динамики труда профессионалов сложных систем;
- действие, вложенное в реальное пространство и реальное время деятельности, должно иметь допуск на точность исполнения;
- действие – как акт борьбы человеческой воли с инерцией лени и успокоенности, которое может становиться разрушительным;
- труд может приносить большее удовлетворение специалистам, у которых высок уровень интеллектуальной активности, глубокие и разносторонние знания и большой опыт работы.

Рис. 1.1

Действуя, субъект может решать несколько задач, которые начались в разное время и в разные моменты закончатся. Задачи при этом выполняются в разных пространствах и требуют разной пространственной точности.

Накапливая опыт, появляется возможность переходить по сложной системе связей от одной задачи к другой [43]. Субъект выполняет эти переходы, продолжая решать задачу или решая несколько задач сразу. Работая с разными группами людей, субъект приспосабливается к каждому из них, запоминая привычки, поступки и особенности поведения. Субъект – **сложная система опыта**, позволяющая отслеживать и продвигать вперед развитие отношений с разными людьми, которые идут с разной скоростью, сравнивать успехи и неудачи и делать все это в рамках какой-то конкретной ситуации, в ходе решения конкретной задачи. Опыт – сложная мыслительная работа объединения и разделения, привлечения и откладывания отдельных задач в ходе решения одной из них. При этом в деле участвуют восприятия и воспоминания, мысли, движения, ценности и переживания.

Понятие **профессиональной задачи** ориентирует субъекта на будущее решение и исполнение. Субъект получает задание от более значимого другого специалиста – в этом **социальный психологический характер задачи** [1]. Его значимость определяется тем, что он может дать задание, которое субъект **обязан** выслушать, должен подтвердить свое согласие или готовность выполнять задание. В задаче указывается цель, средства, сроки и пространственная точность, каким путем можно прийти к требуемому результату. Продумывание исполнения и оценка сложности задачи субъектом дают **план исполнения**.

В профессиональной деятельности много примеров **группового решения задач**, что имеет свои особенности. Члены группы обсуждают все, что связано с действием: формы коммуникации, временные и пространственные пункты координации, режим исполнения. В ходе исполнения формулировка задачи служит точкой отсчета, критерием, который позволяет субъекту судить о приближении к цели, как пространственная и временная форма, в которую должно быть уложено исполнение в конкретных условиях. Функцию контроля за исполнением берет на себя один из членов коллектива – руководитель. При разборе выполнения при оценке неудач в отдельных фрагментах ведутся поиски виновного, того, кто совершил ошибку.

В содержание профессиональной деятельности входят не только технологические, но и психологические компоненты. Квалифицированный специалист должен всегда видеть и учитывать **психологическое наполнение трудовой деятельности**. Опыт освоения задачи специалистом основан на удачных и неудачных выполнениях. Слож-

ность задачи определяется тем, насколько быстро и легко удается специалисту освоить критические операции, необходимые для ее выполнения, и определяется по времени решения. Сложность задачи тем выше, чем больше одиночных, уникальных подзадач содержится в ней.

Для определения сложности задачи решающее значение имеет **опыт субъекта**. Факторы, определяющие сложность практической задачи: неопределенность, новизна и неожиданность событий и объектов окружающего мира, неполнота, неясность, неточность информации, поступающей к специалисту. Задачи становятся более сложными при временном дефиците и жестких сроках, для них предполагается построение особых планов и стратегий, для которых характерны: неполнота, неопределенность, незавершенность. Сложные задачи связаны с **высокой ответственностью** за промахи и социальным (или административным) давлением на исполнителя. При опоздании или пространственной ошибке специалист переходит на **уровень анализа**, когда к делу подключаются опознание ориентиров и воспоминания.

Динамика профессиональной инженерной психологии – это область знания и сфера практики, в центре ее находится человек, управляющий технологическим процессом в системе, в производстве продукта. Вместе с другими науками инженерная психология призвана участвовать в изучении, усовершенствовании и проектировании человеческого труда. Быстрое развитие и усложнение техники, расширение условий эксплуатации ведут к возрастанию количества человеческих ошибок и их цены. Общая тенденция развития индустрии заключается, прежде всего, в непрерывном росте размеров и сложности систем. Нарушения правил эксплуатации или ошибки, допускаемые при использовании больших систем, приводят к серьезным последствиям. Большие системы должны быть сконструированы и построены таким образом, чтобы вероятность угрозы сбоев и ущерба была минимальной, а свобода действий специалистов – как можно больше ограничена и предоставлена возможностям строгого контроля.

Специалисты и конструкторы часто оказываются в ситуации, когда они должны определить, **кто виноват** в происшедшем, а потом распределить ответственность, что часто становится **причиной конфликта**. Поэтому в тех случаях, когда интерес к безопасности системы велик, а требования к персоналу резко возрастают, но при этом не поддаются точному определению, важно обратиться к понятию **ошиб-**

ка человека и вспомнить о его значении для определения стратегии проектирования и производства сложных систем. Способы управления поведением человека постольку важны, поскольку их необходимо знать для предупреждения и выявления человеческих ошибок [1, 43]. Ошибка в человеческом действии может быть прямо установлена только в том случае, если имеется четкое предварительное описание (эталонное) той функции или операций, которую должен выполнить человек.

Для построения *психологической модели труда* необходимо иметь средства, фиксирующие рабочее пространство и время. Субъект удерживает цель, он строит план исполнения действия и контролирует выполнение, фиксирует ошибки и подвергает анализу выполненные действия. В процессе выполнения субъект анализирует ситуацию, оценивает ее и выбирает один из вариантов действия, продолжая выполнять движение. Поведение специалиста запускается незнакомыми событиями в системе. Структура его деятельности состоит в том, чтобы оценить ситуацию и спланировать подходящую последовательность целенаправленных действий, и зависит от знаний о процессах, функциях и структуре системы.

В *инженерной психологии профессиональный труд* характеризуется следующими особенностями [1, 43]:

- на некоторых этапах специалист должен перерабатывать за короткие периоды времени большие объемы информации, в такие периоды труд бывает очень напряженным;
- технологический процесс является достаточно сложным, велика ответственность за ошибки и цена ошибки;
- возможны быстрые переходы от периодов чистого ожидания отклика системы к напряженной работе, переходы могут происходить неожиданно, а возникшая при этом ситуация часто непохожа на другие;
- возможно быстрое изменение состояния системы и внешней среды.

Ситуация характеризуется пространством и временем. Однако часто ее характеристика остается статичной – только в терминах пространства. В ситуацию необходимо ввести субъекта, решающего задачу, необходимо учсть напряжение, которое сопровождает субъекта до окончания действия. Когда субъектом совместного действия является *группа специалистов*, выполнение трудовых функций требует

пространственной координации индивидов. Синхронизация движений в групповой деятельности достигается предварительным планированием и использованием команд в ходе исполнения, ускорением и замедлением движений, ориентацией на ряды внешних событий, сроки. Выполнение трудовых функций невозможно помимо общения и воздействия группы на индивида.

Психологическое содержание динамики труда рассматриваются профессионалов характеризуется [43]:

- **сбором информации об объекте, системе и среде**, анализом ситуации в целом, выделением вариантов и определением их последствий, координацией действий, контактами с различными специалистами, особенностями труда и человека;
- **особенностями психологического содержания** труда специалистов: скоростью поступления и объемом информации, сложностью задач, ответственностью, нерегулярностью сложных ситуаций и скорость их развития, зачастую длительные периоды наблюдения и пассивного ожидания;
- **подготовкой к выполнению** сложной деятельности: изучением документов, проигрыванием ситуаций на моделях и тренажере, продумыванием наиболее сложных моментов, подготовкой для точного выполнения задачи;
- **факторы, определяющие сложность** практической задачи: неопределенностью, новизной и неожиданностью событий и объектов окружающего мира, неполнотой, неточностью информации, поступающей к профессиональному, неопределенностью, незавершенностью планов, стратегий и вариантов действий, временным дефицитом, социальным давлением на исполнителей;
- **выполнение трудового действия:** сбор информации, анализ ситуации в целом, выделение вариантов и определение их последствий, момент выбора, эмоциональный и исполнительный аспекты, исполнение действия, разделение задач и слитность цели;
- **анализ последствий действия:** коллективный разбор результатов, разные видения произошедшего и разные интерпретации последовательности событий, продумывание произошедшего и переживание последствий, трудность осознания и интерпретации результатов, влияние команды на формирование стандартов выполнения операций у каждого отдельного специалиста.

Временные характеристики труда и действия содержат: начало, конец, последовательность движений, настоящее, прошедшее, будущее, повторения, циклы, сроки, уникальность события и возможность его воспроизведения или повторения, непрерывность потока и дискретность ряда событий, этапы действия. Между элементами действия возникают динамические отношения. **Динамика действий** связана, прежде всего, со стремлением довести действие до конца – в **направленности на результат** для преодоления препятствий волевыми актами.

Пространственные характеристики действий определяются теми предметами и промежутками, которые прямо или косвенно относятся к действию. Топологическая характеристика включает определение мест субъекта, цели, пути движения и расположения препятствий, различие районов, их внешних и внутренних границ. В психологии принято так использовать **понятие действие**, будто исполнение всегда **остается совершенным**. Однако действие, вложенное в реальное пространство и реальное время, должно получить **допуск на точность исполнения**. Понятие действия как сплошной напряженной структуры включает различные возможности исполнения: **от точного до ошибочного**. Действие запускается и поддерживается заданием, задачей. При отклонениях в условиях исполнения или в характере процесса неизбежно обращение к верbalным формам. Этим обеспечивается идентификация привычной обстановки для выполнения действия.

Действие – это **живой процесс исполнения**, который невозмож но удержать неизменным, он колеблется – то возрастает, то ослабевает. Чтобы преодолеть инерцию предыдущего состояния, необходимо отбросить все влияния и **сосредоточиться на цели**. Сосредоточившись на результате, человек может отвлечься от контроля за безопасностью. Человек сам по себе является сложной системой, состояние которой может отклониться и стать неблагоприятным, если не для результата, то для самого человека. С каждым успешным исполнением позитивная, нужная **тревога за результат** становится все меньше, контроль за безопасностью все слабее, требуется особое усилие для контроля за исполнением, за влиянием опасных факторов. Постоянным **усилием воли** специалист должен сосредотачиваться на привычных правилах безопасности. Особенно в групповой деятельности, где расхолаживающее влияние опыта становится еще сильнее, поскольку система еще сложнее. Действие – это **акт борьбы человеческой воли** с

инерцией лени и успокоенности, которая может становиться разрушительным безразличием.

В труде есть *активная и пассивная стороны*. Выполняя нормальные трудовые задачи в автоматическом режиме, опытный специалист может заняться выполнением нового действия, поскольку его ум частично освободился. Наблюдая трудовую деятельность, нельзя установить момент, когда начинается действие, врезающееся в поток операций. По внешним движениям можно наблюдать поток операций, а для выделения действий из потока потребуется отчет профессионала. Предполагается, что детали и части будут выполнены, если будет выполнена суть задачи. Это должно обеспечиваться обучением и выдерживанием технологии. Чтобы установить, обнаружить действие, необходим ряд критериев, среди которых такие, которые основаны на использовании сообщений действующего человека.

Зачастую технология строится в процессе эксплуатации, и специалист с его знаниями, навыками и опытом играет одну из важнейших ролей. Выстраивая режим самостоятельно, "*от нуля*", специалист иначе видит его, чем при регулировании в соответствии с жестко заданными требованиями. Благодаря этому обогащается содержание труда, труд приносит большее удовлетворение исполнителю. В таком режиме могут работать только те специалисты, у которых *высок уровень интеллектуальной активности*, глубокие и разносторонние знания и большой опыт работы. Особенно следует подчеркнуть преимущественное положение специалиста по отношению к тем, кто определяет наставления, правовые, технологические и медицинские нормы эксплуатации, строит систему и содержание подготовки и повышения квалификации.

Инженерная психология уделяет много внимания режимам работы специалистов. Чтобы обогатить труд профессионалов, необходимо выйти за рамки непосредственной трудовой ситуации и понаблюдать их во время учебы: на лекциях и групповых разборах, при обмене опытом, во время тренажерной подготовки, где при отработке уникальных, критических или аварийных ситуаций обнаруживаются глубоко скрытые особенности человеческого характера. Рассмотрение труда специалистов *как творческого* должно решать психологические проблемы, связанные с неудовлетворенностью отношениями в трудовых коллективах, конфликтами и т.д. Без знания творческой стороны

характеристика труда будет неполной и дажеискаженной, поскольку акценты делались бы только на отрицательных сторонах труда.

Основы психологии профессиональных субъектов при производстве сложных продуктов

Производство определенной продукции определяется **потребностями людей** и стимулируется конкретной нуждой в соответствии с профессией, культурным уровнем и личностью индивида. Все потребности человека по основным признакам можно подразделить на абсолютные и относительные, положительные и отрицательные, прямые и косвенные, общие и особенные, постоянные и временные, беспрерывные и прерывные, индивидуальные и коллективные, частные и государственные и т.д. [42, 51] (рис. 1.2).

*Руководителям коллективов целесообразно учитывать общие
психологические свойства профессиональных субъектов
знать и уметь использовать:*

- основные потребности субъектов: первичные – необходимые для жизнеобеспечения и вторичные – обусловленные психологией профессиональной деятельности;
- психологические причины мотивации деятельности личностей: внутренняя – личные цели и заинтересованность или внешняя – материальные и моральные интересы;
- система, факторы и степень стимулирования личностей: административное воздействие, степень удовлетворенности, значимость, справедливость, ожидание оценок или поощрения;
- психологическая ориентация и факторы формирования карьеры личностей;
- психологические характеристики и типы личностей – профессионалов: интроверт или экстраверт; сенситивная или интуитивная личность; мыслящая или ощущающая (эмоциональная) личность; рассудительная или восприимчивая.

Рис. 1.2

Конкретные потребности людей выражаются в тех или иных объектах, позволяющих удовлетворить нужду тем способом, который присущ данному предприятию, коллективу и человеку. Производи-

тели пытаются сформировать связь между выпускаемыми товарами и потребностями человека. Тот или иной продукт пропагандируется и рекламируется как средство удовлетворения одной или нескольких специфических нужд человека. Потребность в дальнейшем превращается в спрос или запрос на конкретный продукт – товар. *Спрос* – это потребность, подкрепленная покупательной способностью людей на определенный товар [1, 16, 37].

Участие человека в экономической деятельности характеризуется его потребностями и возможностями их удовлетворения, которые обусловлены, прежде всего, его личными и профессиональными способностями и многими обстоятельствами. В условиях рынка каждый человек выступает, с одной стороны, как потребитель материальных благ, производимых предприятием, а с другой – как обладатель рабочей силы, знаний, умений и способностей, необходимых различным предприятиям и организациям для осуществления своей экономической деятельности. Взаимодействие потребителей и производителей, равновесие спроса и предложения осуществляется через систему человеческих потребностей на рынках труда и производства, экономических ресурсов и материальных благ.

На рынке *труда* величина спроса на рабочую силу определяется потребностями производителей, количеством выпускаемой продукции, необходимым числом рабочих мест, продуктивностью труда работников, а предложение – численностью желающих работать (трудовых ресурсов), их профессионально – квалификационным составом. На рынке *товаров* объем спроса формируется исходя из потребностей в той или иной продукции, а предложение – с учетом наличия у производителей ресурсов, в том числе количества и состава рабочей силы. Система человеческих потребностей является также основой совершенствования предприятия и управления трудом в современном производстве.

Современная экономическая теория классифицирует все человеческие потребности как первичные и вторичные. *Первичные* потребности являются по своей природе физиологическими и, как правило, врожденными, заложены в человеке генетически. *Вторичные* потребности по их происхождению можно считать *психологическими*. Они появляются с опытом и различаются в большей степени, чем первичные.

Трудовая деятельность персонала в любом предприятии является целенаправленной и мотивированной. Под **мотивацией** в научном и практическом менеджменте понимается процесс побуждения себя и других к деятельности для достижения личных и общих для предприятия целей. Совпадение личных и общественных целей служит основой управления трудом персонала на предприятиях, поскольку выбор и обоснование единой производственной или экономической цели является первой функцией современного менеджмента, главной задачей персонала на всех уровнях управления [11, 43, 51]. Внешняя мотивация (например, деньги) достигает того же результата, ибо кто-то может выполнять задачу ради денег, которые ему нужны, хотя сама эта задача не вызовет у него интереса. Поэтому вместо того, чтобы надеяться заинтересовать группу материально, руководитель проекта должен, прежде всего, обратить внимание на те задачи, выполнение которых ни у кого не вызовет особого энтузиазма. Поэтому важная цель руководителя проекта состоит в том, чтобы подобрать отдельного исполнителя для решения конкретной задачи. Мотивацию следует рассматривать с точки зрения заинтересованности отдельного лица соответствовать требованиям конкретной задачи или даже отличаться конкретным поведением.

При планировании и организации работы коллектива каждый руководитель определяет, что конкретно должно выполнить в целом предприятие, что, когда и как должен сделать тот или иной сотрудник. Если выбор и обоснование решений о цели совместной деятельности были сделаны правильно и мотивировано, то всякий руководитель получает возможность координировать усилия своих сотрудников и наилучшим образом использовать потенциальные возможности каждого работника и всего персонала. Таким образом, правильное управление персоналом и хорошая мотивация создают условия не только для успешного достижения общей конечной цели производства, но и для полной трудовой отдачи всех работников, развития их творческой активности и удовлетворения как физических, так и духовных потребностей.

Модель производственного поведения работников в процессе трудовой деятельности должна учитывать не только воздействие внешних факторов со стороны руководителя или всего организационного окружения, но и многих внутренних человеческих мотивов. Включаясь в трудовую деятельность, каждый специалист в любом

предприятии интересуется не столько общими целями и результатами работы, сколько возможностью удовлетворить свои личные потребности. Каждого работника при рыночных отношениях интересуют, прежде всего, его **личные цели и задачи, затраты и результаты**: что он должен конкретно и в каких условиях делать; какие затраты физических и умственных усилий от него потребуются; какими свободами он должен жертвовать во имя общего дела; с какими людьми и как ему предстоит взаимодействовать; как будет оцениваться и вознаграждаться его труд за выполнение работы. От этого и ряда других факторов зависит не только удовлетворенность человека собой, своим взаимодействием с предприятием, но и **мотивация его личного отношения к работе**, величина трудового вклада в общие производственные и финансовые результаты деятельности предприятия. **Сочетание личных и общественных мотивов деятельности** человека на предприятиях является одной из важнейших задач как управления производством, так и в целом всего производственно-го менеджмента.

Путь к эффективному производству проходит через понимание побудительных мотивов человека к трудовой деятельности, представляющей собой совокупность воздействия внутренних и внешних сил к осуществлению определенных действий. Процесс воздействия на человека в целях побуждения его к выполнению запланированной деятельности путем пробуждения соответствующих внутренних мотивов и внешних поступков принято называть **мотивацией**. Взаимодействие мотивов и мотивации в трудовой деятельности человека составляет основу управления трудом персонала (см. рис. 1.2).

В зависимости от того, какие цели преследует мотивация, можно выделить два вида мотивирования: внешнее и внутреннее. **Внешнее** мотивирование представляет собой своего рода процесс административного воздействия или управления: руководитель поручает работу исполнителю, а тот ее выполняет. При таком виде мотивирования руководителю надо знать, какие мотивы могут побуждать конкретного работника выполнить работу качественно и в срок. Это может быть как нормальная оплата результатов работы или премия, так и простая похвала или иной вид морального поощрения.

Внутреннее мотивирование является более сложным процессом и предполагает формирование определенной мотивационной струк-

туры человека. В этом случае следует найти **психологический способ усиления** желательных качеств личности и ослабления отрицательных факторов, например, снижения монотонности труда. Внутренняя мотивация в большей степени определяется содержанием или значимостью самой работы. Если она интересует работника, позволяет реализовать его природные, профессиональные способности и склонности, то это само по себе является сильным мотивом к активной трудовой деятельности. Наряду с этим существенным внутренним мотивом может быть значимость работы для развития квалификации и определенных качеств человека, а также полезность данного вида деятельности для профессиональной группы работников.

Стимулирование выполняет на предприятии важную роль действенных **мотиваторов** или основных носителей интересов работников. В качестве стимулов могут выступать отдельные предметы, действия других людей и многие иные ценности, которые могут быть предложены работнику в компенсацию за его повышенные умственные или физические усилия. Мотивация и стимулирование персонала оказывают значительное воздействие на развитие у работников таких важных характеристик их трудовой деятельности, как качество работы, результативность труда, старание, настойчивость, добросовестность. В управлении производством на предприятиях применяются **две основные формы мотивации**: по результатам и по статусу. Мотивация по **результатам** обычно применяется там, где можно сравнительно точно определить и разграничить результат деятельности одного работника или группы сотрудников. Мотивация по **статусу** основана на интегральной деятельности сотрудника, учитывающей уровень его квалификации, качество результатов труда, отношение к работе.

Действенная мотивация означает, что стимулирующим источником являются поощрения, когда отдельные лица полагают, что их поведение приведет к определенным результатам в виде оплаты или похвалы. В качестве принятых в настоящее время моделей мотивации на основе отношений обмена считаются рассматриваемые далее теории ожидания и справедливости. К известным внешним поощрениям относятся: возможность развивать деловые отношения с коллегами и начальством, оплата, продвижение по службе, дополнительные льготы и гарантия занятости. Рассматривая **мотивацию с психологической** стороны, можно выделить три вида поощрений:

хологической точки зрения, можно выделить три типа профессионалов [51]:

- **люди с целевой внутренней ориентацией**, получающие достаточно мотивации от работы, которую выполняют, к этому типу относятся «технари», мотивация которых вызвана интеллектуальными задачами по разработке программного продукта;
- **люди с самоориентацией**, мотивация которых основана на личном успехе и признании, они заинтересованы в разработке программного продукта, преследуя при этом личные интересы;
- **люди с внешней ориентацией**, мотивация которых требует присутствия и деятельности сотрудников, так как создание программ становится все более ориентированным на пользователя, такие люди все чаще вовлекаются в разработку программного продукта.

Мотивация на основании внутреннего процесса означает мотивацию отдельных лиц, обусловленную, главным образом, внутренним процессом и возможностью заниматься только теми видами деятельности, которые вызывают определенный интерес. Такие люди зачастую уходят от выполнения задач, направленных на достижение конкретной цели, и обращаются к задачам, решение которых доставляет большее удовольствие. К известным факторам, вызывающим внутреннее удовлетворение, относятся сложность задачи, возможность взять на себя ответственность и руководить работой, выполняемой другими, разнообразие, творческий потенциал, возможность проявить свои способности и добиться ответной реакции.

Внешняя мотивация на основе самооценки формируется внешним образом, когда отдельное лицо ориентируется, в основном, на других. В этом случае наиболее идеальным считается принятие обязанностей, которые ожидает от отдельного лица эталонная группа. Отдельное лицо пытается оправдать ожидания других, выявляя ответную социальную реакцию и согласуя ее с представлением о самом себе. Если ответная реакция положительна, отдельное лицо считает необходимым сообщить об этом членам эталонной группы, либо группы равных по положению. Отдельное лицо ведет себя таким образом, чтобы удовлетворить ожидания членов эталонной группы сначала для завоевания признания, а затем и положения.

Внутренняя мотивация на основе самооценки формируется внутренним образом, когда отдельное лицо рассчитывает в основном на самого себя. Такая мотивация принимает форму установления от-

дельным лицом собственных внутренних норм, которые могут считаться идеальным основанием для его поведения. При этом отдельное лицо пытается сначала упрочить представления о собственной компетентности, а затем добиться повышения ее уровня. Для таких лиц важно, чтобы их усилия были крайне необходимы для достижения определенных результатов и чтобы их идеи и действия способствовали успешному выполнению задачи.

Системы стимулирования относятся к тем организационным аспектам, которые побуждают к действию, направляют или поддерживают поведение. К наиболее распространенным относятся поощрительные, административные и социальные системы стимулирования, направленные на решение задачи. Поощрение является типичной системой материальной компенсации и продвижения по службе, стимулирование решения задачи связано с мотивационными аспектами планирования заданий и постановки задач. Мотивационное влияние рабочей группы или организации в качестве социальной системы определяет систему социального стимулирования.

Существуют следующие **основные причины мотивации** и **системы стимулирования** [51]. Мотивационные свойства системы **поощрительного стимулирования** тесно связаны с ожиданием того, что дополнительные усилия будут вознаграждены повышением оплаты труда, а сама величина оплаты послужит дополнительным стимулом для отдельного лица. Система стимулирования **решения задачи** описывает способ повышения внутренней мотивации к делу, отдельное лицо выполняет работу либо она интересна, либо является для него вызовом. Система **административного стимулирования** побуждает к действию, направляет или поддерживает поведение сотрудника посредством целого ряда первопричин. В системе социального стимулирования эффективно мотивированные отдельные лица реагируют на нормы и санкции, применяемые организацией.

Сила стремления действовать некоторым образом зависит от **степени ожидания** того, что это действие приведет к определенному результату, а также от привлекательности результата для отдельного лица. При этом рассматриваются в основном следующие соотношения:

- соотношение трудозатрат и результатов исполнения, или вероятность того, что приложенные отдельным лицом трудозатраты приведут к выполнению той или иной задачи;

- соотношение исполнения и поощрения, или степень, до которой отдельное лицо полагает, что исполнение на определенном уровне приведет к желаемому результату;
- соотношение поощрения и личных целей, или степень, до которой организационные поощрения удовлетворяют личным целям или потребностям отдельного лица, либо привлекательность для него возможных поощрений.

Факторы мотивации относятся к элементам, которые делают труд человека более разнообразным. В частности, **удовлетворенность работой** в значительной степени определяют следующие **факторы**: достижение, признание, сама работа, ответственность, продвижение и повышение по службе. Эти факторы мотивации (или удовлетворенности) связаны с длительными положительными эффектами, оказывающими влияние на выполнение работы, тогда как факторы неудовлетворенности обуславливают только краткие перемены в отношении к работе и ее выполнению и быстро приводят к возврату на предыдущие уровни.

Причины неудовлетворенности работой отличаются от тех, которые вызывают удовлетворенность ею. На поведение сотрудников на работе оказывают влияние две совершенно разные группы факторов. Одни из них - **факторы мотивации**, а другие – **факты неудовлетворенности**. Факторы неудовлетворенности связаны с отношением человека к окружению, в котором он выполняет работу. К ним относятся условия труда и его оплата, а также межличностные отношения, связанные с ситуацией, в которой человек трудится. Факторы удовлетворенности связаны с тем, что человек делает, тогда как факторы неудовлетворенности - с ситуацией, в которой человек что-то делает. Одного исключения факторов неудовлетворенности недостаточно.

Стимулирование посредством поощрений относится к материальным или психологическим видам вознаграждения за труд. Внешние поощрения представляют собой виды вознаграждения, выдаваемые отдельному лицу другими, в том числе в виде денег, льгот, признания, похвалы, а внутренние поощрения относятся к наградам, выдаваемым самому себе (например, **чувство удовлетворения достигнутым**). Для повышения производительности с помощью внутренних поощрений последние должны удовлетворять индивидуаль-

ные потребности, причем работники должны быть убеждены в том, что их усилия будут вознаграждены.

Ожидание – это представление о том, что усилия приведут к достижению желаемых целей исполнения. Такое восприятие, как правило, основано на прошлом опыте отдельного лица и уверенности в себе, а также на воспринимаемой сложности или цели результатов деятельности. В ожидании предполагается, что сила стремления действовать некоторым образом (мотивация) зависит от степени ожидания того, что это действие приведет к определенному результату, а также от привлекательности результата для отдельного лица.

Степень мотивации определяется воспринимаемыми возможностями достижения успеха, т.е. предположением или ожиданием того, что одно приведет к другому. При этом учитывается направленность мотивации, когда поведение отдельных лиц побуждает к действию, они ищут определенные варианты поведения. Это представление о том, что усилия приведут к достижению желаемых целей исполнения, вероятность того, что приложенные усилия приведут к исполнению. Такое представление или восприятие, как правило, основано на прошлом опыте отдельного лица и уверенности в себе (нередко называемой собственной эффективностью), а также на воспринимаемой сложности нормы или цели исполнения.

Значимость – это степень привлекательности или непривлекательности конкретного результата, который представляет собой все, к чему приводит определенное поведение. Это желательность результата для отдельного лица или значение, которое он лично придает поощрениям, а также функция потребностей, целей и ценностей. Для того чтобы у работников имелись большие стимулы к проявлению желательного поведения, необходимо наличие высокого уровня ожидания и значимости результатов. Отдельное лицо действует некоторым образом в зависимости от ожидания того, что это действие приведет к определенному результату, а также от привлекательности для него этого результата. Ожидания по принципу «усилия – исполнение – поощрение» определяют **высокую или низкую степень мотивации**.

Справедливость (или несправедливость) является **внутренним психологическим состоянием** отдельного лица. Справедливость подразумевает такое же честное отношение к отдельному лицу, как и ко всем остальным. Отдельные лица сравнивают свой и чужой

трудовой вклад и результаты труда и соответственно реагируют, стремясь устраниить несправедливость. Они озабочены не только абсолютными величинами вознаграждения их усилий, но и соотношением этих величин с теми, что получают другие. Справедливость представляет собой **процесс социального выравнивания**, в результате которого отдельные лица сравнивают свои и чужие результаты. В зависимости от применяемых критериев сравнения отдельные лица вырабатывают разное восприятие самого понятия справедливости.

Руководитель программного проекта может определить основной результат, которого стремится достичь каждый работник и решить, какие именно уровни и виды исполнения требуются для достижения целей, убедиться в том, что желаемые уровни исполнения возможны. Для этого следует связать желаемые результаты с исполнением, проанализировать сложившуюся ситуацию на предмет противоречивых ожиданий, обеспечить достаточно весомые вознаграждения и убедиться в **справедливости общей системы поощрений** для каждого сотрудника.

Когда отдельное лицо считает, что его вклад оказывается наравне со сравниваемым, а вознаграждение – меньшим, или его вклад больше, а вознаграждение – наравне со сравниваемым, то он испытывает чувство несправедливости. Каждый работник предприятия должен иметь правильное представление о поощрениях, если работники должны поощряться в большей степени за качество, а не за объем выполненной работы, этот факт должен быть ясно до них доведен. Принимая во внимание связанный с восприятием характер справедливости, вклады могут быть разделены на **следующие категории** [16]:

- трудовые вклады, в том числе и факторы, позволяющие отличить один труд от другого, как правило, к ним относятся ответственность, влияние, квалификация, образование и условия труда, необходимые для самой работы;
- личные свойства сотрудников, в том числе индивидуальные качества результатов, которые, по мнению отдельных лиц, могут выгодного отличать их от остальных работников предприятия: опыт, стаж работы и дополнительный уровень образования;
- производительные вклады, в том числе дополнительные усилия и результаты, отличающие одних работников от других.

Критерии сравнения справедливости, применяемые отдельными лицами, могут быть разделены на следующие классы:

- должностная справедливость – отдельные лица сравнивают оплату своего труда и других лиц, занимающих ту же самую должность в коллективе;
- справедливость в пределах предприятия – отдельные лица сравнивают оплату своего труда и других лиц, занимающих разные должности в коллективе;
- профессиональная справедливость – отдельные лица сравнивают оплату своего труда и других лиц, занимающих ту же самую должность в других предприятиях;
- групповая справедливость – отдельные лица сравнивают оплату своего труда и других лиц в сходных группах (например, по возрасту или уровню образования);
- справедливость по отношению к самому себе – отдельные лица сравнивают оплату своего труда в разные периоды времени.

Руководитель проекта может определить основной результат, которого стремится достичь каждый работник, решить, какие именно уровни и виды исполнения требуются для достижения целей. Убедиться в том, что желаемые уровни исполнения возможны, связать желаемые результаты с исполнением, проанализировать сложившуюся ситуацию на предмет противоречивых ожиданий, обеспечить достаточно весомые вознаграждения и убедиться в **справедливости общей системы поощрений** для каждого сотрудника.

Поощрения относятся к материальным или психологическим видам вознаграждения за труд. Для повышения производительности с помощью внутренних поощрений последние должны удовлетворять индивидуальные потребности, причем работники должны быть убеждены в том, что их усилия будут вознаграждены. Из понятий справедливости руководители могут сделать выводы: каждый сотрудник должен иметь правильное представление о поощрениях; работники стремятся получить многостороннее представление о своем поощрении; а, кроме того, работники предпринимают конкретные действия в зависимости от того, как они воспринимают реальность. У членов коллектива всегда будут **стимулы к снижению несправедливости**. Для этого у них могут быть следующие варианты выбора:

- изменить свой вклад, снизить производительность;
- изменить результаты труда, попытаться повысить оплату;

- изменить собственное восприятие, рационализировать или изменить отношение к поощрению либо вкладу на основе собственного восприятия;
- изменить восприятие других, попытаться снизить производительность других;
- изменить способы сравнения;
- оставить ситуацию без изменения, уйти из организации.

Поскольку связь между текучестью кадров и оплатой труда не редко прослеживается более четко, чем между платой и поведением, обусловленным выполнением дополнительных обязанностей, текучесть зачастую становится единственным предметом внимания руководства. Таким образом, руководство заинтересовано, главным образом, в *удержании на месте ценных работников*. Трудность здесь заключается в разрешении вопросов, связанных с ограничениями творческого потенциала, не позволяющими полностью удовлетворить интересы обеих сторон. В частности, невысокие моральные качества зачастую являются следствием несправедливости.

Психологические характеристики и типы личностей целесообразно оценивать и учитывать при анализе и отборе специалистов для проектирования и производства сложных продуктов. В практической психологии труда и производственных процессов разработаны и исследованы основные характеристики личностей, которые следует оценивать и учитывать руководителям сложных проектов при формировании коллективов и подборе специалистов и сотрудников (см. рис. 1.2) [1, 43, 51].

Интроверсия/экстраверсия отражается в стилях поведения людей, когда они взаимодействуют, участвуют в диалогах или беседах. *Экстраверт* может быть охарактеризован следующим образом: склонность к внешним побуждениям, болтливость, несдержанность, увлеченность людьми и вещами, способность к взаимодействию, активность и склонность к необдуманным действиям. Они стремятся участвовать во всех видах общения, обсуждать любые вопросы и формировать свои суждения благодаря взаимодействию с другими людьми. Их стиль состоит в том, чтобы больше говорить и слушать и лишь иногда размышлять. *Интроверта* характеризуют: склонность к внутренним побуждениям, сдержанность, углубленность, способность выражать идеи и соображения, концентрация, склонность к размышлению и обдуманным действиям. Интроверты предпочитают

общение в письменной форме, благодаря которому они могут обдумывать вопросы, прежде чем их комментировать. Их стиль состоит в том, чтобы больше слушать и размышлять и лишь иногда говорить. В связи с этим задача менеджера проекта состоит в том, чтобы организовать общение как в письменной форме, так и лично, а также предоставить возможность обсуждения и выражения мнений и идей на личном уровне.

Мотивационные различия между экстравертами и интровертами, состоят в том, что у первых преобладает *стремление к вознаграждению*, а у вторых – к *избеганию наказания*. В связи с этим экстраверты лучше выполняют работу, если они ожидают вознаграждения, а интроверты – если им грозит наказание. Поэтому в профессии и предприятии, где в основном работают экстраверты, стимулирование сотрудников целесообразно при регулярном и разнообразном вознаграждении. В бюрократических же организациях, где большинство сотрудников – интроверты, наилучшим способом мотивации является угроза применения санкций.

Сенситивность/интуиция оказывает влияние на все виды информации, которым люди уделяют особое внимание, когда они говорят, слушают или пишут. *Сенситивные личности* стремятся сосредоточиться на конкретных деталях, реальных причинах происходящего. Сенситивная личность характеризуется: умением пользоваться всеми органами чувств, ориентацией на реальность, факты и настоящее, практичность, использование установившихся практических навыков, полезность и последовательность. *Интуитивные личности* стремятся вести общение в более общем контексте и широком смысле. Они получают информацию, главным образом, из внутреннего, или воображаемого мира, устанавливая связь между предметами и идеями, пользуясь своим «шестым чувством» и особенно подходят для мозговых штурмов, чтобы представить дело с разных точек зрения. Интуитивную личность характеризуют: развитое шестое чувство, наитие, склонность оценивать возможные последствия и будущие возможности, теоретизирование, проницательность, стремление получить новый практический опыт, новаторство и непредсказуемость мышления.

Мышление/ощущение связано более непосредственно с решением личности принимать участие в общем деле или самоустраниться. *Мыслящую личность* характеризуют: рассудительность, объективность, справедливость, критицизм, принципиальность, благородство.

зумие, твердость и беспристрастность. Мыслящие личности высоко ценят логику и ясность в общении, стремясь сосредоточить его на работе. Они обособляются, если видят, что таким, не относящимся к делу вещам, как личные эмоции, уделяется слишком много времени. **Ощущающие личности** стремятся придать общению личный характер. Их характеризуют: система ценностей, субъективность, сострадание, любезность, сочувствие и сопереживание. Они самоустраниются при невозможности установления контактов с людьми, либо при напряженных личных отношениях.

Суждение/восприятие оказывает влияние на структуру и стиль общения в коллективе. **Рассудительные личности** стремятся упорядочить всю свою жизнь и действовать, строго придерживаясь намеченных планов. Они уделяют основное внимание: постановке целей, планированию, выработке стратегии и контролю, а также предпочитают организованный и спланированный образ жизни. Рассудительная личность может быть описана следующим образом: упорядоченность, контроль, умение организовать свою жизнь, поставить перед собой цель, решительность и организованность. Они стремятся к тому, чтобы общение как в письменной, так и в устной форме носило целевой и конкретный характер и происходило в следующем порядке: поставить вопрос, завершить его обсуждение, а затем перейти к следующему вопросу. **Восприимчивые личности** склонны к импровизации и поиску разных вариантов. Они уделяют основное внимание: приспособляемости, занятию определенного положения, выявлению ошибок и предпочитают свободный образ жизни. Восприимчивую личность характеризует: открытость, гибкость, склонность к экспериментам, стремление плыть по течению и собирать информацию. Восприимчивые личности склонны к лирическим отступлениям, стремятся предоставить дополнительные сведения, которые могут оказаться полезными, и, как правило, оставляют вопрос открытым.

При общении в производственном коллективе или при проведении совещаний полезно признавать и учитывать индивидуальные отличия специалистов, что способствует продуктивности их делового взаимодействия. Для этого разработана совокупность рекомендаций [1, 51] способствующих повышению эффективности деятельности личностей, различных, **психологических типов**, в частности, при взаимодействии и в процессах совещаний. Для удовлетворения потребности в оценке сотрудников крайне важно дать понять людям,

насколько важна *их роль* в предприятии. Открытое признание их достижений – наиболее простой и эффективный способ удовлетворения этой потребности. Кроме того, люди должны чувствовать, что их работа оплачивается на должном уровне, который определяется их знаниями и опытом. Чтобы удовлетворить потребности персонала в самореализации важно предоставить каждому сотруднику определенный *уровень ответственности за сделанную работу*.

Психологические особенности лидеров производственных коллективов

В каждом проекте программного продукта и его крупном компоненте должен быть лидер – руководитель, менеджер. *Лидер должен иметь талант и квалификацию, а также быть обучен решать следующие задачи:*

- руководить процессом выявления, конкретизации и формирования требований заказчика продукта;
- осуществлять проверку спецификаций требований к программному комплексу, чтобы удостовериться, что они соответствуют реальной концепции заказчика, представленной детальными функциями;
- вести переговоры с заказчиком, пользователями и разработчиками, определять и поддерживать равновесие между тем, чего хочет заказчик, и тем, что может создать коллектив разработчиков за ресурсы и время, выделенные заказчиком для реализации продукта;
- рассматривать конфликтующие пожелания, поступающие от различных участников проекта, и находить компромиссы, необходимые для определения набора функций, представляющих наибольшую ценность для максимального числа участников и проекта в целом (рис. 1.3).

Для крупных комплексов программ целесообразно *формировать иерархию лидеров – руководителей*, адекватно отражающих структуру и взаимодействие базовых функций и компонентов программного продукта [1, 51]. Основному руководителю всего проекта должны быть подчинены руководители его ключевых функций и компонентов, которых следует выбирать с учетом основных психологических свойств и характеристик лидеров соответствующего уровня ответственности. В едином процессе управления крупным проектом

можно выделить несколько **типовых взаимодействующих уровней руководителей**, обеспечивающих реализацию основных целей и политики производства и консолидирующих его корпоративные знания.

Организаторам крупных коллективов целесообразно учиться психологические особенности лидеров производственных коллективов знать и уметь использовать:

- реальный талант и квалификацию лидера для управления и решения производственных задач определенным коллективом;
- уровень размера и сложности задач, доступных для лидера управляющего определенным коллективом;
- методы обсуждения и принятия производственных решений в коллективе – демократический, либеральный или автократический;
- стили руководства и взаимодействия лидера с подчиненными: жесткий, хозяйствский, демократический, либеральный, непоследовательный;
- карьерную ориентацию лидера при управлении коллективом на: интеграцию; независимость специалистов; стабильность коллектива; служение делу, проекту, профессии.

Рис. 1.3

На высшем уровне формируются общие цели, критерии их достижения, а также принципы и схемы взаимодействия с окружением. Формализуются общая причинно-следственная модель деятельности, цели, измеримые способы отображения уровня достижения целей. Ответственность за правильность и полноту представления данного уровня возлагается на руководителей высшего звена.

Установленные цели распределяются по отдельным направлениям деятельности руководителей следующих уровней и согласуются между собой. Проводится планирование мероприятий на уровне всего проекта по достижению целей. Определяются сбалансированные показатели, позволяющие давать количественные и качественные оценки достижения поставленных целей проекта как единой системы, а также определяются конкретные желаемые значения этих показателей. На высшем уровне устанавливаются отношения следующих объектов корпоративного знания: *перспектива (направление) → цель →*

ключевые характеристики цели → связь с другими целями → мероприятия по достижению цели → ответственность.

К типовым задачам управления, которые необходимо решать **руководителю на высшем уровне**, относятся:

- основные цели проекта;
- направления деятельности для реализации целей проекта;
- способ измерения достижения целей;
- факторы, влияющие на достижение целей;
- мероприятия, действующие на достижение каждой цели;
- субъект, отвечающий за достижение цели;
- ресурсы, необходимые для достижения целей.

Уровень руководителей групп и функций поддерживает целевую деятельность проекта, управление логикой совокупности рабочих процессов и структуризацию деятельности. Целевая деятельность и критерии достижения целей декомпозируются для рабочих процессов и подразделений, формулируются и уточняются модели факторов, влияющих на достижение целей. Формируются требования по компетенции и ответственности руководителей. Выделяются фрагменты деятельности групп (отдельные процессы и работы), обеспечивающие достижение установленных целей применительно к функциям проекта, определяются показатели, характеризующие степень приближения к целям. Формализуются ограничения на выполнение выделенных работ и необходимая компетенция руководителей.

Проводится общее согласование границ рабочих процессов групп, их интерфейсов и результатов процессов. Проводится декомпозиция системы сбалансированных показателей, комплексно характеризующих состояние проекта для уровня операционных целей и работ. К типовым задачам руководства и управления, которые необходимо решать **на уровне групп и функций продукта**, относятся:

- какие имеются ограничения, условия и факторы, влияющие на достижение цели, и каково их текущее состояние;
- как обеспечивается реализация функций;
- какие работы выполняются для обеспечения каждой цели и каковы их основные характеристики;
- как распределяются задачи и работы по подразделениям;
- какими ресурсами обеспечиваются рабочие процессы;
- какую степень компетенции по принятию управленческих

решений имеют руководители и ответственные исполнители и како-
вы типовые управленические решения.

Лидеры группы должны быть в курсе повседневной деятельности коллектива, гарантируя эффективную его работу и тесное сотрудничество с заказчиком проекта при планировании деятельности по его реализации. Лидер, как правило, назначаемая должность, он подотчетен главному менеджеру проекта. Такой лидер может и не быть лидером коллектива в прямом смысле этого слова, он ведет группу только в технических вопросах.

Иногда бывает полезным **разделить техническое лидерство и административное управление проектом или компонентом**. Часто случается, что компетентные в технических вопросах специалисты слабо разбираются в администрировании. Исполняя роль администратора, такие лидеры уже не могут вносить полноценный вклад в техническую работу коллектива. В этой ситуации лучше назначить администратора, который освободит лидера от ежедневных административных проблем. Если в группу назначается лидер, который не приемлем для членов команды, это может привести к достаточно напряженной обстановке. Эта проблема весьма существенна для такой быстро изменяющейся области производства, как программная инженерия, где новые члены коллектива могут владеть более современными знаниями и опытом, чем лидеры групп, имеющие только практический опыт. После того как группа будет сформирована, менеджер проекта должен перейти к осуществлению непосредственного руководства и продемонстрировать качества лидера группы. Эффективное управление проектом приводит к пониманию как организационного, так и индивидуального поведения специалистов.

В **обязанности руководителя программного проекта** входит управление, стимулирование и контроль группы разработчиков проекта, а также создаваемого ею программного продукта (см. рис. 1.3). При этом главный менеджер проекта должен выполнять роль технического директора, специалиста по интеграции систем, плановика и ведущего специалиста проекта, ответственного за принятие решений, организатора атмосферы общения и нормального психологического климата в коллективе, посредника и человека, налаживающего взаимоотношения. Для выполнения своих обязанностей менеджер проекта должен обладать умением работать с людьми, практическим опытом работы и осуществления контроля, умением точно представить

позицию предприятия, а также личным стилем руководства. Можно выделить и характеризовать *три основных стиля действий и управления руководителей*: автократический, демократический и либеральный стили [1, 16].

Автократический (авторитарный) стиль подразумевает жесткое руководство: распоряжения, инструкции, требующие от специалистов неукоснительного выполнения. Придерживающийся этого руководитель определяет всю деятельность сотрудников, прописывает все технические приемы и действия. Для этого стиля характерны четкая постановка целей деятельности, энергичная форма распоряжений, приказаний, нетерпимость ко всему, что идет вразрез с мнением руководителя. Члены группы у такого руководителя постоянно находятся в состоянии неопределенности относительно своих будущих действий, знают только свои частные задачи. Связи между участниками сводятся руководителем до минимума, так чтобы они осуществлялись только через его посредничество или при его наблюдении. В похвалах и критике придерживается только своего мнения.

При автократическом решении проблем менеджер определяет и тщательно анализирует проблему, а затем формирует, оценивает и выбирает наиболее подходящее решение. Кроме того, при автократическом подходе коллектив может получить вводную информацию в виде определяемой руководителем проблемы, после чего он использует группу в качестве источника информации для получения сведений, позволяющих определить причину появления проблемы. Используя составленный им список вариантов решения проблемы, руководитель может снова обратиться за информацией к сотрудникам для оценки этих вариантов и окончательного выбора. Следующим шагом привлечения коллектива к принятию решений является сочетание автократического стиля с анализом и ответной реакцией сотрудников, когда менеджер определяет проблему, тщательно анализирует ее причины и выбирает наиболее подходящее решение. В этом случае руководитель представляет коллективу сотрудникам свой план для рассмотрения, анализа и ответной реакции. В *авторитарном стиле можно выделить три варианта*.

Жесткий – руководитель строг, но справедлив, поддерживает уже сложившиеся взаимоотношения в коллективе даже при изменении ситуации; приверженец субординации, хорошо относится к подчиненным, которые «знают свое место». Многие принимают данный

способ руководства, так как оно уменьшает их личную ответственность за неудачу коллектива.

Хозяйский – руководитель навязывает свою позицию отеческой заботы о подчиненных, уделяет внимание повышению их материального благополучия, вмешивается в их дела, даже личные. Преувеличенная защита и опека подчиненных может вызвать у тех ответную агрессивную реакцию. Они будут оказывать серьезное сопротивление руководителю, который, по их словам, не позволяет им «даже дышать». В итоге этот стиль руководства снижает деловую и общественную активность подчиненных и в ряде случаев лишает их умения вести самостоятельную деятельность.

Непоследовательный – неумелый, несобранный, как правило, деспотичный руководитель. Он проявляет видимость могущества, но не уверен в себе, предпочитает выдвигать более слабого, но покорного сотрудника, чем более знающего, но самостоятельного. Мелочно придирчив, готов идти на компромисс, который поможет ему уйти от ответственности в случае неудачи.

При **демократическом стиле** руководитель коллектива в той или иной мере подключается к организации его деятельности. Руководитель разрешает членам коллектива участвовать в определении содержания его деятельности, распределяет между ними ответственность, поощряет и развивает отношения между подчиненными, стремясь уменьшить внутригрупповое напряжение; создает атмосферу товарищества и делового сотрудничества. Все члены группы в общем виде знают все основные этапы деятельности. Когда кто-то просит совета, руководитель предлагает на выбор не один, а несколько способов достижения цели. Руководитель старается быть объективным в своей похвале и критике участников.

Позицию руководителя, придерживающегося демократического стиля, можно охарактеризовать словами «первый среди равных». Его власть необходима для рационального выполнения стоящих перед коллективом задач, но не основывается на приказах и репрессиях. Предоставление подчиненным в ряде случаев самостоятельности и инициативы способствует формированию у них зрелого в социальном плане поведения. Руководитель, взаимодействующий с подчиненными, обращает на это взаимодействие большое внимание. Руководители такого типа не свободны в своих действиях: для них важно, как они выглядят в глазах своих подчиненных. Поведение руководителя,

взаимодействующего с подчиненными, зависит от того, что они думают о нем. Для осуществления своих функций руководители такого типа выявляют потребности своих подчиненных и за исполнение ими своих должностных обязанностей на определенном уровне адекватно вознаграждают их.

При *либеральном стиле* руководитель участвует в организации деятельности группы и вмешивается в нее только в том случае, когда члены этой группы обращаются к нему, причем его вмешательство реализуется в форме совета и объяснения. Членам группыдается полная свобода принимать свои собственные решения. Информация предоставляются им только тогда, когда они просят об этом сами. *Руководитель, взаимодействующий с подчиненными*, обращает на это взаимодействие очень большое внимание. Руководители такого типа не свободны в своих действиях: для них важно, как они выглядят в глазах своих подчиненных. Для осуществления своих функций руководители такого типа выявляют потребности своих подчиненных и за исполнение ими своих должностных обязанностей на определенном уровне адекватно вознаграждают их.

Руководители, влияющие на представления своих подчиненных, более свободны в выборе действий, представления подчиненных их не ограничивают. Такие руководители будоражат воображение подчиненных теми перспективами, которые могут открыться, если будут достигнуты поставленные цели. Они создают образ той корпоративной культуры, которая может быть создана, заражают своей верой подчиненных и стимулируют развитие их собственных способностей; при этом они не отказываются ни от обратной связи, ни от советов. Те из них, кого подчиненные характеризуют как влияющих на них руководителей, более эффективные лидеры. Выявлены три компонента, которые характерны для стиля руководства, основанного на *влиянии на подчиненных*:

- харизма руководителя – уровень доверия к нему и его умения воодушевить подчиненных;
- забота о подчиненных – уровень внимания и поддержки, оказываемых руководителем подчиненным;
- интеллектуальное стимулирование – в какой мере руководители убеждают подчиненных пересмотреть свои взгляды на исполнение ими своих должностных обязанностей.

Еще одну *группу характеристик лидеров* – руководителей рекомендуется оценивать и учитывать как особенности их *карьерной ориентации* [1, 16]. Частота проявления определенных карьерных ориентаций лидеров в коллективах не одинакова, различается по профессиональным признакам, свойствам и характеристикам проектов, для каждой личности существует своя специфика карьерных ориентаций. Оценивание кандидатов в коллективе сотрудников для *формирования их карьеры* состоит в проверке способности работать по определенной специальности, например, *психологические тесты*. Такие тесты направлены на создание психологической карты личности, определяют отношение опрашиваемого к работе определенного типа и способность ее выполнить. Если руководители проектов сталкиваются с трудностями в подборе подходящего персонала с нужными способностями и опытом, они вынуждены составлять команды из неопытных специалистов, что, в свою очередь, вызывает определенные проблемы, связанные с неосведомленностью в области применения продуктов или технологий, используемых в проекте.

Для *руководителя с карьерной ориентацией* первостепенное значение имеет стремление к интеграции усилий других людей. Он готов взять на себя всю полноту ответственности за конечный результат. Человек с такой ориентацией будет считать, что не достиг целей своей карьеры, пока не займет должность, на которой сможет управлять различными сторонами деятельности предприятия: финансами, маркетингом, производством продукции.

Главной заботой людей с *ориентацией на независимость* является освобождение от организационных правил, предписаний и ограничений. У них ярко выражена потребность делать все по-своему, самостоятельно решать, когда, сколько и над чем работать. Такие люди не хотят подчиняться правилам предприятия. Люди данной ориентации могут работать в организации, которая обеспечивает достаточную степень свободы, но не будут испытывать преданности этой организации и станут сопротивляться любым попыткам ограничить их автономию.

Карьерная ориентация на стабильность вызвана потребностью в безопасности и стабильности для того, чтобы жизненные события были предсказуемы. Можно выделить *два ее вида*: стабильность места работы и стабильность места жительства. Первая связана с поиском работы в такой организации, которая обеспечивает определен-

ленный срок службы, имеет хорошую в этом отношении репутацию. Человек с ориентацией на стабильность места жительства связывает себя с географическим регионом и меняет работу только тогда, когда это может предотвратить его «срывание» с места жительства.

Основными ценностями для людей ***карьерного типа – служение*** являются «работа с людьми», «помощь людям», «желание сделать мир лучше». Он откажется от продвижения или вообще не будет работать в организации, которая не соответствует его целям и ценностям.

Главное стремление человека ***с ориентацией вызов – конкуренция, победа*** над другими, преодоление препятствий, решение трудных задач. Человек ориентирован на то, чтобы бросать вызов. Социальная ситуация чаще всего рассматривается с позиции выигрыша/проигрыша. Процесс борьбы и победа для такого руководителя важнее, чем конкретная область деятельности или квалификация. Если деятельность слишком проста, она вызывает у них скуку.

Лидер с ***карьерной ориентацией – предпринимательство*** стремится создавать что-то новое, он хочет преодолевать препятствия и готов к риску. Он не желает работать на других, а хочет иметь собственное дело, свою марку, свой капитал. Он будет продолжать свое дело, даже если вначале потерпит неудачу и ему придется серьезно рисковать.

Лекция 2

ПСИХОЛОГИЧЕСКИЕ ОСОБЕННОСТИ КОЛЛЕКТИВОВ СПЕЦИАЛИСТОВ ДЛЯ ПРОИЗВОДСТВА ПРОГРАММНЫХ ПРОДУКТОВ

Психологические особенности групп и небольших команд специалистов для производства программных продуктов

Основная часть сложных программных продуктов разрабатывается коллективами специалистов (от двух и до нескольких сотен человек). Поскольку невозможно эффективно работать над одной крупной задачей в большой команде, они делятся на группы. Каждая группа отвечает за определенную часть проекта комплекса программ и работает над одной функциональной задачей или компактной системой. В данном разделе рассматриваются особенности относительно небольших по размеру групп и комплексов программ. При грамотном подборе группа специалистов обычно состоит не более чем из восьми человек. В группах небольшого размера легче снизить риск возникновения проблем во взаимоотношениях между ее членами. Для таких групп нет необходимости применять сложные средства коммуникации [11, 42]. **Цели создания группы** состоят в приобретении следующих качеств ее членами: независимости, основания для совместной работы, стремления к коллективному труду, работоспособность, умеренность в отношении соперничества и конфликтов. Кроме того, необходимо добиваться, чтобы каждый член группы придерживался одного и того же общего направления и видел одни и те же намеченные и согласованные цели выделенного компонента или функции комплекса программ (рис.2.1).

Группам – командам должны быть присущи **свойства всех производственных коллективов** [51]. Создание групп представляет собой процесс планируемого и хорошо продуманного стимулирования к эффективному труду при одновременном сведении к минимуму

трудностей и препятствий, мешающих проявлению навыков и изобретательности членов группы. Группы имеют существенное значение для организации эффективного совместного труда разных индивидуумов, причем этот аспект может оказаться самым важным в работе руководителя проекта.

***Руководителям коллективов целесообразно учитывать
психологические особенности групп и небольших команд –
коллективов профессиональных специалистов знать и
уметь использовать:***

- цели и принципы формирования производственных групп – команд;
- особенности и различия методов организации групп – команд специалистов;
- различия структур и способов формирования групп – команд квалифицированных специалистов;
- преимущества и психологические проблемы обеспечения сплоченности команд специалистов;
- особенности и психологические трудности привлечения в стабильные группы исключительно высоких профессионалов;
- психологические особенности принятия решений в сплоченных группах – командах специалистов;
- конфликты в группах специалистов и их урегулирование;
- обеспечение стабильности коллективов – команд специалистов и сокращение текучести кадров.

Рис. 2.1

Создание группы может быть осуществлено практическим способом достижения ясного ощущения членства в группе и единого коллективного духа, причем этот процесс должен быть частью повседневной работы над компонентами проекта. Существуют разные по функциям задачи групп: добровольные группы, собирающиеся для решения конкретных задач повышения качества и снижения затрат, группы с самоуправлением, высокопроизводительные группы, в которых предусмотрено выполнение таких традиционных административных обязанностей, как планирование работ и комплектация штата.

На **эффективность результатов коллективной работы группы** влияют следующие основные факторы [42, 51]:

- команда должна иметь правильное соотношение навыков,

опыта и личностных качеств членов группы;

- члены группы должны воспринимать себя как единую команду, а не как простую совокупность индивидуумов, работающих над одной проблемой;
- между членами команды должны быть дружеские отношения;
- необходимо организовать команду таким образом, чтобы каждый чувствовал свою ценность и был удовлетворен своей ролью в результатах.

Организация команды, которая могла бы эффективно работать над программой, является сложной задачей для руководителя. Необходимо, чтобы в команде было *равное соотношение технических навыков, опыта и выражения индивидуальности*. Хорошо функционирующая команда – нечто большее, чем простой набор людей с необходимым соотношением навыков. В хорошей команде присутствует дух товарищества, который мотивирует сотрудников через успехи всей команды, включая и достижение собственных целей. Поэтому менеджеры должны стимулировать деятельность, направленную непосредственно на **«строительство команды»**, чтобы содействовать формированию чувства преданности ее интересам, которые в той или иной степени влияют на групповую работу.

Наилучший способ воспитать дух команды – дать возможность каждому почувствовать, что он несет *определенную долю ответственности* и что ему доверяют, а также гарантировать доступ к проектной информации для всех членов группы. Иногда менеджерам кажется, что они не должны раскрывать некоторую информацию при разработке. Однако такая линия поведения будет постоянно создавать в группе чувство недоверия. Простой обмен информацией – самый дешевый и эффективный способ дать людям почувствовать себя частью команды.

Если руководители проектов встречаются с трудностями в подборе подходящего персонала с нужными способностями и опытом, они вынуждены составлять команды из неопытных специалистов, что, в свою очередь, вызывает определенные проблемы, связанные с неосведомленностью в области применения конкретных инструментов или технологий, используемых в производстве продуктов. Часто причина такой ситуации кроется в том, что в некоторых организациях технически одаренные сотрудники быстро достигают вершины карьеры. Для дальнейшего продвижения таким сотрудникам необходим

статус менеджера. Если переводить их в категорию менеджеров, для компании это будет означать потерю ценного технически квалифицированного персонала. Чтобы избежать такой ситуации, некоторые компании учредили в своей структуре два параллельных рода деятельности – технический и управленческий, имеющих одинаковую значимость. Опытный технический персонал ценится на таком же уровне, как и менеджеры. С продвижением карьеры специалист может ориентироваться на технический либо управленческий род деятельности и переходить на тот или иной вид деятельности без потери статуса и зарплаты.

Формирование группы – команды, которая могла бы эффективно работать над компонентом или функцией комплекса программ, является достаточно сложной задачей (см. рис. 2.1). Необходимо обучать подчиненных справляться с большим напряжением и поощрять **профессиональную работу во взаимодействии и по графику** [47, 51]. Для обеспечения приверженности подчиненных выполняемой работе следует нанимать людей, цели которых совпадают с целями предприятия, а также людей, высоко ценящих конкретную работу в целом. Ориентироваться следует на работников с реалистичными взглядами на работу в целом, и ни в коем случае не на цели и стремления каждого члена группы в отношении его личной карьеры. Полезно как можно раньше поставить перед сотрудниками интересную задачу и обеспечить условия для их удовлетворенности результатами собственного труда: достижения, независимость, признание, интересные задачи.

Необходимо предоставить неограниченные возможности для общения и ответной реакции. Задания должны быть ясными, подчиненные должны осознавать важность выполняемых ими работ. Рекомендуется использовать оценки и проверки производительности в качестве средства обучения, ставить перед подчиненными трудные и интересные задачи, своевременно реагировать на появляющиеся проблемы, проявлять справедливость и объективность, открытость новым идеям, конструктивно и вовремя реагировать, давать работникам достаточно времени для выполнения порученных работ и заручиться их доверием.

Для создания **благоприятной, здоровой и безопасной рабочей обстановки** целесообразно давать подчиненным разнообразные, интересные задачи, обеспечить социальные гарантии, подбирать со-

трудников, с которыми приятно работать, добиваться положительных откликов заказчиков, заручиться доверием сотрудников, уделять внимание к работе других отделов и рабочих групп. Необходимо также вовлекать сотрудников в процесс принятия решений и поощрять хорошую работу продвижением по службе и предоставлением особых прав, отдельных рабочих мест, помещений и вспомогательного персонала. Для эффективного выполнения проектных работ следует предоставить все необходимые ресурсы и обеспечить безопасность работ. Для поддержания производительности труда на должном уровне рекомендуется разработать хорошую программу оценки и контроля сроков и качества выполненных работ, а также обучения персонала. Следует создать у подчиненных уверенность в том, что администрация готова решать их проблемы, справедливо распределять премии, выдвигать сотрудников только по их деловым качествам, продвигать новые идеи, сохранять лояльность к работникам, предоставлять возможности для роста кадров, а также доступ к информации.

Если менеджеры проекта и предприятия сделают все возможное для достижения указанных выше целей, сотрудники в ответ проявят **ответственность за результаты труда**, уверяясь в его значимости, станут преданными предприятию, используют свои возможности и проникнутся чувством собственной значимости. Когда руководство предпринимает действия по содействию или поддержке желательного поведения, необходимо тесно связать производительность с поощрением, учитывая особенности выполняемого проекта, а также придерживаясь основного правила руководства: **какое поощрение, такое и поведение**.

Группа, в которой **сотрудники дополняют друг друга**, может работать намного эффективнее группы, отбор в которую проводился исключительно на основе некоторых ограниченных личных навыков, например, программирования. Люди, которые любят свою работу, могут стать прекрасными профессионалами. Сотрудники с внешней ориентацией успешно налаживают общение внутри группы. Они настроены на общение и поэтому могут предотвратить возникновение какого-либо напряжения или конфликтов на ранней стадии.

Иногда просто невозможно организовать команду так, чтобы все ее члены взаимно дополняли друг друга по личным качествам. Если такое случается, менеджер проекта обязан контролировать

группу с тем, чтобы не позволить личным целям сотрудников стать выше цели предприятия или группы. Достичь такого контроля намного легче в том случае, если члены команды будут участвовать в каждом этапе проекта. Проявление индивидуальной инициативы будет более вероятным, если с членами группы проводить инструктаж, упоминая о роли, которую их часть работы играет во всем проекте.

В зависимости от типа и размера задачи в группу специалистов могут быть **приглашены высокие профессионалы** в качестве временного или постоянного члена для работы в команде. Это могут быть высококвалифицированный программист, администратор, специалист по инструментальным средствам разработки программ, специалист по операционным системам или языкам программирования, специалист по тестированию программных продуктов или компонентов систем. Обоснованием такого подхода может служить следующее: сколько специалистов по программному продукту, столько различий в способностях создания комплексов программ. Уровень производительности труда (по условной шкале от наилучших до наихудших программистов) может разниться в 25 раз. Поэтому нужно с наибольшей эффективностью использовать возможности лучших работников, обеспечив им оптимальную поддержку. Хотя идея **команды с ведущим программистом** уже более 25 лет, она все еще остается одним из эффективных способов организации небольших групп программистов.

Однако в таких группах **есть проблемы** [51]. Талантливые разработчики компонентов и программисты встречаются нечасто, когда организация группы может быть основана на самом компетентном ведущем программисте и/или его помощнике. Если они совершают ошибки, их решения не с кем обсудить. В демократической группе, наоборот, каждый может обсудить решение и таким образом обнаружить ошибки и избежать их. Ведущий программист ответственен за полное выполнение проекта и может также **взять на себя заслуги в случае успеха**. Однако члены группы могут с этим не согласиться, если их роль в проекте не будет признана в достаточной мере. В таком случае не удовлетворены их потребности в оценке, так как все заслуги будут приписываться ведущему программисту.

Возможен провал проекта в случае болезни или увольнения ведущего программиста и его заместителя. Руководители проектов могут не согласиться на такой риск. Организационная структура

предприятия может оказаться не способной обеспечить подобный тип группы. Большие компании, как правило, имеют хорошо развитую служебную иерархию, поэтому назначение ведущего программиста со стороны может оказаться достаточно трудной задачей. В компаниях небольшого размера практически невозможно выделить одного сотрудника на выполнение всего задания. Поэтому **структура групп с ведущим программистом** может оказаться для предприятия весьма **рискованной**. Однако необходимо поддерживать талантливых программистов, выделяя для них помощников, администраторов и т.д. Таким образом, можно использовать способности одаренных сотрудников наиболее эффективно. Назначение узкоспециализированного специалиста на короткие промежутки времени в отдельные группы разработчиков может быть эффективнее, чем использование программиста с большим опытом на протяжении долгого периода времени в работе над одним проектом.

Члены **сплоченной команды** должны быть привержены ее интересам больше, чем своим собственным [42, 47]. У них высоко развит командный стиль поведения и отождествление целей группы с личными интересами. Это укрепляет группу, она становится способной самостоятельно справляться с проблемами и непредвиденными ситуациями. Члены группы поддерживают друг друга в процессе различных изменений, что помогает им преодолевать трудности. Хорошо сплоченная команда имеет **ряд преимуществ**:

- возможность установления стандарта действий членов группы, который определяется всей группой единогласно, его легче контролировать, чем чужие стандарты, навязываемые группе извне;
- члены команды поддерживают тесные рабочие контакты, люди учатся друг у друга, уменьшаются затягивания работы, вызванные незнанием или неосведомленностью, по мере того, как происходит взаимное обучение;
- члены команды ознакомлены с деятельностью друг друга, этим достигается возможность продолжения работы даже после ухода из команды одного из сотрудников;
- возможно внедрение в практику группы безличного программирования, когда программный продукт должен быть собственностью всей команды, а не отдельной личности.

Безличное программирование [42, 51] означает определенный стиль работы, при котором продукты, компоненты программы и до-

кументация считаются собственностью всей группы, а не отдельного человека, который занимался их разработкой. Если разработчики психологически воспринимают свою работу именно таким образом, они охотнее отдают ее на проверку другим членам группы, легче воспринимают критику, стремятся работать над усовершенствованием системы. При этом команда разработчиков становится более сплоченной, так как каждый чувствует себя в ответе за разработку всей системы. Кроме того, что безличное программирование улучшает качество системной архитектуры, программ и документации, оно также **совершенствует взаимоотношения внутри группы**, стимулирует непринужденное обсуждение задач, независимо от социального положения, опыта и пола сотрудников. В ходе выполнения проекта члены группы более активно поддерживают рабочие отношения между собой. Все эти факторы содействуют сплоченности и тому, что каждый работник воспринимает себя как часть команды.

Многие факторы могут оказывать влияние на **сплоченность группы**, в том числе организационная культура и выражение личностных качеств в группе. Менеджеры могут развивать сплоченность несколькими путями. Можно организовывать социальные мероприятия для работников и их семей. Можно привить группе чувство самобытности, для чего ее надо назвать, определить сущность команды и сферу ее деятельности. В сильных и сплоченных группах **могут возникнуть психологические проблемы** (см. рис. 2.1):

- если лидера сплоченной группы требуется заменить кем-то со стороны, члены группы могут сообща выступить против нового лидера, поэтому по возможности следует назначать нового лидера из имеющегося состава группы;
- ценные рабочие качества членов группы могут разрушаться в результате преданности команде, когда любая идея принимается, если большинству это выгодно, альтернативы при этом не рассматриваются.

Чтобы избежать **проблемы групповой солидарности**, нужно организовывать официальные совещания, где члены команды будут вынуждены так или иначе высказывать собственные мысли. Команда должна состоять из людей, которые способны дискуссионировать, задавать вопросы и не обращать особого внимания на устоявшееся положение. Для группы по разработке программных продуктов необходим **развитый коммуникационный фактор**, общение и хорошие средст-

ва связи между членами группы. Работники должны информировать друг друга о том, как идет их работа, о решениях, которые предпринимались в отношении проекта, и тех необходимых изменениях, которые вносились в предыдущие версии. Постоянное общение также способствует сплоченности, поскольку работники лучше начинают понимать мотивацию своих коллег, их слабые и сильные стороны. На **эффективность общения** могут оказывать влияние следующие факторы:

- **чем больше группа**, тем труднее обеспечить постоянное общение между ее членами, в группе из 7 – 8 человек могут быть люди с низким показателем связей, так как различие в социальном положении членов группы приводит к появлению большего количества односторонних связей;
- члены группы с более высоким положением в обществе, чаще доминируют в общении со своими коллегами, которые неохотно идут на контакт или высказывают критические соображения;
- специалисты, состоящие *в группах с неформальной структурой*, легче общаются между собой, чем в группах, которые имеют определенную официальную иерархию в отношениях, когда сотрудники на одной и той же иерархической ступеньке могут не общаться между собой, что в проектах крупных программных продуктов, включающих несколько групп, может приводить к запаздыванию решений;
- если в группе много людей с *похожими личностными характеристиками*, они могут конфликтовать друг с другом, вследствие чего может значительно снижаться уровень общения.

Небольшие команды специалистов лучше всего организовывать *в неформальном духе*, когда лидер участвует в работе над программным продуктом наравне с другими членами группы. Техническим лидером может стать человек, который лучше всего будет управлять процессом разработки. Неформальная группа всегда обсуждает предстоящую работу со всеми членами коллектива, а задания назначаются в соответствии с возможностями и опытом конкретного сотрудника. Высокоуровневое проектирование системы может осуществляться старшими специалистами, проектирование низкого уровня предоставляемое тому сотруднику, который назначен на выполнение конкретного задания. Работа неформальных групп может оказаться эффективной, особенно если большинство членов группы опытные и

квалифицированные специалисты. Когда группа функционирует в качестве **демократического коллектива**, где решения принимаются большинством голосов, в психологическом плане это порождает дух команды и, следовательно, приводит к укреплению сплоченности и повышению производительности труда. Наиболее независимой организацией группы является **самоуправляемая единица**, в которой отсутствует формальный руководитель, а группа организована на принципах самоуправления.

Групповое принятие решений необходимо для сплочения группы, особенно в том случае, если она состоит из подгрупп, специализирующихся на выполнении конкретных функций. По мере увеличения группы повышается вероятность появления большего количества заинтересованных сторон, причем их интересы могут быть учтены в критериях, применяемых в процессе принятия решений. Вовлечение отдельных лиц с особым практическим опытом в процесс принятия решений позволяет повысить вероятность того, что в данном процессе будут использованы более точные причинно-следственные допущения. Ведь группы стремятся выработать большее число возможных и более творческих вариантов. Отдельные члены группы, принимающие участие в выработке решений, ощущают большую принадлежность к решениям, особенно в том случае, если это касается их лично.

Первым шагом для руководителя группы является выбор собственного **стиля принятия решений**, который может изменяться в зависимости от конкретных обстоятельств. Этот стиль может быть любым: от автократического до полного участия. Если руководитель склоняется к некоторой форме группового принятия решений, в этом случае важно назначить кого-либо из участников группы, кто будет руководить или способствовать упрощению данного процесса. Этот человек должен взять на себя ответственность направлять группу на протяжении всего процесса принятия решений. Руководитель должен выяснить для себя, достаточно ли у него квалификации для выполнения этой роли, чтобы в случае необходимости поручить ее кому-нибудь другому. Иногда кандидат на эту роль появляется в ходе самого процесса принятия решений. При этом очень важно, чтобы руководитель явным образом передал полномочия по управлению процессом принятия решений, чтобы члены группы не оставались в неведении относительно того, кто отвечает за данный процесс.

Когда решения принимаются в группе, участники стремятся прийти к варианту, который удовлетворял бы каждого члена группы. Ожидать соответствия принимаемого решения общей цели или интересам группы при этом не приходится. Главное, чтобы решение соответствовало интересам членов, представляющих заинтересованные стороны. Но при групповом принятии решений всегда существует главная цель группы независимо от целей и интересов отдельных ее членов. Если группа работает эффективно, деятельность каждого ее члена направлена на объединение всех целей группы в одну и подчинению собственных интересов достижению этой групповой цели.

Отдельный участник усваивает групповые цели как собственные. Если группа действует успешно, отдельный участник видит в этом некоторую личную выгоду, реализуемую, в частности, посредством премирования за достижение групповых целей. Если отдельное лицо видит, что группа действует успешно, в его представлении эта группа соответствует эталонной группе, члены которой будут приписывать успех группы его компетенции. Таким образом, отдельный участник видит в успехе группы подтверждение или подкрепление его внутренней самооценки собственных способностей. Поэтому если группа действует успешно, этот успех является подтверждением компетентности отдельного лица. Для этого требуется, чтобы отдельный участник понимал, что выполняемая им роль оказывает существенное влияние на успешную деятельность всей группы.

Менеджеры проектов и их команды могут пользоваться самыми разными **стилями принятия решений**. При автократическом или направляющем стиле решения проблем менеджер определяет и тщательно анализирует проблему, а затем формирует, оценивает и выбирает наиболее подходящее решение. Кроме того, при автократическом подходе команда может получить вводную информацию в виде определяемой руководителем решения проблемы, после чего он использует группу в качестве источника информации для получения сведений, позволяющих определить причину появления проблемы. Используя составленный им список вариантов решения проблемы, руководитель может снова обратиться за информацией к группе для оценки этих вариантов и окончательного выбора.

При использовании **стиля отдельных консультаций** менеджер определяет проблему и делится своей точкой зрения по той или иной проблеме с отдельными членами рабочей группы. При этом он инте-

ресурсется их мнением по поводу причин появления проблемы и возможных ее решений. Кроме того, руководитель может воспользоваться опытом отдельных членов группы для оценки альтернативных решений. Получив подобную информацию, руководитель делает окончательный выбор.

Стиль групповых консультаций дополняет стиль отдельных консультаций в том отношении, что руководитель делится своим пониманием сути проблемы со всей группой. В частности, менеджер проекта может воспользоваться групповым стилем, чтобы поделиться своим мнением по проблеме с рабочей группой, после чего члены группы тщательно анализируют причину ее появления. В результате этого анализа в группе формируются альтернативные решения, затем производится их оценка и окончательный выбор. При использовании стиля участия, вся группа задействована в процессе принятия решений.

Организация рабочего места имеет важное значение для производительности труда и получения удовлетворения от работы, в развитии или торможении коммуникационных связей в группе. Психологические исследования доказали влияние на поведение сотрудников размера комнаты, мебели, оборудования, температуры, влажности, яркости и качества света, уровня шума и возможности уединения. Поведение членов группы разработчиков также зависит от архитектурных особенностей помещения и возможностей доступных средств связи. Руководителю дорого обойдется неумение обеспечить хорошие рабочие условия для сотрудников.

Конфликты являются обычным фактом жизни трудового коллектива. Если их нельзя избежать, значит можно урегулировать. Поскольку руководителям программных проектов неизменно приходится иметь дело с конфликтами, они должны научиться их распознавать и справляться с ними самыми разными способами и умением вести переговоры. Каждая сторона, участвующая в конфликте, несет ответственность за ясное изложение своих целей и выяснение у другой стороны ее целей. Конфликты существенно отличаются от нормального общения своей интенсивностью. Они могут мешать общению: препятствовать ему, изменять его стиль, оказывать влияние на то, как мы слушаем, а также ограничивать услышанное нами. В конфликтах, как в общении и на совещании, проявляются предпочтения типов личности.

Конфликты между сотрудниками определяются как нарушение процесса принятия решений, когда невозможно выбрать альтернативный вариант, а конфликт эмоциональный проявляется в разногласиях сторон по альтернативному варианту или по позициям [51]. Причины конфликтов коренятся в следующих *двух типах разногласий*: либо критерии, интересы и цели, либо причинно-следственные убеждения, теории и допущения. Если существующий конфликт по поводу критериев, интересов и целей может быть разрешен в процессе принятия решений, то эмоциональный конфликт по поводу убеждений, теорий и допущений может оставаться довольно долго. **Эмоциональный конфликт** приводит к тому, что другая сторона начинает в своих действиях руководствоваться личными эмоциями, которые зачастую еще долго сказываются после окончательного их разрешения.

Методы урегулирования конфликтов, возможные для руководителей программных проектов, требуют [51]:

- затрачивать дополнительное время и энергию для получения взаимовыгодных результатов;
- энергично приступать к решению проблемы, но не прибегать к нападкам на личном уровне;
- внимательно выслушивать другую сторону: слушать, понимать, признавать, ценить и предпринимать ответные действия;
- стремиться к прямому, открытому и доверительному общению;
- вмешиваться только по мере необходимости во время обсуждения проблемы с третьей стороной: слушать, наставлять или советовать, упрощать процесс обсуждения, договариваться, выступать в качестве посредника;
- находить вполне нормальным явлением наличие конфликтов; считать, что конфликт может быть полезным, важным, ценным и выгодным; в большинстве случаев скрывать проблему или конфликт нецелесообразно.

Каждая сторона, участвующая в конфликте, **несет ответственность** за ясное изложение своих целей и выяснение у другой стороны ее целей. К навыкам и методам общения, которые могут оказаться полезными для руководителя проекта в конфликтных ситуациях, относятся следующие:

- «допущение» гнева и умение соответствовать гневному пове-

дению;

- выяснение требований другой стороны;
- признание правоты другой стороны;
- сообразность, честность и откровенность без всяких нападок;
- умение слышать;
- способность мыслить творчески.

Текущесть кадров обходится недешево, ибо при этом увеличиваются расходы на расчет увольняемого работника, собеседование с ним и связанные с этим административные мероприятия. Кроме того, текучесть кадров повышает расходы на поиск замены уволившегося работника, в том числе на объявления, наем и собеседование с новыми кандидатами, обучение и ввод в курс дела. Перед увольнением производительность труда работника, как правило, падает, а вместе с ней падает и производительность остальных членов группы.

Эффективно работающий руководитель проекта должен **знать цену стимулированию**, способствующему желанию группы работать много и стремиться к достижению поставленных целей посредством удовлетворенности от труда, под которой подразумевается связь между сотрудником и окружающей его рабочей обстановкой, а также взаимодействие между ними. Те менеджеры проектов, которые постоянно заботятся о стимулировании, удовлетворенности и преданности подчиненных, понимают, что при этом улучшается также моральный климат во взаимоотношениях руководителей и подчиненных. Одновременно уменьшается число чрезвычайных случаев, повышается эффективность и производительность труда, у сотрудников появляется больше стимулов для продвижения по службе и повышения квалификации, становится меньше конфликтов, а также **уменьшается текучесть кадров и ее отрицательные последствия**.

Расторжение договора о найме – явление столь же частое, как прием на работу. С правовой точки зрения, можно выделить несколько способов расторжения трудового соглашения:

- расторжение трудового соглашения на основании предварительного заявления о расторжении со стороны работника;
- расторжение трудового соглашения на основании нарушения сторонами условий контракта или изменения условий работы;
- расторжение трудового соглашения с момента истечения срока, на который оно было заключено.

Как правило, работодатель заранее точно определяет мотивы, которые могут обусловить форму увольнения. С этой точки зрения увольнение может рассматриваться как одно из наиболее серьезных средств, находящихся в распоряжении руководителя, которое может быть использовано как самое последнее средство. Эту ситуацию можно трактовать как свидетельство признания неудачи воспитательного процесса, организуемого каждым руководителем в отношении своих подчиненных.

Жизненный цикл развития производственного коллектива программных продуктов

Для каждого коллектива, предназначенного для проектирования и производства *сложных программных продуктов*, характерны определенные *процессы развития*, которые целесообразно учитывать менеджерам. При этом большое значение имеют *психологические характеристики специалистов*, их квалификация и зрелость всего коллектива. Обычно основные интересы коллектива специалистов сводятся к [1, 16]:

- обеспечению постоянства работы;
- нормальным условиям и привлекательности процессов производств труда;
- получению максимально высокого заработка за выполненную работу;
- возможностям квалификационного и должностного продвижения;
- социальным гарантиям и возможностям, предоставляемым предприятием своим работникам;
- нормальному социально-психологическому климату в коллективе.

Формирование и развитие трудовых коллективов проходит ряд этапов, для каждого из которых характерны свои особенности (рис. 2.2).

Первым является этап *проектирования трудового коллектива*. К этому времени должен быть проведен выбор техники и технологии, определена структура предприятия. Исходя из общих характеристик программного продукта, запланированного объема работы в подразделении, на основе расчетов необходимой общей трудоемкости опре-

деляется потребная численность специалистов, оценивается число вспомогательных и обслуживающих работников, устанавливается структура управления подразделением. Выявляются общие требования профессионального, квалификационного и личностного, в частности, психофизиологического плана к претендентам на занятие вакантных должностей.

Руководителям коллективов целесообразно учитывать жизненный цикл коллективов специалистов для производства программных продуктов, знать и уметь использовать:

- организацию, формирование и развитие жизненного цикла профессиональных коллективов специалистов;
- методы сохранения стабильности и благоприятного социально-психологический климата в производственных коллективах;
- культуру взаимодействия и социального поведения специалистов в профессиональных коллективах;
- оценки характера личностей в профессиональных коллективах;
- совещания, стиль переговоров, принятия решений в профессиональных коллективах и учет индивидуальных особенностей специалистов.
- психологические особенности процессов взаимодействия обучающихся с преподавателями;
- компетентность и желательные психологические свойства руководителя – лидера для управления конкретным крупным коллективом специалистов.

Рис. 2.2

Второй этап – процесс организационного формирования трудового коллектива. На этом этапе осуществляется подбор работников в соответствии с проведенными ранее расчетами потребности, прием на работу, уточняется распределение труда внутри подразделения, что закрепляется в положении о подразделении и должностных инструкциях. Рассматриваются практические вопросы психологической совместимости возможных работников, взаимоотношений между руководителем и его подчиненными.

Третий этап – первичное совместное функционирование и адаптация специалистов в коллективе. Происходит практическая «притирка» членов коллектива друг к другу, совершенствуется организация рабочих мест, углубляется процесс разделения сфер труда,

развиваются профессиональные навыки, выявляются неформальные связи и лидеры, формируется ядро коллектива. Коллектив расстается с теми, кто не смог адаптироваться ко всем этим факторам, а на их место приходят другие претенденты. Следующие *две стадии* относятся к уже сложившемуся, зрелому коллективу.

Первая из них – *период продуктивного функционирования коллектива*, когда основными задачами является выполнение наилучшим образом возложенных на него функций. Должно быть обеспечено реагирование на изменения в технике и технологии, повышение квалификации специалистов, совершенствование сплоченности коллектива, контроль за соблюдением нормального психологического климата в коллективе, контроль за стабильностью коллектива.

Заключительная стадия – *стагнация*. Со временем в коллективе начинают накапливаться отрицательные качества. Часть членов в коллективе сохраняется без перемещения на одном и том же рабочем месте, ядро коллектива во главе с руководителем может ограничивать устремления и инициативу молодежи. Научно-технический прогресс может предъявить к коллективу повышенные требования, которым последний не будет удовлетворять, может значительно возрасти средний возраст коллектива, и в связи с этим упасть производительность труда. На этом этапе следует предпринять целый ряд действий, которые дадут возможность не довести дело до краха коллектива. Это изменение структуры организации, ее разукрупнение, внедрение новой техники и технологии, перестановка управленческих кадров, перетасовка сотрудников коллектива.

Комплекс организационных мер, соответствующих каждому из этих этапов развития трудовых коллективов, составляют управленческие воздействия на процессы их формирования и функционирования. При этом надо стремиться к *сохранению стабильности коллектива*. Этими процессами надо *управлять методами*, к которым относятся:

- регулирование процессов перемещения, продвижения и понижения работников;
- оценка результатов работ и специалистов;
- профессиональная подготовка и повышение квалификации специалистов;
- поддержание нормального социально-психологического климата в коллективе;

- управление мотивацией и стимулированием специалистов;
- выбор оптимальной системы материального и морального стимулирования специалистов.

Большую роль в деле *стабилизации* коллектива играет *социально-психологический климат*. Это психологическое состояние коллектива, характер ценностных ориентаций, межличностных отношений и взаимных ожиданий в нем. Оно зависит от среды и уровня развития коллектива и непосредственно влияет на эффективность деятельности его членов, на осуществление основных его функций. Благоприятным является климат такого коллектива, ценности и отношения в котором отвечают *задачам развития коллектива*: у членов достаточно развита потребность в труде как сфере актуализации личности; в межличностных отношениях развиты взаимное доверие и уважение друг к другу, взаимная информированность по значимым вопросам, взаимовыручка и взаимная ответственность. Руководитель обязан обеспечивать формирование благоприятного климата, используя методы социального управления, преодоление конфликтов и развитие сплоченности коллектива: комплектование коллектива с учетом психологической совместимости его членов.

Культура взаимодействия в коллективе представляет собой характерную совокупность убеждений людей, связанных общим делом и ценностями, которыми они руководствуются в своей жизни. К элементам культуры относятся образование и убеждения, социальная организация и политическое воспитание в широком понимании общения. [43, 51]. При анализе и организации производства программного продукта необходимо рассматривать два комплекса проблем – первый, в основном, охватывает социальные проблемы, второй – технические, и оба совместно – экономические. *Социальное поведение* отдельного лица в коллективе тесно связано с его особенностями. Если отдельное лицо видит, что группа действует успешно, в его представлении эта группа соответствует эталонной группе, члены которой будут приписывать успех группы его компетенции. Для этого требуется, чтобы отдельный участник понимал, что выполняемая им роль оказывает существенное влияние на успешную деятельность всей группы.

Применительно к целям и стандартам работ разрабатываются *стандарты деловых и личностных качеств работников*, претендующих на занятие определенной должности, и проводится сопос-

тавление характеристик конкретных оцениваемых сотрудников с этими стандартами. Оценивается то, как конкретные работники справляются с порученной им работой. В ходе проведения этого анализа результаты отдельных этапов не обязательно должны соответствовать друг другу. По результатам оценки необходим комплекс мер по исправлению выявленных недостатков. На этой стадии используются результаты оценки для получения *обратной связи* от существующей организации и планирования новых направлений деятельности или от специалиста – исправление или улучшение отдельных сторон его профессионального поведения.

Важнейшим элементом успешно функционирующих систем управления сотрудниками является *использование результатов оценки*. Есть разнообразные пути того, как руководители могут применить эти результаты на практике. Использование результатов должно включать:

- развитие имеющихся способностей и исполнительных качеств сотрудников;
- оценивание степени использования специалистом своих потенциальных возможностей и решение вопроса о целесообразности должностного повышения сотрудника;
- проведение регулирования размера и формы его зарплаты;
- помочь при решении вопроса о должностных перемещениях;
- увольнение или вынесение окончательного заключения по должностной карьере специалиста.

Первоочередное использование оценочных результатов нацелено на *развитие текущих, относящихся к выполняемой работе* деловых качеств сотрудников. Такая постановка целей показывает, что качественность результатов зависит от непосредственного начальника и от подчиненного. Однако некоторые руководители не любят сообщать результаты своих оценок подчиненным, считая, что последние и так автоматически знают свое место в организации. Профессионально грамотные начальники, тем не менее, используют результаты оценок, чтобы помочь подчиненным *повысить их производственную отдачу*. Это требует специальной подготовки и необходимых знаний в области обеспечения технологии обратной связи и умения осуществлять профессиональное консультирование.

Оценка потенциала возможностей для отбора кандидатов на продвижение менее надежно, чем использование их для совершенст-

вования процесса. Оценка не дает гарантий будущего успешного выполнения работы человеком, особенно если трудовая деятельность специалиста, который может быть представлен к продвижению, будет существенно отличаться от той работы, которую он выполняет в настоящее время. Такие факторы, не учитываемые методикой оценки, также могут значительно повлиять на возможности его успеха в новой обстановке, при этом большинство руководителей, обсуждая кандидата на продвижение, стремятся строить свое заключение на базе рассмотрения оценочных результатов, полученных ранее.

Важная стратегическая проблема в использовании оценок лежит в их *связи с зарплатой и, конкретно, с ее повышением*. Связывание между собой решений по зарплате с оценочными результатами специалиста обеспечивает автократическому руководителю получение единственного инструментария в дополнение к другим источникам его власти. Оценки обеспечивают полезную дополнительную информацию для решений о приостановке деятельности, *снижении в должности, перемещении специалиста или о его увольнении*. Хотя такая информация обычно не является решающим элементом в принятии решения о целесообразности оставления сотрудника в должности или о переводе его на другое место. Развитие индивидуальных способностей и точное определение источников формирования потенциала для продвижения хорошо служат там, где проводятся систематические и периодические оценки сотрудников. Они также могут быть использованы при отборе кандидатов для открывающихся вакансий, для поощрения саморазвития и для успешного руководства процессом формирования карьеры.

Иногда *отдельные лица оказывают большое влияние* на других лиц, группы работников, организацию в целом и на окружающую обстановку. Те менеджеры проектов, которые постоянно заботятся о стимулировании, удовлетворенности и преданности подчиненных, могут обнаружить, что при этом улучшается также моральный климат во взаимоотношениях руководителей и подчиненных. Кроме того, уменьшается число чрезвычайных случаев, повышается эффективность и производительность труда, у работников появляется больше стимулов для продвижения по службе и повышения квалификации, становится меньше конфликтов, а также уменьшаются отрицательные последствия текучести кадров.

Принятие производственных решений в коллективах зачастую рассматривается как единый процесс, реализуемый посредством **совещаний** [1, 51], которые обычно включают: обсуждение, компромисс, исследование вопроса, аргументацию, конфликт, голосование, сравнение, предложения о пересмотре. У **каждого совещания** должна быть **повестка дня**, оглашенная до его начала. В ней указывается время и место, участники совещания и распределение их ролей, подготовительная работа к совещанию, рассматриваемые темы, регламент, предлагаемый процесс рассмотрения вопросов с учетом времени.

Председатель совещания или направляющий данный процесс, желательно, выбирается из незаинтересованных специалистов, не относящихся также к руководству и не наделенных властью. Он направляет совещание в нужное русло, руководит всем процессом, делает содержание совещания доступным каждому его участнику, подводит итог обсуждению, соблюдает его повестку дня и следит за регламентом. Присутствие этого лица необходимо на тех совещаниях, где происходит обмен профессиональной и производственной информацией, решаются проблемы, принимаются решения, составляются планы или делаются оценки. По возможности, целесообразно назначать секретаря, ведущего протокол, объективно регистрирующего все высказанные на совещании соображения, систематизирующего информацию и выполняющего ряд других полезных вспомогательных функций.

Участниками совещания могут быть специалисты, обладающие необходимым опытом и квалификацией для внесения предложений, выражения идей, предложения вариантов, высказывания догадок, проявления активности и опытности. Они работают на заказчика или руководителя проекта, являются главным резервом совещания и выбираются с учетом их возможного вклада в решение вопросов, рассматриваемых на совещании. Они играют активную и ответственную роль на тех совещаниях, где происходит обмен информацией, решаются проблемы, принимаются решения, составляются планы или делаются оценки проектирования и производства программного продукта.

Принятие решений зачастую рассматривается как единый процесс. Он обычно включает следующее:

- обсуждение, конкретное или беспорядочное;

- согласие, компромисс, взаимные уступки;
- исследование вопроса, аргументация, голосование, сравнение;
- анализ, исследование и изучение.

В качестве **результатов совещаний** могут служить протоколы, в которых зафиксированы обсуждавшиеся вопросы, принятые решения и пункты повестки дня, а также назначенные действия – **кто, что и когда** должен предпринять для выполнения каждого назначенного действия, хотя конкретный срок иногда опускается, если назначенный исполнитель способен самостоятельно оценить объем предстоящей работы.

Стратегические или плановые совещания призваны определить перспективное направление развития предприятия либо его подразделения – группы. Как правило, на таких совещаниях рассматриваются вопросы соответствия поставленной цели и текущих стратегий ее достижения, а, следовательно, дается оценка текущего направления развития. Если в итоге обнаруживается, что перемены в окружающей обстановке привели к расхождению какой-либо цели и стратегии ее достижения, вырабатывается новый стратегический план. При появлении конкретной проблемы, которая обычно проявляется в виде некоторой реакции неудовлетворенной заинтересованной стороны, собирается совещание, посвященное решению данной проблемы. Такие совещания проводятся в соответствии с одной из следующих двух моделей.

Решение **насущной проблемы** – основное внимание на таком совещании уделяется тому, как удовлетворить оперативно текущие потребности неудовлетворенной заинтересованной стороны. Так, если заказчик получил версию программы с серьезными ошибками, может возникнуть следующий вопрос: как исправить ошибки в программном продукте, избежав появления новых проблем, и быстро предоставить заказчику очередную версию.

Решение **отдаленной проблемы** – основное внимание на таком совещании уделяется прогнозированию и снижению вероятности появления подобного рода проблемы в отдаленной перспективе путем тщательного анализа причин ее повторного появления и выработки подходящего решения данной проблемы, исходя из этих причин. Подобная проблема может быть связана с повышением эффективности проводимых проверок, улучшением процедуры классификации ошиб-

бок, созданием более информативных отчетов и ознакомление с эти-ми отчетами нужных лиц. Если при проведении совещаний призна-ются и учитываются индивидуальные отличия специалистов, такие совещания оказываются более продуктивными (см. лекцию 1) .

Умение вести переговоры относится к необходимым, главным качествам **любого** руководителя, для выполнения своих обязанностей [1, 43, 51]. **Переговоры** – это процесс выхода двух людей из затруд-нительного или конфликтного положения и создание ситуации, кото-рая способствует выработке решения и согласию. При этом две или более сторон с общими или противоположными интересами испыты-вают потребность или имеют поручение достичь согласия относи-тельно разделения или распределения ресурсов или устранения тех-нических разногласий. К методам ведения переговоров относятся планирование, проведение и достижение согласия, критика и принятие последующих мер. Это процесс внесения прямых и ответных предложений, обсуждения и уступки, благодаря чему обе стороны достигают взаимопонимания относительно наилучшего результата, которого им удалось добиться. К основным **принципам ведения пе-реговоров** относятся следующие:

- отделение личных мотивов от решаемых вопросов;
- сосредоточение основного внимания на общих интересах;
- выработка возможных решений, отражающих общие интересы;
- достижение результатов, исходя из целей проекта и критери-ев его успешного выполнения.

Торг не считается переговорами и в рабочей обстановке не-уместен, если в этом нет абсолютной необходимости, которая иногда все же возникает [51]. Торг начинается в том случае, когда возникает соперничество, вступают в конфликт мировоззрения или системы ценностей, применение силы или угрозы становится нормой, взаимо-отношения носят короткий и формальный характер, предполагаются или оказываются уместными споры либо возникает безвыходное по-ложение. Существуют **две основных позиции** при ведении торга: превосходство, когда сила на данной стороне, а другая сторона ока-зывается в подчиненном положении и при этом возникает резкая критика, а также подчинение, когда сила на другой стороне, а лесть и воздержание от споров считаются приемлемыми методами ведения торга.

Целесообразно при обучении специалистов учитывать **психологические особенности процессов взаимодействия обучающихся с преподавателями** [36]. Многие учащиеся воспринимают слова преподавателя как нечто, не подлежащее сомнению; привыкли к тому, что их решения оцениваются извне, практически не умеют задавать вопросы; осознанно формулировать, что же ими не понято, а сознаться в непонимании зачастую считают позором. В связи с этим необходима психологическая подготовка учащихся, которая кратко отражается следующими тезисами.

Любое научное знание односторонне, является результатом самоограничения. Это самоограничение должно стать сознательным, а возможные альтернативы – осознаваться, а не отвергаться. Нужно стараться открывать дыры и недостатки даже в самых устоявшихся областях знания, уметь показывать альтернативы.

В идеале преподаватель должен помогать учащемуся выбирать наиболее подходящее решение. Одно из их важнейших различий творческого и рутинного мышления – **отношение к ошибкам**. Творчески мыслящий человек рассматривает ошибку как закономерный этап на пути к решению задачи, а рутинно мыслящий – как огрех. Первый стремится превратить выявленную и понятую им ошибку в элемент решения для того, чтобы потом сделать наоборот, а второй не признает ошибку до последнего момента и когда все же вынужден признать ее, пытается отбросить все сделанное и начать сначала.

Ошибка – неизбежный шаг на пути к решению трудной задачи. Цель обучения – научить учащихся здравому отношению к ошибкам и превращению их в правильное решение. Для развития критического мышления учащихся требуется их научить критическому отношению к словам преподавателя, поэтому иногда на занятиях стоит допускать намеренные ошибки либо недостаточно обоснованные утверждения, а затем поощрять тех, кто их заметит. Хороший **психологический прием** – иногда реагировать на правильные ответы, как на неправильные, требуя дополнительных объяснений, либо всячески вселяя в учащихся сомнение.

Полезно, вырабатывая у обучающихся здоровое отношение к ошибкам, демонстрировать его на собственном примере. Рекомендуются [36] некоторые провокационные приемы, чтобы разрушить стереотип безошибочного продвижения к истине и выработать навыки ответственного критического подхода к решениям. Как известно, быстрая понимания учащимся не означает его глубины, хотя и не противоре-

чит ей, поэтому необходимо поощрять коллективную работу группы. Если один что-то не понял – полезно сначала спросить товарища, а уже затем обращайся к преподавателю.

Целесообразно поощрять коллективную работу в маленьких группах и взаимообмен идеями и решениями по инициативе учащихся. В какой-то мере этот принцип связан еще и с **особенностями национальной психологии** учащихся. В российских вузах нет ни американского духа глубокого индивидуализма и безжалостной конкуренции, ни английской щепетильности в вопросах чести к своим знаниям. Поэтому подсказки и списывание здесь неискоренимы, их лучше легализовать, обратив на пользу обучению.

Человек с **ориентацией на достижение мастерства** в своей области, удовлетворен своей деятельностью, если достигает в ней успеха, но быстро теряет к ней интерес, если работа не позволяет ему реализовать и развивать имеющиеся способности. Такой человек ищет признания своих способностей и требует статуса, соответствующего его таланту и квалификации. Управление другими людьми может не представлять для него интереса, хотя он может это делать в пределах своей компетентности. Поэтому многие из этой категории людей **отвергают работу руководителя**, а управление рассматривают лишь как необходимое условие для продвижения в своей профессиональной карьере. Это самая многочисленная группа сотрудников в большинстве предприятий, обеспечивающая принятие компетентных решений.

Во всякой иерархии каждый сотрудник имеет тенденцию достигать **своего уровня некомпетентности**. Если человек успешно справляется со своими обязанностями, его считают подходящей кандидатурой для повышения статуса. После ряда выдвижений он достигает уровня, где обнаруживается его некомпетентность, так как его новые обязанности оказываются ему не по силам. Больше его не повышают, но он остается на том месте, куда попал, хотя с обязанностями своими по-прежнему справиться не в состоянии. Каждая разновидность способностей по-своему вполне реальна, но трудно сравнима с компетентностью, предполагаемой у претендентов на руководящую должность. Этот процесс приводит к тому, что многие, особенно **руководящие должности заняты некомпетентными людьми**, долго остающимися на своих постах. Целесообразно отказываться от всякого повышения в должности, пока специалист

еще находится на уровне своей компетентности. Большинство повышений определяются компетентностью кандидата, обнаруженной им на низшей ступени деловой лестницы.

Помимо навыков планирования, организации, руководства и контроля, которые необходимы для управления проектом, **любой руководитель должен обладать рядом общих психологических свойств**, в том числе [51]:

- оказывать влияние на эмоциональную реакцию других, на перемены в поведении других в противоположность их действиям;
- сдерживать гнев, осознавать и контролировать свои эмоции;
- выработать в себе привычку к строгой самооценке, которая не должна постоянно зависеть от внешнего подтверждения правильности предпринятых действий;
- придерживаться разумных норм проявления чувств;
- разрешать затруднения, с которыми сталкивается группа;
- доверять другим, контролировать свой стиль поведения и ставить интересы коллектива выше личных предпочтений;
- принимать решения относительно подчиненных, полностью исключив всякие эмоции;
- находить внутренние мотивы, чувствуя несогласие;
- находиться в небольшой зависимости от внешнего подтверждения правильности предпринятых действий;
- изучать самооценку других и выяснить их потребности в подтверждении правильности предпринятых действий;
- разрешать конфликты к взаимному согласию;
- убедительно доводить замысел, план или решение задачи до подчиненных;
- осознавать и не допускать проявление вспыльчивости своего характера в ответ на эмоциональную реакцию;
- понимать первопричины мотивации других, умение поставить себя на место другого;
- понимать причины, по которым некоторые вещи могут доставлять или доставлять удовольствие.

Подбор людей для конкретных работ требует реалистичного предварительного анализа, периодической смены работ, ограниченности риска и определения возможных сроков окончания работ, а значит, и поощрения, если работы завершаются досрочно. Выбор ра-

бот для конкретных людей может потребовать укрупнения работ, объединения двух или более специализированных задач для повышения мотивации, повышения разнообразия работ и перепланировки работ ради роста их стимулирующего потенциала. Повышение разнообразия работ может включать в себя учет квалификации, идентичности и важности решаемых задач, автономности и обратной связи с выполняемой работой.

Выполняя в какой-то степени функции, присущие отделу трудовых ресурсов (а возможно, и полностью, если таковой отсутствует), руководитель проекта должен обеспечить ***продвижение по службе и профессиональный рост своих подчиненных***. Для этого он должен проводить аттестацию и контроль эффективности работы подчиненных, делать назначения, продвигать по службе и принимать меры для удержания в организации хороших сотрудников. Изучение деятельности лидеров, администраторов, позволяет сделать вывод о том, что те из них, кого подчиненные характеризовали как влияющих на них руководителей, более эффективные работники.

При производстве сложных комплексов программ большими коллективами значительно повышается роль ***квалификации лидеров – руководителей*** проекта, что непосредственно отражается на производительности и результатах труда всего коллектива. Известны случаи, когда вследствие квалификации руководителей сложных программных проектов, суммарные затраты на разработку изменились в несколько раз: как в лучшую, так и в худшую сторону. Некоторые методы учета характеристик лидеров, организации и структуры коллектива, процессов производства позволяют ***сокращать негативное влияние человеческого фактора***.

Лекция 3

РЕКОМЕНДАЦИИ ПО ОСНОВАМ ОРГАНИЗАЦИИ ПРОФЕССИОНАЛЬНОГО ОБУЧЕНИЯ СПЕЦИАЛИСТОВ ДЛЯ ПРОЕКТИРОВАНИЯ И ПРОИЗВОДСТВА КРУПНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Основные свойства руководителей и специалистов, необходимые при проектировании и производстве крупных программных продуктов

В предыдущих лекциях рассмотрены общие психологические характеристики и стимулирование производственных личностей, особенности и свойства лидеров производственных коллективов. На этой основе начинаются лекции рекомендаций и требований по обучению конкретных специалистов для проектирования и производства крупных программных продуктов. При этом будут представлены перечни и комментарии к свойствам руководителей и специалистов, которые не всегда могут использоваться полностью, но являются базой для выбора из них необходимых в зависимости от характеристик конкретных комплексов программ.

Теоретические и концептуальные основы обучения программной инженерии лежат, прежде всего, в различных областях применения информатики, однако для получения полноценного профессионального образования необходимо быть знакомыми с рядом концепций из иных областей, таких как математика, системотехника, инженерия, управление проектами и одна или же несколько конкретных предметных областей техники. Изучающие программную инженерию должны уметь интегрировать теорию и практику, понимать важность абстракции и моделирования, быть способными разбираться в новых для себя предметных областях, а также понимать значимость хорошего проектирования [37, 39] – рис. 3.1.

По мере повышения квалификации коллективов и автоматизации творческой части труда следует ожидать асимптотического при-

ближения проектов к предельным значениям относительных экономических характеристик новых разработок.

Квалифицированные руководители и специалисты для обучения программной инженерии должны обладать определенными свойствами характера и знаниями:

- теоретических и концептуальных основ современной информатики;
- высокими интеллектуальными возможностями по интенсивности принятия творческих решений;
- системотехники и технологии деятельности коллективов при производстве крупных вычислительных систем;
- освоить профессиональные принципы и коллективную дисциплину, ориентированные на решение крупных проблем;
- обладать навыками общения и межличностных отношений с сотрудниками, способностью критически оценивать конкурирующие решения;
- лидер – руководитель обязан уметь последовательно, поэтапно организовать эффективно действующий производственный коллектив;
- уметь сочетать и балансировать этические, общественные, юридические и экономические интересы различных заинтересованных лиц.

Рис. 3.1

Эти значения определяются ***интеллектуальными возможностями человека по интенсивности принятия творческих решений***. Им соответствуют наличие предельных значений производительности труда и длительности разработки сложных комплексов программ. Для их оценки необходимо изучение и экстраполяция precedентов и экспериментальных данных реальных разработок с лучшими экономическими характеристиками с учетом возрастания профессиональной квалификации специалистов и уровня автоматизации проектирования и производства. Вряд ли можно ожидать в ближайшие годы радикального повышения производительности труда при создании полностью новых, крупных программных продуктов для систем реального времени. Еще более консервативна длительность таких разработок.

Программная инженерия должна преподаваться как дисциплина деятельности коллективов, ориентированная на решение крупных

проблем. Важнейшей задачей проектов по разработке программного продукта является разрешение проблем заказчика – как явных, так и неявных. Необходимо учитывать это при составлении программ и курсов. Осознание данного факта поможет учащимся рационально подходить к процессу обучения, облегчит понимание материала. К сожалению, наиболее частой ошибкой является фокусирование исключительно на технических проблемах, что в дальнейшем приводит к созданию непригодных для использования систем. Существует множество классов важных проблем. Некоторые из них, такие как проблемы анализа, проектирования и тестирования, относятся к продукту и напрямую связаны с разрешением проблем заказчика. Другие же, такие как улучшение и автоматизация процессов, являются мета-проблемами, решение которых ускорит процесс разрешения проблем, относящихся к продукту и к разрешению проблем заказчика. Кроме того, существует ряд проблем, не входящих в эти две категории (например, психологические, этические проблемы). Навыки решения проблем наилучшим образом развиваются при анализе примеров и решении учебных задач во время практических занятий. Руководители и специалисты по программной инженерии **должны быть предварительно обучены и хорошо знать системотехнику производства больших вычислительных систем**, поскольку в них программный продукт играет определяющую роль. Технологии программной инженерии часто являются критическим фактором при разработке сложных вычислительных систем. Высокие темпы роста основных доступных ресурсов аппаратных средств и сохраняющаяся потребность в увеличении использования их функций со стороны различных пользователей и сфер применения приводят к необходимости **адекватного совершенствования технологий создания программных продуктов**.

Для преподавания в этой области, как и для любой другой образовательной деятельности, необходимо иметь достаточное количество преподавателей; преподавательская и административная нагрузки должны позволять им участвовать в научной и профессиональной деятельности. Это важное условие успешности обучения такой динамичной дисциплине как программная инженерия. Для преподавания программной инженерии необходимо наличие преподавателей, прошедших длительное обучение с упором на сложные программные продукты и достаточный практический опыт в этой области. Однако программная инженерия является относительно молодой дисциплиной, и сего-

дня сложно найти преподавателей, соответствующих традиционным академическим стандартам (ученая степень, способность к эффективному преподаванию, исследовательский потенциал) в совокупности с опытом работы в области программной инженерии. Необходимо поощрять и всячески поддерживать стремление преподавателей программной инженерии идти в ногу со временем, предоставляя им возможность проводить исследовательские работы, получать необходимые консультации.

Для *организации эффективной структуры производственного коллектива* следует научиться учитывать согласованность конкретных целей специалистов, участвующих в проекте, их *психологическую совместимость*, способность к дружной коллективной работе, наличие опыта взаимодействия в составе определенного коллектива, а также другие объективные и субъективные свойства участников проекта. При этом большое значение может иметь личная мотивация и психологические особенности поведения разных специалистов при комплексной работе над сложным проектом. Эти характеристики могут быть обобщены в качественный показатель влияния сложности взаимодействия специалистов в коллективе, которому могут быть сопоставлены оценки возможного изменения эффективности производства (см. лекцию 5). Наилучшим обычно считается непрерывное корректное взаимодействие организованных специалистов с большим опытом работы в конкретном коллективе при согласованности их целей, планов и методов работы. В остальных случаях в той или иной степени (даже в 3 – 5 раз) может возрастать трудоемкость разработки, что нельзя не учитывать при экономическом обосновании крупных проектов.

Необходимо обучение *профессиональных коллективов специалистов* современной *программистской культуре* промышленного создания крупных высококачественных программных продуктов – умению формализовать требования и достигать конкретные значения характеристик функционирования и применения сложных комплексов программ с учетом тех ресурсов, которые доступны для обеспечения и совершенствования качества. Для этого при обучении необходимо воспитывать у каждого специалиста *ряд коллективных профессиональных свойств* [42, 51].

Способность критически оценивать конкурирующие решения – одна из основных черт, присущих профессиональному инженеру. По-

этому учебный план и методы преподавания должны помочь учащимся приобрести необходимые знания, навыки анализа и методы эффективной оценки. Важной чертой является стремление к критическому мышлению. Необходимо также обучить умению оценивать надежность различных источников информации.

Оценка и оспаривание получаемых знаний – учащиеся должны научиться не принимать на веру любую информацию от преподавателей или из книг. Они должны также понимать ограниченность современных знаний по программной инженерии и направления, в которых эти знания должны развиваться.

Осознание собственных ограничений вследствие чего профессионалы зачастую прибегают к консультированию у других профессионалов, а также показать преимущества командной работы.

Эффективная коммуникация учит эффективно обмениваться информацией различными способами: письменно, при проведении презентаций, демонстраций собственных и чужих программных разработок, а также во время различного рода обсуждений, должны развить навыки восприятия на слух, навыки сотрудничества и переговоров.

Этичное и профессиональное поведение – должны научиться контролировать соответствие своей деятельности нормам этики, конфиденциальности и безопасности.

Стремление учиться и саморазвиваться – так как большая часть изучаемого изменится в течение будущей профессиональной деятельности специалиста, и лишь малая часть необходимой для изучения информации преподается в вузе. Крайне важно, чтобы учащиеся развивали в себе навыки постоянного расширения собственных знаний.

Специалист программной инженерии – это **член команды – коллектива**, поэтому должен обладать **навыками общения** и межличностных отношений, а также уметь планировать не только свою работу, но и координировать ее с работой других специалистов. Известно, что индивидуальная производительность программистов может отличаться в десятки раз [2, 8, 11]. Производительность одного и того же программиста в разных коллективах может так же отличаться в десятки раз. Поэтому основные усилия руководителя, если он стремится получить наивысшую производительность труда коллектива или рабочей группы, должны быть направлены на изучение и улучшение взаимодействия специалистов: обеспечивать эффективность работы каждого участника коллектива и обеспечивать эффективность про-

цессов взаимодействия специалистов в коллективах разного размера и назначения. Если не учитывать индивидуальные особенности конкретной личности, то эффективность их взаимодействий может сильно снижаться.

Для того чтобы сотрудник мог эффективно решать поставленную задачу, необходимо: понимание целей работы; умение ее делать; иметь возможности ее сделать и желание ее сделать. Для этого **руководитель должен уметь** эффективно выполнять свои функции и решать задачи: обеспечить сотрудникам общее видение целей и стратегии их достижения; обучать, быть наставником и образцом для подражания; помогать, обеспечить сотрудника всем необходимым, убрать препятствия с его пути; обеспечить адекватную мотивацию участника коллектива на протяжении всего проекта. У каждого участника коллектива должна быть личная цель, которую он сможет достичь, продвигая производство комплекса программ. По ходу проектирования и производства мотивы людей, как правило, изменяются, самооценка, скорее всего, неадекватно завышается. Наиболее эффективные производственные процессы складываются в самоуправляемых и самоорганизующихся рабочих командах, для которых характерны ясность общих ценностей и целей, самоконтроль, взаимопомощь, взаимозаменяемость, коллективная ответственность за результаты труда, всемерное развитие и использование индивидуального и группового потенциалов. Эффективные команды обычно **концентрируются вокруг признанного лидера** (см. лекцию 2).

Руководителю разработки программного продукта недостаточно быть хорошим управленцем, он должен стать признанным лидером. Для этого необходимо: признание коллективом профессиональной компетентности и превосходства руководителя и доверие коллектива к действиям и решениям руководителя; признание его исключительных человеческих качеств, убежденность в его честности, порядочности, вера в его искренность и добросовестность. В зависимости от состава, квалификации и ситуации задач участников группы или коллектива у лидера могут быть различные **стратегии руководства при назначении**: руководителем в новый коллектив; в известный слаженный дружественный коллектив; в коллектив, настроенный критически; где каждый сам себе может быть руководителем. Во всех ситуациях лидер обязан последовательно организовать эффективно

действующий производственный коллектив, для чего **пройти этапы** [1, 43, 46]:

- формирования, когда люди должны преодолеть внутренние противоречия, переболеть конфликтами прежде, чем сформируется действительно спаянный коллектив;
- разногласия и конфликты – неизбежные сложности или неудачи порождают конфликты и «поиск виновных», участники команды методом проб и ошибок вырабатывают наиболее эффективные процессы взаимодействия;
- становление – в команде растет доверие, люди начинают замечать в коллегах не только проблемные, но и сильные стороны, закрепляются и оттачиваются наиболее эффективные процессы взаимодействия;
- команда работает эффективно, высок командный дух, люди хорошо знают друг друга и умеют использовать сильные стороны коллег, это лучший период для раскрытия индивидуальных талантов.

Часто случается, что рабочая группа **не достигает высокой производительности**. На совещаниях бесконечные неконструктивные споры и дискуссии. Постоянно доминируют одни и те же лица, другие предпочитают отмалчиваться. Все стараются избегать конфликтов и поддерживать согласие. При возникновении трудных ситуаций, все ждут решения от руководителя, редкие противоречия разрешаются общим голосованием. Подобное избегание производственных конфликтов снижает здоровую интеллектуальную конкуренцию, ведет к шаблонности и застою.

Альтернатива – когда **в эффективном коллективе** целенаправленно стимулируется творческое инакомыслие и интеллектуальная конкуренция. Назначают рецензентов – критиков, которые должны найти слабые стороны в решении, предлагаемом их коллегой для общего обсуждения, или поручают двум специалистам решить одну и ту же критичную для проекта задачу, используя разные подходы или технологии, а затем сравнить и оценить полученные результаты. Ни одно предлагаемое участником команды решение не принимается на веру, активно анализируются возможные негативные последствия или упущеные возможности при принятии решения. Конфликты носят исключительно производственный характер, а все, что человек делает с удовольствием, он делает максимально эффективно. В ре-

зультате **достигается высокая эффективность** производственного коллектива.

В процессе работы над программными комплексами обучающиеся должны быть способными **эффективно решать поставленные перед ними задачи** как индивидуально, так и в команде. В реальной жизни специалистам предстоит выполнять некоторое количество проектов в одиночку или малыми группами, однако многие задачи требуют работы в большом коллективе с другими людьми. Соответственно, специалисты должны овладеть максимально полной информацией о сущности работы коллектива и ролях в команде. Они должны понимать важность таких задач, как дисциплинированный подход, необходимость придерживаться установленных сроков и оценка как индивидуальной, так и командной производительности. Следует разрешать противоречия в стоящих перед проектом целях, находя приемлемые компромиссы в рамках существующих ограничений (стоимость, время, знания, существующие системы и организации). Специалисты должны выполнять задания, содержащие противоречивые и даже изменяющиеся требования.

Необходимо научиться использовать множество различных подходов к инженерному проектированию и к решению специфических проблем в конкретных предметных областях. Проектировать решения в одной или более предметных областях приходится, используя подходы программной инженерии, **сочетающие и балансирующие этические, общественные, юридические и экономические интересы различных заинтересованных сторон**. Следует понимать достоинства и недостатки различных доступных альтернатив и последствия выбора того или иного подхода в каждой конкретной ситуации. В предлагаемых проектных решениях должны адекватно учитываться этические, общественные, юридические, экономические факторы, а также вопросы безопасности.

Следует **понимать важность способности к ведению переговоров**, результативно работать, осуществлять руководство и эффективно общаться с заинтересованными лицами в типичных ситуациях для разработки программных продуктов. Программные инженеры должны осознавать, что им необходимо создавать программные комплексы, прежде всего, **являющиеся полезными для заказчиков и пользователей**. Это должно способствовать получению опыта и созданию необходимой среды для подготовки высококвалифицированных специалистов по программной инженерии.

Рекомендации преподавателям и учащимся при обучении проектированию и производству крупных программных продуктов

Руководителям и специалистам необходимо освоить современный комплекс задач, методов и стандартов промышленного создания и развития сложных, многоверсионных, тиражируемых программных средств и баз данных требуемого высокого качества. Обучение должно быть ориентировано на коллективную, групповую работу коллективов специалистов над средними и крупными программными проектами. Внимание следует акцентировать **на комплексе индустриальных методов и международных стандартов программной инженерии**, которые непосредственно обеспечивают эффективный жизненный цикл сложных высококачественных программных средств и баз данных – рис. 3.2.

Образовательные модели, описанные в [39], позволяют скомпоновать разделы знаний информатики в рационально построенные и легко реализуемые учебные курсы – модули. Четкая формулировка образовательных моделей существенно упростит взаимообмен педагогическими стратегиями и инструментами между вузами. Это также обеспечит определенной базой издателей, выпускающих учебники и другие образовательные материалы. Следует определить фундаментальные навыки и знания, которыми необходимо обладать всем выпускникам, специализирующимся на программной инженерии. **Использование Руководства** [39] должно способствовать выявлению ключевых тем дисциплины программной инженерии и гарантировать включение соответствующего материала в учебные программы.

Студенты, обученные специальности «Программная инженерия», по Руководству SE2004 **должны уметь**:

- показать владение знаниями и навыками программной инженерии, необходимыми для того, чтобы приступить к практической работе;
- работать индивидуально или в группе над созданием качественных программ;
- искать приемлемые компромиссы в рамках ограничений, накладываемых затратами, временем, срокам, знаниями, существующими системами и организацией;
- выполнять проектирование в одной или не скольких пред-

метных областях, используя подходы программной инженерии, объединяющие функциональные, этические, социальные, юридические и экономические интересы;

- демонстрировать понимание и применение существующих теорий, моделей и методов, необходимых для программной инженерии;
- демонстрировать такие навыки, как межличностное общение, эффективные методы коллективной работы, лидерство;
- изучать новые модели, методы и технологии по мере их появления.

***При обучении методам проектирования и производства
крупных программных продуктов преподавателям и учащимся
целесообразно знать и уметь:***

- акцентировать внимание на комплексе индустриальных методов и международных стандартов программной инженерии;
- осваивать методы, учебные планы и требования к процессам обучения на этапах жизненного цикла производства комплекса программ;
- применять обновление учебных планов на основе концепции «обучения на протяжении всей жизни»;
- применять распространенные теории, модели и методы, которые составляют базу для анализа проблем производства крупных программных комплексов;
- осваивать методы обеспечения требуемого качества комплексов программ;
- исследовать и применять качества специалистов наиболее важные с точки зрения конкретных работодателей;
- применять требования конкретных заказчиков к стандартизированной организации процессов производства программного продукта;
- освоить международный стандарт ISO 19759:2005 – ТО. – SWEBOK – Совокупность знаний о разработке программных средств;
- основы формирования совокупности требований к сложным комплексам программ;
- требования к учащимся при завершении обучения программной инженерии, что они должны знать и уметь.

Рис. 3.2

Рекомендации по разработке учебных планов в области программной инженерии должны быть основаны на соответствующем определен-

нии совокупности знаний в этой области. Описание необходимой совокупности знаний должно быть по возможности кратким, подходящим для использования в высшем образовании и опираться на наработки предыдущих исследований. Это описание должно содержать **обязательный набор тем**, необходимых для вузовских программ по программной инженерии. Список обязательных тем должен начинаться с материала для вводных курсов, распространяться на основной учебный план и сопровождаться материалом дополнительных курсов, которые могут варьироваться в зависимости от потребностей вуза, учебной программы или отдельного студента. Для достижения успеха в создании рекомендаций по организации образования в области программной инженерии должны быть привлечены как преподаватели программной инженерии, так и **представители различных заинтересованных сторон**, в том числе индустрии, бизнеса и правительственные органы.

Профессиональная деятельность в области программной инженерии заставляет специалистов сталкиваться с широким спектром задач, таких как разрешение проблем и дефектов, менеджмент и управление, практическое применение этических и моральных норм, письменное и устное общение, работа в команде и необходимость постоянного изучения последних достижений в быстро меняющейся дисциплине. Наравне с рекомендациями высокого уровня, полезно включать в обсуждение различные тактики и методики реализации учебных программ. Очень важно предоставить обобщенное видение вопросов обучения программной инженерии, что успех любого учебного плана существенно зависит от деталей реализации и советов по практическим вопросам учебного плана.

Быстрая эволюция программной инженерии и особенности профессиональной деятельности требуют постоянного **обновления учебных планов**. При разработке планов по информатике необходимо учитывать изменения в технологиях, методиках и приложениях, новые разработки в сфере педагогики, а также важность концепции **«обучения на протяжении всей жизни»**. В такой быстро развивающейся области, как программная инженерия, учебные заведения должны оперативно перенимать передовые стратегии, реагируя на происходящие изменения. Кроме того, обучение программной инженерии должно готовить специалистов к дальнейшему обучению на протяжении всей жизни, что позволит им идти в ногу со временем и быть способными разрешать сложные проблемы будущего.

В результате обучения квалифицированные **специалисты должны демонстрировать знания и навыки в области программной инженерии**, а также иметь профессиональные качества, необходимые для работы над крупными комплексами программ. Необходимо развивать уверенность в возможностях специалистов на протяжении всего периода обучения программной инженерии. В большинстве случаев знания и навыки развиваются путем поэтапного подхода, различные уровни достигаются по мере углубления обучения. Учащиеся должны обрести понимание и способность самостоятельно решать профессиональные вопросы, связанные с этикой профессионального поведения, экономикой и общественными потребностями (см. рис. 3.2).

Необходимо научиться демонстрировать **понимание и способность к применению распространенных теорий, моделей и методов**, которые обеспечивают современную базу для идентификации и анализа проблем, проектирования, разработки, аттестации и документирования программных комплексов. Учебный проект, представляет собой крайне важную деятельность, **логически завершающую обучение**, дает возможность продемонстрировать умение объединять знания из различных курсов и эффективно их применять, продемонстрировать понимание широкого спектра тем программной инженерии и способность применять приобретенные навыки для достижения желаемого эффекта.

Необходимо изучать новые модели, методы и технологии по мере их появления, а также осознавать необходимость **постоянного профессионального роста**: вследствие растущего спроса на высококачественные программные продукты, увеличивающийся **уровень ответственности** разработчиков за их качество и необходимость постоянного повышения квалификации по окончании учебного заведения. Чем больше профессиональной практики получают учащиеся, тем привлекательнее становится для них учеба и реальнее будущая работа. Соответственно, профессиональная практика в учебной программе может служить своеобразным катализатором пробуждения и поддержания интереса специалистов.

Пробелы в обучении специалистов методам программной инженерии и обеспечения требуемого качества комплексов программ при их практической деятельности оставляют широкое поле для произвола при оценивании качества программных продуктов и к проявлению многочисленных дефектов и ошибок. Возрастание сложности и ответственности современных задач, решаемых ком-

плексами программ реального времени, а также возможного ущерба от недостаточного качества их результатов, значительно повысило актуальность освоения учащимися методов полного, *стандартизированного описания и реализации требований* к характеристикам качества на различных этапах жизненного цикла комплекса программ. Необходимо освоение специалистами понятий, определений и способов оценивания реальных характеристик качества программных продуктов. Выявились необходимость систематизации реальных характеристик качества, а также обучения специалистов применению стандартов выбора из них и адаптации необходимой номенклатуры и требуемых значений характеристик для конкретных проектов комплексов программ.

Частный и государственный секторы индустрии заинтересованы в обучении профессионализму. Учащиеся, знакомые с реалиями профессиональной деятельности, понимают значение навыков эффективного общения с коллегами и клиентами, прилагают все усилия для того, чтобы делать свою работу качественно, стремятся к постоянному повышению своей квалификации и усовершенствованию своей фирмы. Каждый год Национальная ассоциация колледжей и работодателей проводит исследования для выяснения, *какие качества специалистов наиболее важны с точки зрения работодателей*. Список десяти основных факторов включает [39]:

- навыки эффективного общения в коллективе (как устного, так и письменного);
- честность;
- умение налаживать межличностные отношения;
- мотивированность и инициативность;
- развитая профессиональная этика;
- аналитические навыки;
- гибкость и адаптируемость к задачам;
- навыки работы с компьютером;
- уверенность в себе.

Результаты этого и подобных исследований подчеркивают важность взгляда на профессионализм как на центральный компонент обучения. Возрастающая потребность в высококачественных продуктах также стимулирует усиление роли профессионализма в процессе обучения. *Анархичные подходы* к написанию программных компонентов завоевали устойчивую репутацию *основного источника проблем и*

дефектов. Результатом этого служат возрастающие *требования заказчиков к стандартизированной организации процессов производства* программного продукта у поставщиков, причем без удовлетворения этих требований зачастую невозможно получить контракты. В частности, после потерь миллионов долларов, выплаченных за неработающие или незаконченные программные продукты, государственные органы США начали требовать от всех своих подрядчиков, соответствия как минимум *третьему уровню зрелости* модели СММ (см. лекцию 11). Для соответствия этим стандартам подрядчики должны наладить стабильный, надежный и устойчивый процесс производства программного продукта. Специалисты должны понимать значимость налаживания личных контактов с заказчиками, согласования детальных требований к будущим программам и необходимых усилий по достижению наивысшего уровня качества.

В качестве основы для *подготовки учебных планов и структуры курсов по программной инженерии* может рассматриваться *международный стандарт ISO 19759:2005 – ТО. – SWEBOK* – Совокупность знаний о разработке программных средств. Руководство представляет фундаментальное, энциклопедическое обобщение современных процессов, методов и средств практического создания и обеспечения жизненного цикла сложных комплексов программ, выполненное большим международным коллективом авторов. Стандарт создан на базе проекта *SWEBOK, A project of IEEE Computer Society Professional Practices Committee*.

В предисловии и введении изложены история развития, цели и функции *практической программной инженерии*. Весь материал можно разделить на три части: базовые методы (разделы 1 – 6); процессы, технология и средства (разделы 7 – 12); приложения и комментарии (разделы А, В, С, Д). Руководство содержит разделы:

1. Введение в Руководство;
2. Требования к программным средствам;
3. Проектирование программных средств;
4. Конструирование программных средств;
5. Тестирование программных средств;
6. Сопровождение программных средств;
7. Управление конфигурацией программных средств;
8. Управление технологией создания программных средств;
9. Технологические процессы программных средств;

10. Технологические методы и средства;
11. Качество программных средств;
12. Взаимодействие дисциплин и технологий программных средств.

Внимание акцентируется на обеспечении эффективности процессов практической разработки и сопровождения сложных комплексов программ. Важнейшими компонентами улучшения этих процессов являются количественные и качественные измерения результатов деятельности специалистов, обучение и повышение их квалификации. Для этого необходимо управление персоналом, освоение новых технологий и инструментальных средств, контроль состояния, качества и изменения компонентов проекта, эффективное использование всех видов ресурсов. Представленное выше **оглавление Руководства адекватно традиционным моделям** жизненного цикла сложных программных средств. В Приложениях, в частности, приводится эволюция развития Руководства и сопоставление его структуры и содержания с существующими стандартами IEEE и ISO по проблемам программной инженерии и управления сложными проектами комплексов программ. Каждый раздел основного текста содержит иллюстрации в виде схем и таблиц, список литературы и соответствующих стандартов. В целом Руководство целесообразно использовать как учебное пособие для повышения системной квалификации специалистов по программной инженерии.

В Руководстве [39] особое внимание обращается на важность преподавания методов и процессов **формирования требований** к сложным комплексам программ, включающим:

- основы инженерии требований к программному продукту;
- процессы инженерии требований: выявление требований, спецификация, анализ и управление;
- типы требований: функциональные, нефункциональные, атрибуты качества;
- выявление требований: определение потребностей, целей и требований, заказчики и другие заинтересованные лица, опросы и наблюдения;
- спецификация требований: текстовые и графические нотации и языки, методы написания высококачественных требований, стандарты документирования;
- анализ требований: инспекция, аттестация, завершенность,

обнаружение конфликтов и несоответствий, анализ взаимодействия компонентов функциональности и разрешение противоречий;

- целенаправленное и ориентированное на варианты использования: моделирование, прототипирование и методы анализа;
- требования к типичным системам: встроенным системам, потребительским системам, бизнес-системам, научным системам и другим инженерным системам;
- управление требованиями: отслеживание, приоритеты, изменения, базовые линии и инструментальная поддержка;
- согласование требований и управление рисками;
- интеграция анализа требований и процессов разработки программного продукта.

Сформулированы требования к учащимся при завершении обучения – они должны знать и уметь [39]:

- обнаруживать или выявлять требования к программному продукту, используя различные методы;
- упорядочивать приоритеты требований;
- применять методы анализа, такие как анализ потребностей, анализ целей и вариантов использования;
- оценивать требования в соответствии с критериями выполнимости, ясности, отсутствием неоднозначностей;
- представлять функциональные и нефункциональные требования для различных типов систем, используя формальные и неформальные методы;
- определять и измерять атрибуты качества продукта;
- вести переговоры с различными заинтересованными лицами для достижения согласия по множеству требований;
- обнаруживать и разрешать проблемы взаимодействия компонентов.

В данном учебнике основное внимание сосредоточено на обучении формализации и реализации ***наиболее важных рекомендаций и требований к профессиональной квалификации руководителей и специалистов*** для:

- формирования требований к крупным программным продуктам;
- оценивания и обеспечения экономических характеристик крупных программных продуктов;

- планирования деятельности коллективов при производстве крупных программных продуктов;
- обеспечения качества программных продуктов;
- сопровождения и управления конфигурацией крупных программных продуктов;
- тестирования компонентов и крупных комплексов программ;
- динамических испытаний крупных программных продуктов;
- применения международных стандартов производства крупных программных продуктов.

Модель обучения и обеспечения квалификации руководителей и специалистов для производства крупных программных продуктов

Для планирования и управления жизненным циклом концептуально целостных, крупных программных продуктов и обеспечения их качества в пределах допустимых затрат необходимы организационные действия менеджеров, направленные на подбор и обучение коллектива специалистов разных категорий и специализаций. Для обеспечения квалификации персонала, участвующего в крупных проектах, *требуется соответствующая их подготовка, отбор и обучение*, которые являются самостоятельной, важной *проблемой в экономике производства* программных продуктов. Обучение представляет собой сложный процесс и требует организации и сопровождения обучаемого персонала. Должны быть разработаны и документированы планы, требования и цели обучения, а также разработаны материалы, используемые для обучения [39, 51]. Преподавательский персонал, ответственный за выполнение конкретных задач, если это необходимо, должен быть *аттестован* на основе соответствующего образования, подготовки и/или опыта работы. Может также быть необходимым включение в подготовку, ознакомление со специфической (проблемно-ориентированной) областью, в которой будет работать данный программный продукт, и повышение квалификации в этой области.

Быстрый рост сложности и повышение ответственности за качество комплексов программ привели к появлению *новых требований к обучению и к дифференциации специалистов по программной*

инженерии, обеспечивающих основные этапы жизненного цикла комплексов программ. Стало ясно, что специалистам недостаточно навыков процедурного программирования небольших компонентов, а необходимы глубокие знания системотехники, технологии и стандартов проектирования, методов обеспечения и контроля качества комплексов программ в определенной области их применения. Эти специалисты должны владеть новой интеллектуальной профессией, обеспечивающей высокое качество программных продуктов, а также контроль, испытания и удостоверение реального достигнутого качества на каждом этапе производства и совершенствования программ. Крупномасштабное проектирование различных классов, разделение труда специалистов по квалификации при разработке программ и данных, структура и организация коллективов, а также экономика таких разработок стали важнейшей частью **проблемы выбора и обучения специалистов разной квалификации** для обеспечения всех этапов и процессов ЖЦ сложных программных продуктов.

Для **оценивания и учета квалификации коллектива специалистов** при производстве программных продуктов полезно освоить представленную ниже **модель уровня зрелости персонала производственного коллектива**. Институт инженерии программного обеспечения (Software Engineering Institute – SEI) в США долгое время занимается **Программой усовершенствования процесса создания сложных программных продуктов**. Модель СММ (Capability Maturity Model – модель оценки уровня развития) является частью этой Программы. Она включает в себя наилучшее из практики **производства** крупных программных продуктов – рис. 3.3. В развитие этой модели институт SEI предложил **модель оценки уровня зрелости персонала коллектива** (People Capability Maturity Model – P-СММ). Ее полезно использовать в качестве основы стратегии управления крупными человеческими ресурсами целостных коллективов в предприятии [42]. Стратегические цели этой модели:

- расширение возможностей предприятий, занимающихся программными продуктами, путем оценивания и повышения квалификации (зрелости) коллективов специалистов;
- гарантирование того, что способность высококачественной разработки программного продукта является отличительной чертой всего коллектива специалистов предприятия, а не только нескольких людей, которые занимаются непосредственно отдельными задачами;

- обеспечение совместимости между мотивацией отдельного индивидуума и мотивацией всего коллектива производства программного продукта и предприятия;
- сохранение в составе коллективов и предприятия ценных человеческих ресурсов сотрудников, имеющих глубокие знания и высокий качественный опыт.

Подобно базовой модели **СММ**, модель **P-СММ** имеет пять уровней зрелости коллективов специалистов (см. рис. 3.3):

1. **Начальный уровень** – осуществляется управление специалистами коллектива, не оформленное в виде определенных правил и инструкций;

2. **Периодически повторяющиеся действия руководителей** – проведение политики, направленной на развитие регламентирующих способностей руководителей и специалистов коллектива проекта или предприятия;

3. **Становление коллектива** – введение в предприятии стандарта управления крупным коллективом специалистов, основанного на лучшем опыте и стандартах управления специалистами;

4. **Управление коллективом** – определяются и вводятся количественные цели и оценки качества управления коллективами специалистов;

5. **Оптимизация коллектива** – центр внимания переносится на непрерывное повышение квалификации и мотивации руководства специалистами коллектива и предприятия.

Модель **P-СММ** – это упорядоченный подход к улучшению и контролю качества управления крупными коллективами специалистов. Он содержит основы для мотивирования, признания, нормализации и совершенствования лучшего опыта в управлении специалистами. При использовании этой модели осознается значимость специалиста как личности и необходимость его дальнейшего совершенствования. Для внедрения модели в полном объеме требуются значительные затраты, а многим предприятиям, не реализующим крупные программные продукты, это не нужно.

Однако эта модель в любом случае является отличным пособием, которое поможет многим руководителям и коллективам специалистов *усовершенствовать свои возможности в производстве сложного программного продукта высокого качества* – рис. 3.4.

Модель оценки уровня зрелости специалистов коллектива

Непрерывное совершенствование методов, развитие профессиональных способностей специалистов и коллектива в целом

Качественный менеджмент развития способностей специалистов и создание коллектива из компетентных сотрудников

Определение уровня исходной профессиональной квалификации коллектива специалистов и управление им в соответствии с этим показателем

Учреждение основ дисциплины среди специалистов коллектива

Рис. 3.3

5. Оптимизация коллектива

- последовательное введение новшеств в производство;
- обучение профессией;
- индивидуальное повышение квалификации специалистов.

4. Управление коллективом:

- выравнивание интенсивности труда в коллективе;
- повышение уровня управления коллективом;
- деятельность, основанная на координации членов коллектива;
- организация коллектива;
- воспитание и руководство коллективом.

3. Становление коллектива:

- культура совместных действий;
- обучение и производство в соответствии с квалификацией;
- развитие карьеры специалистов;
- повышение квалификации;
- планирование работы коллектива;
- анализ знаний и квалификации специалистов.

2. Периодически повторяющиеся действия руководителей:

- оплата труда специалистов;
- начальное обучение специальности;
- укомплектование штатов кадров;
- коммуникации в коллективе;
- рабочая среда коллектива;
- управление производительностью.

1. Начальная стадия – формирование коллектива

Квалифицированные руководители и специалисты для производства крупных программных продуктов должны быть обучены:

- оцениванию и учету профессиональной квалификации коллектива специалистов при производстве программных продуктов на основе модели уровня зрелости коллектива;
- подготовке, отбору, обучению и обеспечению профессиональной квалификации персонала производственного коллектива;
- организации взаимодействия представителей заказчика и разработчиков программного продукта;
- деятельности проблемно-ориентированных руководителей и системных архитекторов, способных переводить требования заказчиков в спецификации компонентов и комплекса программ;
- организации эффективной структуры производственного коллектива и процессов в соответствии с задачами комплекса программ;
- сокращению творчества на этапах, где возможны типовые, стандартные решения и использование апробированных программных компонентов;
- обеспечению качества продукта, на базе стандартов, поддерживающих регистрацию, контроль, документирование и показатели качества на этапах ЖЦ комплекса программ;
- профессиональному в процессах обучения и юридической ответственности за результаты деятельности коллектива специалистов.

Рис. 3.4

Уровень ***квалификации заказчика*** и определенность технического задания на разработку программного продукта может весьма сильно влиять на суммарные затраты и длительность создания комплекса программ.

Первоначальные требования технического задания зачастую оказывается недостаточно квалифицированными и подвергаются в дальнейшем многократным изменениям. Изменения технического задания заказчиком и объем переделок непосредственно отражаются на производительности труда коллектива специалистов. Особенно сильно низкая достоверность технического задания может влиять на попытки заказчика форсировать сроки разработки.

Этому же может способствовать различие между заказчиком и разработчиком в квалификации, уровне понимания целей разработки и необходимых затрат на реализацию возникающих дополнительных требований. Даже при испытаниях заказчик зачастую обнаруживает,

что решаются не совсем те задачи и не совсем так, как ему нужно, вследствие чего необходима большая переработка готовых программ. Даже квалифицированные заказчики вынуждены иногда корректировать техническое задание на любых этапах производства, что может влиять на снижение производительности на 10 – 20%. Представители заказчика, участвующие в проекте, **должны обучаться формализации** функций систем и технологических процессов, для которых предназначены соответствующие программные продукты, и иметь представление об эффективных путях их реализации.

Для создания высококачественных комплексов программ, прежде всего, **необходимо обучение специалистов** процессам организации **взаимодействия представителей заказчика и разработчиков** проекта. Взгляды и требования заказчика, в основном, отражаются на функциональных и потребительских характеристиках продукта. Устремления разработчиков направлены на способы их реализации с требуемым качеством. Эти различия исходных интересов и точек зрения на проект приводят к тому, что некоторые неформализованные представления тех и других имеют зоны неоднозначности и взаимного непонимания, что может приводить к конфликтам. Организация четкого взаимодействия и сокращение этих зон требует определенных мероприятий и контактов по обмену знаниями в процессе проведения совещаний, взаимному повышению квалификации и обучению.

Для определения проекта и уточнения целей заказчика может понадобиться вести с ним переговоры об объеме работ для команды. Во время переговоров с заказчиком о техническом задании и требованиях разработчикам необходимо учитывать неизбежные издержки при разработке комплекса программ (непредвиденные технологические риски, изменения требований, задержки при приобретении закупаемых компонентов, непредвиденный уход членов команды), которые могут приводить к существенному нарушению графика производства. Однако заказчики обычно желают получить как можно больше новых функциональных возможностей программного продукта. Именно эти функциональные возможности создают добавленную стоимость, которая важна для достижения бизнес-целей системы. Специалистам команды следует **научиться пониманию расширений требований заказчиков**, поскольку именно они, в конечном счете, добиваются успеха при применении программного продукта.

При разработке сложных программ большими коллективами значительно повышается роль **квалификации руководителей – менеджеров** проекта, что непосредственно отражается на производительности труда всего коллектива. Известны случаи, когда только вследствие квалификации руководителей крупных проектов суммарные затраты на разработку изменялись в несколько раз как в лучшую, так и в худшую сторону. Некоторые методы организации структуры коллектива и процессов производства позволяют сокращать негативное влияние этого человеческого фактора. Однако формализовать и количественно учесть влияние особенностей конкретного руководителя разработки и ведущих специалистов на реализацию комплекса программ пока трудно.

Коллектив специалистов должен иметь в своем составе квалифицированных менеджеров, **проблемно-ориентированных системных архитекторов**, способных переводить не всегда полные и корректные функциональные требования заказчиков в конкретные спецификации и технические требования к комплексам программ и к их компонентам. Специалисты по проектированию (системные архитекторы) должны иметь, прежде всего, хорошую подготовку по системному анализу алгоритмов и комплексов программ в определенной проблемной области, по методам оценки эффективности проектов, организации и планированию крупных проектов и компонентов программ и баз данных. Им необходима **высокая квалификация по архитектурному построению**, комплексной отладке и испытаниям программных продуктов определенных классов, умение организовать коллектив для решения общей целевой задачи системы. Это позволит на ранних этапах исключать или сокращать дефекты, обусловленные различием представления ими целей и задач проектов, а также их показателей качества.

Принципиальным путем улучшения экономических характеристик при производстве сложных программных продуктов является исключение творчества на тех этапах, где возможны **типовые, стандартные решения** и использование апробированных заготовок программных компонентов, не требующие при их применении высококвалифицированного творческого труда. Основой такого подхода является применение унифицированной **профессиональной технологии**, готовых, испытанных программных компонентов и стандартизированной архитектуры комплексов программ определенных классов.

Использование готовых апробированных модулей может почти исключать творческий труд по их программированию, автономной отладке и документированию. На этих этапах творческие усилия необходимы для отбора и контроля готовых компонентов, а также для разработки новых, отсутствующих среди апробированных. Однако практически полностью сохраняется творческий труд при системном анализе, при комплексировании компонентов и их комплексной отладке, а также во время испытания программного продукта в целом.

Специалисты, управляющие обеспечением качества программного продукта, должны овладеть стандартами и профессиональными методиками предприятия, поддерживающими регистрацию, контроль, документирование и воздействия на показатели качества на всех этапах ЖЦ комплекса программ. Они должны быть *обучены эксплуатации системы качества* проекта, выявлению всех отклонений от стандартов, от заданных показателей качества объектов и процессов, а также от предписанной технологии на промежуточных и заключительных этапах проектирования и производства.

Предприятие в целом (или его часть) и каждый процесс должен быть рассмотрен с точки зрения ответственности (обязанностей) *юридической стороны* договора с заказчиком. Предприятие может выполнять один или несколько процессов. Сторона, выполняющая процесс, несет ответственность за результаты и весь данный процесс, даже если выполнение отдельных задач поручено другим специалистам. *Принцип ответственности* специалистов по архитектуре и процессам жизненного цикла комплекса программ облегчает контроль применения стандартов для конкретного проекта, в который может быть вовлечено множество лиц.

Для предотвращения негативных последствий при применении программных продуктов в процессе систематического обучения руководителей и специалистов внимание целесообразно акцентировать *на комплексе индустриальных методов и международных стандартов программной инженерии*. Это непосредственно обеспечивает эффективный жизненный цикл сложных высококачественных программных комплексов. Необходимо обучение специалистов современной *программистской культуре* промышленного создания крупных высококачественных продуктов – умению формализовать требования и достигать конкретные значения характеристик функционирования и применения сложных комплексов программ, с учетом тех ре-

сурсов и сроков, которые доступны для обеспечения и совершенствования качества. Для этого необходимо *разрабатывать и включать в учебные программы вузов* по специальности «Информационные системы» *учебный курс «Программная инженерия»* и совокупность поддерживающих его курсов.

Основы организации и система курсов для преподавания в вузах «Информатики» и «Программной инженерии» изложены в детальном документе – **SE2004** [39]. Этот документ ориентирован на планирование обучения студентов и разработку учебных планов по программной инженерии в составе полной вузовской программы по информатике и информационным системам разных классов. Основой базового курса по Программной инженерии может быть учебник [33]. Кроме того, для освоения и расширения знаний, а также при подготовке преподавателями лекций в этой области, можно рекомендовать монографию [35].

В фундаментальных книгах в этой области [6, 20, 42, 51], изданных в последние годы (переводы с английского), последовательность тем и разделов, в основном, соответствует естественным этапам жизненного цикла крупных программных комплексов. Для облегчения подготовки преподавателей и учащихся иногда в учебных планах предлагается для более глубокого изучения структурировать полную тематику программной инженерии на несколько семестров в разных учебных курсах. Однако это может нарушать концептуальную целостность предоставления учащимися о методах и задачах программной инженерии в жизненном цикле крупных программных средств разных классов.

Предлагаемые ниже лекции *рекомендаций и требований к квалификации специалистов основным по этапам производства крупных программных продуктов* предназначены для заказчиков, менеджеров – руководителей крупных проектов, для аналитиков и ведущих специалистов, обеспечивающих все этапы жизненного цикла крупных программных средств и систем. Систематизируются требования к обучению специалистов, что может быть полезно исполнителям научных проектов и опытно-конструкторских работ, к которым предъявляются высокие требования к качеству функционирования и ограничены доступные ресурсы производства программных продуктов. Материалы лекций могут служить базой при *подготовке и внедрении систем качества предприятий*, создающих сложные конкурентоспособные программные продукты, для их сертификации на соответствие международным стандартам.

Лекция 4

ТРЕБОВАНИЯ К ПРОФЕССИОНАЛЬНОЙ КВАЛИФИКАЦИИ РУКОВОДИТЕЛЕЙ И СПЕЦИАЛИСТОВ, РАЗРАБАТЫВАЮЩИХ ТРЕБОВАНИЯ К КРУПНЫМ ПРОГРАММНЫМ ПРОДУКТАМ

Основы формирования требований к крупным программным продуктам

Неформальный подход определения требований, применяющийся при построении простых программ, недостаточен для разработки больших комплексов программ. Стоимость аппаратных средств постепенно снижается, тогда как *стоимость программных продуктов стремительно возрастает*. Возрастает как размер программного продукта, так и его сложность. Это также является одной из причин возникновения проблем при разработке крупных программных продуктов, как и то, что многие компании, занимающиеся их производством, не уделяют должного внимания эффективному применению специалистами современных методов и стандартов, разработанных в программной инженерии. Менеджеры и коллективы специалистов должны понимать, что они работают в организационных и финансовых рамках заключенных контрактов, и должны решать поставленные задачи *с учетом условий и требований договора на систему*. Программирование компонентов – это дело, главным образом, индивидуальное, а программная инженерия систем – всегда коллективная работа.

Менеджеры, руководители – разработчики должны применить методы и процессы для того, чтобы *понять решаемую проблему заказчика* до начала разработки программного продукта (рис.4.1). Для этого следует использовать *анализ, выявление и освоение проблемы и интересов заказчика*:

- достигнуть соглашения между заказчиком и разработчиком по определению проблемы, целей и задач продукта;

- выделить основные причины – проблемы, являющиеся их источниками и стоящие за основной проблемой проекта системы и программного продукта;
- выявить заинтересованных лиц и пользователей, чье коллективное мнение и оценка в конечном итоге определяет успех или неудачу продукта;
- определить, где приблизительно находятся область и границы возможных решений проблем;
- понять ограничения, которые будут наложены на проект, коллектив специалистов и решение проблем.

***Квалифицированные руководители и специалисты
при формировании требований к крупным программным
продуктам должны знать и уметь:***

- понять и освоить проблему, которую предстоит решать программным продуктом для заказчика и пользователей;
- выделить и сформулировать основные факторы, определяющие успех решения проблемы;
- определить основные системно-функциональные требования к программному продукту;
- определить общесистемные ограничения и выделяемые ресурсы для производства программного продукта;
- выделить и ранжировать приоритеты требований к программному продукту для различных заинтересованных лиц;
- проверить корректность и реализуемость требований к программному продукту с учетом доступных ресурсов и сроков;
- обосновать и выбрать схему организации коллектива специалистов для производства программного продукта.

Рис. 4.1

Понимание потребностей пользователей необходимый организационный этап, так как разработчики редко получают совершенные спецификации требований к создаваемой системе, квалифицированные руководители должны сами ***добывать*** информацию, необходимую им для успешной работы. Термин ***выявление требований*** точно отражает данный процесс, в котором разработчики должны играть активную роль. Чтобы помочь коллективу специалистов решить эти проблемы, лучше понять потребности пользователей и других заинтересованных лиц, целесообразно использовать ***методы*** [9, 25, 57]:

- интервьюирования и анкетирования – создание структурированного интервью;
- проведение интервью с 5 – 15 пользователями и/или заинтересованными лицами;
- подведение итогов совокупности интервью, формулирование 10 – 15 наиболее часто упоминавшихся функциональных и архитектурных потребностей заказчика и пользователей;
- совещания, посвященные анализу и синтезу требований – формулирование и определение целей программного продукта;
- ознакомление с ними всех участников проекта и установление, что они с ними согласны; если это не так, следует остановиться и уточнениями добиться согласия; обязательно убедиться в согласии заказчиков;
- анализ иллюстративных прецедентов в приложении к концепции требований, чтобы их функции были наглядны и понятны;
- по возможности, выявление или создание временных прототипов на основе первичных требований.

Хотя ни один из методов не является универсальным, каждый из них позволяет лучше понять потребности пользователей и тем самым превратить неясные требования в требования, которые известны и понятны. Каждый из этих методов эффективен в определенных ситуациях, однако целесообразно отдавать предпочтение совещаниям, посвященным требованиям.

При планировании, производстве и реализации *требований заказчика к характеристикам программного продукта* необходимо, в первую очередь, учитывать следующие основные факторы [9, 42, 51]:

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

Эти факторы целесообразно применять последовательно, итерационно на каждом из основных этапов жизненного цикла комплекса программ. При первоначальном определении требований к функциональной пригодности и к конструктивным характеристикам качества,

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

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

Программные **руководители и системные аналитики** должны быть знакомы с основными способами проектирования сложных систем, знать, **как перевести расплывчатые требования и пожелания заказчика системы в четкое техническое задание**, и уметь разговаривать с заказчиком и потенциальными пользователями системы на языке предметной области, а не на профессиональном программистском жаргоне. Такие способности требуют, в свою очередь, гибкости и открытости, чтобы выделять сущность предметной области программных комплексов систем. Руководитель комплекса программ – это член команды, поэтому должен обладать навыками общения и

межличностных отношений, а также уметь планировать не только свою работу, но и координировать ее с работой других.

В требованиях к продукту и процессу должно проводиться разграничение соответствующих требований как свойств продукта, который необходимо получить, и процесса, с помощью которого продукт будет создаваться; при этом ряд требований может быть заложен неявно и программные требования могут порождать требования к процессу. Они устанавливают основные соглашения между пользователями (заказчиками) и разработчиками в отношении того, что должна делать система и чего от нее не стоит ожидать. Документ должен включать процедуры проверки получаемого программного продукта на соответствие предъявляемым ему требованиям, характеристики, определяющие качество и методы его оценки, вопросы безопасности и другие свойства. В любом случае, задача состоит в том, чтобы программные требования были ясны, связи между ними прозрачны, а содержание спецификаций не допускало разнотечений и интерпретаций, способных привести к созданию программного продукта, не отвечающего *потребностям заинтересованных лиц* [9, 25].

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

Для конкретного комплекса программ доминирующие требования выделяются и определяются его *функциональным назначением*. Программные продукты для ЭВМ как объекты проектирования, производства, испытаний и оценки качества характеризуются в жизнен-

ном цикле следующими **обобщенными характеристиками** (см. рис. 4.1):

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

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

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

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

При разработке заказчиком и руководителем проекта требований технического задания на крупный комплекс программ должны учитываться **общесистемные ограничения, которыми могут быть** [16, 25, 46]:

- экономические – финансовые или бюджетные ограничения ресурсов; себестоимость и ценообразование; проблемы лицензирования;
- политические – внешние или внутренние политические ограничения предприятия, влияющие на потенциальное решение; проблемы в отношениях между подразделениями;
- технические ограничения проекта в выборе технологий; требования работать в рамках существующих платформ или технологий; запрет использования любых новых технологий; ограничения использовать закупаемые пакеты программной индустрии;
- системные требования обеспечивать совместимость с существующими решениями и системами; применение определенных операционных систем и внешней среды комплекса программ;
- эксплуатационные ограничения информационной среды; правовые или юридические ограничения; требования безопасности; ограничения требованиями стандартов;

- ресурсы – ограничения сроками и графиком проекта; ограничения доступными техническими и кадровыми ресурсами; возможность привлечения дополнительных специалистов.

Квалифицированные специалисты по созданию программного продукта **должны понимать функции, задачи и методы создания системы**, поскольку возникающие проблемы часто являются результатом решений, принятых руководителями и разработчиками проекта всей системы. Высокие темпы роста основных ресурсов аппаратных средств (приблизительно на порядок каждые пять лет) и сохраняющаяся потребность в увеличении их использования со стороны различных пользователей и сфер применения приводят к необходимости изменения и повышения квалификации специалистов и адекватного совершенствования технологий создания крупных комплексов программ.

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

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

Особенности подготовки и реализации требований заинтересованных лиц к программному продукту

Системная эффективность – функциональная пригодность комплекса программ может быть описана количественно или качественно в виде набора полезных свойств и характеристик программного продукта, их отличий от имеющихся у других комплексов программ, а также источников возможной эффективности. Она определяет назначение, основные функции и требования заказчика, какие задачи должны решаться для удовлетворения пользователей, а дополнительные, конструктивные характеристики качества – как, и при каких условиях заданные функции могут выполняться с требуемым качеством. В результате может быть formalизована **цель использования** и набор главных характеристик, **требований заказчика и пользователей** при заказе или приобретении программного продукта, а также предполагаемая его сфера назначения и применения. Полнота и точность представления этих характеристик являются исходными для прослеживания реализации всех последующих, производных свойств и качества функциональной пригодности комплексов программ.

Данное требование связано с тем, **какие** функции и задачи должен решать программный продукт для удовлетворения потребностей пользователей в то время как другие, конструктивные требования главным образом связаны с тем, как и при каких условиях, заданные функции могут выполняться с требуемым качеством. Функциональная пригодность – это набор и описания атрибутов, определяющих **назначение, основные, необходимые и достаточные функции программного продукта**, заданные техническим заданием и спецификациями требований заказчика или потенциального пользователя. Номенклатура и значения всех остальных показателей качества непосредственно определяются требуемыми функциями программного средства и в той или иной степени влияют на выполнение этих функций. Поэтому выбор функциональной пригодности, подробное и корректное описание ее свойств являются основными исходными данными для установления требуемых значений всех остальных стандартизованных показателей качества. Атрибутами этой характеристики могут быть функциональная полнота решения заданного комплекса задач, степень покрытия функциональных требований специфика-

циями и их стабильность при совершенствовании программ, число и полнота реализуемых требований заказчика.

Функции и основные характеристики крупных комплексов программ условно можно отразить многомерными *пространствами свойств и значений требований*, контроля и обеспечения их реализации. В жизненном цикле крупных программных комплексов можно выделить три пространства требований, функции и характеристики которых используются и взаимодействуют в следующих целях и назначении:

- исходные *утвержденные требования заказчика* к функциям и характеристикам программного продукта, согласованные с разработчиками в виде конкретных документов, в соответствии с которыми разработчики обязаны создать и обеспечить применение программного продукта пользователями или в составе системы;
- *реально реализованные разработчиками* функции и характеристики программного продукта, которые обычно не могут полностью и абсолютно точно соответствовать исходным требованиям, достоверно *не известны* заказчику, разработчикам и пользователям, и не отражены документами;
- реальные функции и характеристики программного продукта, которые *практически используются пользователями* и/или системой в соответствии с эксплуатационной документацией и могут не совпадать с исходными реализованными требованиями вследствие превышения их значений или не полного их использования.

Функции и характеристики в перечисленных областях обычно не совпадают, но в некоторой степени пересекаются и в совокупности отражают возможные ситуации положения пространства требований, которые полезно иметь в виду при организации и планировании тестирования. Функции и характеристики программного продукта, которые достигаются реально разработчиками, обычно достоверно не известны или не используются заказчиком и пользователями, или могут находиться за пределами заданных исходных требований. Наибольшее значение для реального обеспечения качества программного продукта имеет *полнота соответствия* исходных требований к функциям и характеристикам, пространствам выполненных проверок при тестировании адекватными им по содержанию и сценариям тестами. В результате совокупности *требований к тестам* могут применяться *разработчиками как эталоны и вторая адекватная форма*

ма описания требований к комплексу программ для сквозной верификации спецификаций требований к тестам сверху вниз, а также сами подвергаться верификации на корректность соответствия исходным требованиям к функциям и характеристикам компонентов программ разного уровня.

Для сложных программных продуктов необходима **стратегия управления требованиями**. Для этого применяется информационная *иерархия*; она начинается с потребностей пользователей, описанных с помощью функций системы, которые затем превращаются в более подробные требования к программному средству, выраженные посредством прецедентов или традиционных форм описания и стандартизованных характеристик. В требованиях к комплексу программ следует указывать, **какие функции** должны осуществляться, а не то, как они могут реализоваться. Нужно использовать набор характеристик для установления и оценки качества комплекса и содержащихся в нем функциональных компонентов.

Функциональные требования – определяют поведение программного продукта, который **должен быть создан разработчиками** для предоставления возможности выполнения заказчиком и/или пользователями своих обязанностей в контексте пользовательских требований. Выбор требований к характеристикам при проектировании программных средств начинается с определения исходных данных. Для корректного выбора и установления требований к характеристикам качества, прежде всего, необходимо определить основные особенности программной системы. На основе этих данных должен формироваться общий **набор требуемых характеристик, свойств, их мер и значений качества для определенных потребителей в жизненном цикле программного средства**:

- цели создания программного продукта и назначение комплекса функциональных задач;
- общие требования к функциям и характеристикам комплекса задач программного продукта;
- перечень объектов и характеристик внешней среды применения программного продукта (технологических объектов управления, подразделений предприятия и т.п.) при управлении которыми должен решаться комплекс задач;
- периодичность и продолжительность решения комплекса задач;

- связи и взаимодействие комплекса задач с внешней средой и другими компонентами системы;
- распределение функций между персоналом, программными и техническими средствами при различных ситуациях решения требуемого комплекса функциональных задач.

Проекты, как правило, инициируются заказчиком *с объемом функциональных возможностей, превышающим* тот, который разработчик может реализовать, обеспечив приемлемое качество при заданных ресурсах и сроках. Тем не менее, необходимо ограничиваться, чтобы иметь возможность предоставить в срок *достаточно целостный и качественный программный продукт*. Существуют различные методы задания очередности выполнения (приоритеты) требований и понятие базового уровня – совместно согласованного представления о том, в чем будут состоять ключевые функции системы как продукта проекта – понятие состава требований, задающих ориентир для принятия решений и их оценки [9, 28].

Если *масштаб проекта и сопутствующие требования заказчика превышают реальные ресурсы*, в любом случае придется ограничиваться в функциях и качестве комплекса программ. Поэтому следует определять, что *обязательно должно* быть сделано в первой или очередной версии системы и программного продукта при имеющихся ресурсах проекта. Для этого приходится вести переговоры. Нельзя ожидать, что данный процесс полностью решит проблемы масштаба и требований к программному продукту. Но такие шаги окажут заметное воздействие на размеры проблемы, позволят разработчикам сконцентрировать свои усилия на критически важных подмножествах требований и функций и итерационно в несколько этапов предоставить последовательные версии высококачественных систем, удовлетворяющих или даже превосходящих основные ожидания пользователей. Привлечение заказчика к итерационному решению проблемы управления масштабом и функциями повышает взаимные обязательства сторон, способствует росту взаимопонимания и доверия между заказчиком и разработчиками. Имея достаточное определение функций продукта (концепцию) и сократив масштаб проекта до разумного уровня, можно надеяться на успех в следующих версиях программного продукта.

При разработке требований к комплексу программ необходимо *выделять и ранжировать по приоритетам заинтересованных*

лиц, которым необходимы определенные функции и показатели качества программного продукта с учетом их специализации и профессиональных интересов. Широкая номенклатура характеристик, представленная в стандарте ISO 9126, определяет разнообразные требования, из которых следует селектировать и выбирать те, которые необходимы *с позиции конкретных лиц - потребителей этих данных*:

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

В таблице 4.1 представлен *пример ранжирования* по степени важности на три уровня (высокая, средняя, низкая) основных стандартизованных характеристик качества комплексов программ для разных категорий специалистов.

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

ся, прежде всего, высокие приоритеты для надежности, эффективности и практичности. Для заказчиков приоритетными могут быть также сопровождаемость и мобильность версий, которые для оперативных пользователей программных продуктов обычно являются второстепенными.

Таблица 4.1

Пример ранжирования важности характеристик комплекса программ для различных категорий специалистов

	Функциональные возможности	Надежность	Эффективность	Практичность	Сопровождаемость	Мобильность
Заказчики	Высокая	Высокая	Высокая	Высокая	Средняя	Средняя
Пользователи	Высокая	Высокая	Высокая	Высокая	Низкая	Низкая
Разработчики	Высокая	Высокая	Средняя	Средняя	Средняя	Средняя
Сопровождающие	Средняя	Средняя	Средняя	Высокая	Высокая	Низкая
Специалисты по внедрению	Высокая	Средняя	Высокая	Высокая	Низкая	Средняя

Последние две группы потребителей интересуют преимущественно характеристики комплексов программ на промежуточных этапах ЖЦ, на которых проявляются, в основном, внутренние структурные, технологические свойства программного продукта, влияющие также на сопровождаемость и мобильность. Их можно не представлять в составе эксплуатационной документации для оперативных пользователей и отражать только в технологической документации разработчиков, специалистов по сопровождению и внедрению программ и данных, а также поставлять заказчикам по специальному запросу.

Приоритеты потребителей при формировании требований к комплексу программ отражаются не только на выделении важнейших для них критериев и ранжировании приоритетов других характеристик, но также на возможности исключении из анализа некоторых характеристик качества, которые для данного потребителя не имеют значения. Представленное в таблице 4.1 ранжирование может детализироваться и изменяться в зависимости от функций комплексов и реальных ресурсов, доступных для обеспечения их жизненного цикла.

Эти приоритеты должны обобщаться и учитываться заказчиком проекта при подготовке контракта и технического задания. После определения назначения и функций подготовка исходных данных и концепции комплекса программ должны завершаться **выделением номенклатуры приоритетных функций и характеристик качества**, имеющих достаточное влияние на функциональную пригодность **для определенных потребителей**. Требования к значениям функций и характеристик программного продукта должны быть предварительно проверены разработчиками на их реализуемость с учетом доступных ресурсов конкретного проекта и при необходимости откорректированы по составу и значениям с учетом рисков. При ограниченности ресурсов проекта распределение приоритетов должно становиться более строгим и могут снижаться приоритеты характеристик, для реализации которых ресурсов недостаточно.

Проверка корректности требований к крупным комплексам программ

В самом начале и в процессе развития проекта необходимо проверять **качество и корректность требований**. Преимущества такого подхода состоят в том, что сводятся к минимуму дорогие переделки за счет уменьшения числа дефектов требований, которые можно обнаружить или предотвратить на ранних этапах жизненного цикла проекта или его версии. Определение требований напрямую связано с процедурами проверки и утверждения (аттестации), как это сформулировано в стандарте **ISO12207**. Принято считать, что требования описаны не полностью, если для них не заданы правила – **проверки и аттестации**, то есть не определены способы контроля корректности и утверждения. Процедуры проверки требований являются отправной точкой **для специалистов по качеству**, непосредственно отвечающих за соответствие получаемого программного продукта предъявляемым к нему требованиям.

Требования должны быть **верифицируемыми**. Требования, которые не могут быть проверены и аттестованы (утверждены) – это всего лишь чьи-то **не обязательные пожелания**. Именно так они будут восприниматься разработчиками, даже в случае их высокой значимости для заказчика и пользователей. Если описанное требование не сопровождается процедурами проверки, то в большинстве случаев го-

вся о недостаточной детализации или неполном описании требования и, соответственно, спецификация требований должна быть отправлена на доработку и, если необходимо, должны быть предприняты дополнительные усилия, направленные на конкретизацию требований. Процедура анализа требований считается выполненной только тогда, когда все требования, включенные в спецификацию, обладают методами оценки соответствия создаваемому программному продукту.

Для каждого функционального и количественного требования необходимо **формировать меру качества**, которую разбить на два класса возможных решений. Те, по которым может достигаться соглашение о том, что они удовлетворяют требованию, и по которым делается вывод, что они не удовлетворяют требованию [9, 25]. Если определить меру качества для каждого требования, то каждое решение, удовлетворяющее этой мере, является **приемлемым для заказчика и/или пользователя**. Прежде всего, надо найти свойство требования, которое предоставит шкалу для измерения. Например, если решено измерять время отклика системы на требование выполнения некоторой функции, то заинтересованные стороны рассматривают функционирование системы и приходят к заключению, что система не будет удовлетворять требованию, если время отклика на запрос пользователя составит больше заданного. Таким образом, время отклика становится критерием качества для данного требования.

При определении критериев качества требований следует выявить и прояснить **нечеткие требования**. Для них в одних случаях можно определить согласованный критерий качества, а в других случаях получается критерий, по которому не достигнуто соглашение с заказчиком. Такое нечеткое требование целесообразно заменить несколькими требованиями, каждое из которых будет иметь свой собственный критерий качества. Не каждое требование может иметь критерий качества, который можно использовать для проверки того, удовлетворяет ли какое-либо решение требованию. Добавив критерий качества для каждого требования, можно сделать их осозаемыми, понятными. Это первый шаг по определению критериев для измерения качества решений.

Когда формулируются требования помимо тех, которые уместны для назначения проекта, есть вероятность, прихватить и несущественные требования, которые часто появляются в результате **непо-**

нимания заинтересованной стороной целей проекта. Если заинтересованной стороне нужно что-то особенное, она полагает, что это является требованием, даже если это несущественно для проекта системы. Чтобы протестировать **требования на значимость**, следует сопоставить требования и сформулированные цели разработки программного продукта:

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

Когда представленные решения для требований ошибочны, часто пропускаются настоящие требования. Также возможное решение не столь хорошее, каким оно могло бы быть, поскольку проектировщик не был свободен в выборе возможных способов для удовлетворения требований. Имея представление о значении, которое придают заинтересованные стороны каждому требованию, можно расставить приоритеты требований к функциям и характеристикам программного продукта. Используя эти знания, можно выстроить приоритеты и компромиссные решения при разработке требований.

Необходимо уметь проверять, что разрабатываемый комплекс программ удовлетворяет каждому из зафиксированных и утвержденных требований. Нужно выделять каждое такое требование, чтобы **проследить его развитие** в ходе подробного анализа, проектирования и, наконец, реализации. Каждый этап разработки комплекса программ формирует, уточняет и реорганизует требования, чтобы сделать их как можно ближе к назначению нового программного продукта.

Каждое требование должно отражать **отдельно распознаваемую, измеряемую сущность**. Необходимо определять и фиксировать связи между требованиями и влияние одних требований на другие. В проектах крупных комплексов программ нужно применять способ работы с большим числом требований и сложными связями между ними. Следует учитывать, что наряду с существованием некоторого числа требований, связанных с одним событием и/или сценарием использования, любое требование может быть связано с другими событиями и/или сценариями использования. Событие и/или сценарий использования требований предоставляют некоторое количество не-

больших минимально взаимосвязанных систем. С помощью конфигурационного управления требованиями можно прослеживать, с какими событиями и/или сценариями использования какие решения связаны. Если в требование вносится изменение, можно определить все части системы, на которые влияет это изменение.

Спецификация требований должна содержать все *требования, которым обязан удовлетворять программный продукт*. В спецификации необходимо объективно определить все, что он должен делать, а также те условия внешней среды, при которых он должен применяться и функционировать. Наиболее ответственный аспект формирования требований – общение со специалистами, которые вносят требования. При наличии согласованного способа формирования требований все заинтересованные стороны могут принимать участие в процессе определения требований. Как только сформулировано хотя бы одно требование, можно приступить к его тестированию на корректность и задавать заинтересованным сторонам подробные вопросы для уточнения и конкретизации тестов. Целесообразно заказчику определять относительную значимость, приоритеты требований, задавать критерий качества для каждого требования и использовать этот критерий для тестирования корректности возможных решений.

Требования к профессиональной квалификации руководителей для эффективного управления производством крупных программных продуктов

Крупные программные продукты размером в сотни тысяч строк создаются большими коллективами специалистов, которые структурируются обычно на группы, разрабатывающие законченные функциональные задачи и могут состоять из нескольких десятков специалистов. Такие группы, в свою очередь, может быть целесообразным делить на более мелкие группы размером в пределах десяти человек. Подобное иерархическое структурирование крупных коллективов в соответствии с архитектурой комплексов программ обеспечивает их *управляемость и возможность детального планирования* развития и контроля состояния проектов (рис. 4.2).

Квалифицированный коллектив руководителей для эффективного управления производством крупных программных продуктов должен уметь оценивать и учитывать:

- квалификацию заказчика при определении требований к характеристикам программного продукта;
- требования к квалификации руководителей для выбора структуры и организации коллектива специалистов для производства программного продукта;
- требования к квалификации руководителей для организации процессов, средств и графиков эффективного производства программного продукта;
- требования к общей квалификации и взаимодействию в коллективе специалистов при производстве программного продукта;
- обобщенный человеческий фактор уровня организации и квалификации производственного коллектива;
- типовую формализованную структуру документов, определяющих базовые требования к конкретному программному продукту.

Рис. 4.2.

Уровень **квалификации заказчика** и корректность требований технического задания к функциям и характеристикам может весьма сильно влиять на выделяемые ресурсы коллектива специалистов для производства конкретного программного продукта. При этом первоначальное техническое задание зачастую оказывается недостаточно квалифицированным и подвергается в дальнейшем многократным изменениям. Этому же может способствовать **различие в квалификации между заказчиком и разработчиком** в уровне понимания целей продукта и необходимых затрат на реализацию основных функциональных требований. Особенно сильно на достоверность технического задания и возрастание затрат может влиять попытка заказчика форсировать сроки разработки версий программного продукта.

Даже квалифицированные заказчики вынуждены иногда корректировать техническое задание на любых этапах проекта, что может влиять на увеличение затрат на 10 – 20%, и изменять требования к квалификации разработчиков. При испытаниях заказчик зачастую обнаруживает, что решаются не совсем те задачи и не

совсем так, как нужно, вследствие чего необходимы: переработка требований и тестирование готовых программ, что отражается большим ущербом вследствие дефектов исходных требований заказчика к продукту.

При создании высококачественных комплексов программ, прежде всего, необходима организация и *тесное взаимодействие представителей заказчика и руководителей проекта*. Взгляды и требования заказчика, в основном, отражаются в функциональных и потребительских характеристиках версий программного продукта. Устремления разработчиков направлены на возможность и способы их реализации с требуемым качеством коллективом специалистов. Эти различия исходных точек зрения на проект приводят к тому, что некоторые неформализованные представления тех и других имеют *зоны неоднозначности и взаимного непонимания* требований к продукту, что может приводить к конфликтам при выборе квалификации коллектива специалистов. Организация четкого взаимодействия и сокращение этих зон неоднозначностей требует проведения определенных мероприятий и контактов по обмену знаниями, взаимному повышению квалификации и обучению. Представители заказчика, участвующие в прогнозировании проекта, должны обучаться формализации автоматизируемых технологических процессов производства, для которых предназначены соответствующие инструментальные средства, и иметь представление об эффективных путях их применения.

Для реализации мероприятий по планированию и управлению жизненным циклом концептуально целостных, крупных программных продуктов и обеспечения их качества необходимы организационные действия системных архитекторов, направленные на *подбор и обучение коллектива специалистов разных категорий и специализаций*. Руководство крупным производством программных продуктов целесообразно осуществлять двум лидерам – менеджерам с различными функциями:

- *менеджер – руководитель проекта* – это специалист, обеспечивающий коммуникацию между заказчиком и коллективом специалистов, его задача – определять и обеспечивать удовлетворение требований заказчика с учетом доступных ресурсов;

- *менеджер-архитектор программного продукта* – должен управлять коммуникациями и взаимоотношениями в большом производственном коллективе, являться координатором последовательно-

сти создания компонентов, разрабатывать базовые, функциональные спецификации требований и управлять ими, вести график проекта и отчитываться за его состояние и развитие, инициировать принятие критичных для хода проекта решений.

Руководителем производства программного продукта в зависимости от особенностей продукта может быть: менеджер продукта, менеджер проектирования, руководитель проекта. **Лидер – руководитель должен уметь** [16, 23, 53]:

- руководить процессом выявления и формирования требований к функциям программного продукта, подлежащим производству; вести переговоры с руководством системы, пользователями и разработчиками компонентов и поддерживать равновесие между тем, чего хочет заказчик, и тем, что может создать команда разработчиков за ресурсы и время, отведенные для их реализации;
- рассматривать конфликтующие пожелания, поступающие от различных участников проекта комплекса программ и находить компромиссы, необходимые для определения приоритетов набора функций, представляющих наибольшую ценность для максимального числа участников проекта, прежде всего, заказчика и пользователей;
- осуществлять проверку спецификаций требований программного комплекса, чтобы удостовериться, что они соответствуют базовой концепции проекта и функций программного продукта;
- осуществлять управление изменением приоритетов задач и функций, а также добавлением и исключением новых функций комплекса программ.

Управление специалистами при производстве включает хранение версий требований к компонентам, отслеживание связей, оценку рисков требований, выстраивание последовательности приоритетов требований. Группа поддержки руководителя должна уметь описать способ управления требованиями к продукту и их реализацией **в плане производства**. Ключевыми соображениями при выборе способа хранения требований должны быть: простота и гибкость сортировки требований и подготовки отчетов, простота и скорость доступа к требованиям, эффективность сопровождения и модификации требований. Группа управления должна составлять график разработки и выполнения процедур, чтобы определить время и ресурсы на эти работы.

При подготовке руководителем графика производства должны:

- назначаться сотрудники, ответственные за выполнение каждой из работ и компонентов;
- учитываться последовательность разработки компонентов, их взаимозависимость и связь с процессами и другими компонентами комплекса программ;
- сокращаться или исключаться возможные конфликты между процедурами производства, которые должны быть документированы;
- процедуры производства должны быть объединены в группы в соответствии с функциями программного продукта;
- разработка производственных процедур и их применение должны быть спланированы таким образом, чтобы исключить дублирование работ;
- в структуре процедур следует учитывать и документировать их приоритеты и риски;

Руководителем производства комплекса программ необходимо уметь описать ***последовательность*** выполнения процедур производства и их ***взаимозависимость***, поскольку определенная функция зачастую не может быть выполнена, пока предыдущая функция не сформирует нужные данные. Необходимо планировать работы по настройке среды, подготовке обязательных отчетов в график разработки и выполнения процедур. График разработки и выполнения производственных процедур должен позволять так организовать действия, чтобы один специалист не мог независимо изменять данные, используемые другим специалистом.

При группировании автоматизированных производственных процедур необходимо по возможности исключать их дублирование. Специалистам нужно изучать Программу и план производства с целью проверки того, как результаты анализа и проектирования для каждого учтены в плане-графике. При составлении графика разработки и выполнения процедур необходимо учитывать ***приоритеты и риски***, присвоенные различным компонентам и функциям комплекса программ. В графике производства специалисты должны уделять особое внимание проверке функций, наиболее значимых для программного продукта, и функций повышенного риска. Такие работы должны выполняться в первую очередь, и в графике следует преду-

сматривать достаточно времени для проверки этих функций и, в случае необходимости, для их регрессионного тестирования.

Руководитель производства должен отслеживать изменения требований и соответственно обновлять график. Имея в виду многообразие возможных изменений, требующих уточнения плана-графика, полезно создавать базовую версию плана-графика и версии при каждом последующем его принципиальном изменении. Нужно уметь документально фиксировать каждое изменение плана-графика с помощью систем их отслеживания. Оригинальный план-график и все последующие изменения его базовых версий должны проходить процедуру утверждения у полномочных руководителей.

Совокупность видов квалификации специалистов и коллектива целесообразно учитывать как **обобщенный человеческий фактор** в процессе производства сложных программных продуктов. На него влияет ряд трудно учитываемых условий. Сложность психологических экспериментов с регистрацией всех основных условий производства и особенностей специалистов приводит к большому разбросу результатов. Вследствие этого некоторые специалисты в основном на опыте разработки относительно **простых небольших программ** утверждают, что при производстве комплексов программ **только человеческий фактор** определяет длительность и трудоемкость разработки. Отсюда делается вывод о принципиальной невозможности планирования и прогнозирования процессов производства программных продуктов. В действительности при крупных разработках **человеческий фактор в коллективах значительно усредняется**. Тем не менее, достаточно часто по этой причине возможен разброс производительности труда в 2 – 3 раза при производстве крупных практически аналогичных продуктов.

По мере повышения квалификации коллектива специалистов и автоматизации творческой части труда следует ожидать асимптотического приближения к потенциальным значениям интегральных характеристик производства новых продуктов. Эти предельные значения определяются психологическими возможностями человека по интенсивности принятия творческих решений. При экономическом обосновании проектов сложных комплексов программ теоретически рассчитать такие характеристики вряд ли возможно, и реальным путем их оценки являются изучение и экстраполяция экспериментальных данных реальных разработок программных продуктов

с учетом **возрастания квалификации специалистов и уровня автоматизации производства**. Эти факторы способствуют эволюционному, относительно медленному приближению к предельным экономическим характеристикам для новых крупных продуктов. Вряд ли можно ожидать в ближайшие годы повышения производительности труда на порядок **для полностью новых, сложных программных продуктов**. Еще более консервативна длительность разработки. Принципиальным путем улучшения экономических характеристик при производстве программных продуктов является сокращение творчества специалистов на тех этапах, где возможны типовые стандартные решения и заготовки, не требующие при их многократном применении высококвалифицированного творческого труда. Основой такого подхода является **сборочное программирование** на базе единой технологии, базы данных готовых апробированных компонентов и стандартизированной архитектуры определенных классов комплексов программ.

При разработке требований к программным продуктам, кроме основных целей, назначения и функций, квалифицированным специалистам важно учесть и сформулировать содержание достаточно полного множества документированных характеристик, каждая из которых может влиять на успех проекта. Для уменьшения вероятности случайного пропуска важного требования **заказчикам и пользователям** целесообразно использовать типовые перечни (шаблоны) наборов требований, которые можно целеустремленно сокращать и адаптировать, обеспечивая **целостность требований в документах для конкретных проектов**. Ниже представлены **примеры** состава требований документов на двух этапах жизненного цикла сложных комплексов программ: на этапе формирования концепции продукта и на этапе детального проектирования комплекса программ.

Состав основных требований к программному продукту:

- описание обобщенных результатов обследования и изучения существующей системы и внешней среды;
- описание целей, назначения программного продукта и потребностей заказчика и потенциальных пользователей к нему в заданной среде применения;
- перечень базовых стандартов для предполагаемого проекта программного продукта;

- общие требования к характеристикам комплекса задач программного продукта:
 - цели создания программного продукта и назначение комплекса функциональных задач;
 - перечень объектов среды применения (технологических объектов управления, подразделений предприятия и т.п.), при управлении которыми должен решаться комплекс задач;
 - периодичность и продолжительность решения комплекса задач;
 - связи и взаимодействие комплекса задач с внешней средой и другими компонентами системы;
 - распределение функций между персоналом, программными и техническими средствами при различных ситуациях решения требуемого комплекса функциональных задач;
- требования к входной информации:
 - источники информации и их идентификаторы;
 - перечень и описание входных сообщений (идентификаторы, формы представления, регламент, сроки и частота поступления);
 - перечень и описание структурных единиц информации входных сообщений или ссылка на документы, содержащие эти данные;
- требования к выходной информации:
 - потребители и назначение выходной информации;
 - перечень и описание выходных сообщений;
 - регламент и периодичность их выдачи;
 - допустимое время задержки решения определенных задач;
- сопоставительный анализ требований заказчика и пользователей к программному продукту и набора функций для удовлетворения требований заказчика и пользователей;
 - обоснование выбора оптимального варианта требований к содержанию и приоритетам функций комплекса;
 - общие требования к структуре, составу компонентов и интерфейсам с внешней средой;
 - ожидаемые результаты и возможная эффективность реализации выбранного варианта требований;

- ориентировочный план реализации выбранного варианта требований;
- общие требования к составу и содержанию документации программного продукта;
- оценка необходимых затрат ресурсов на разработку, ввод в действие и обеспечение функционирования программного продукта;
- предварительный состав требований, гарантирующих качество применения программного продукта;
- предварительные требования к условиям испытаний и приемки системы и программного продукта.

Спецификация требований к системе и к программному продукту на этапе детального проектирования:

- требования проекта системы к программному продукту как к целому в общей архитектуре системы;
 - требования к унификации интерфейсов и базы данных комплекса программ;
- требования и обоснование выбора проектных решений уровня системы, состава компонентов системы, описание функций системы и программного продукта с точки зрения пользователя;
- спецификация требований верхнего уровня комплекса программ, производные требования к компонентам и требования к интерфейсам между системными компонентами, элементами конфигурации программ и аппаратуры;
- описание распределения системных требований по компонентам программного продукта с учетом требований, которые обеспечивают заданные характеристики качества;
 - требования к архитектуре системы, содержащей идентификацию и функции компонентов системы, их назначение, статус разработки, аппаратные и программные ресурсы;
 - требования совместного целостного функционирования компонентов комплекса программ, описание и характеристики их динамических связей;
 - требования для системы или/и подсистем и методы, которые должны быть использованы для гарантии того, что каждое требование к комплексу программ будет выполнено и прослеживаемо к конкретным требованиям системы:
 - к режимам работы;
 - к производительности системы;

- к внешнему и пользовательскому интерфейсу системы;
- к внутреннему интерфейсу компонентов и к внутренним данным системы;
- по возможности адаптации программного продукта к внешней среде;
- по обеспечению безопасности системы, программного продукта и внешней среды;
- по обеспечению защиты, безопасности и секретности данных;
- по ограничениям доступных ресурсов комплекса программ;
- по обучению и уровню квалификации персонала;
- по возможностям средств аттестации результатов и компонентов, включающие в себя демонстрацию, тестирование, анализ, инспекцию и требуемые специальные методы для контроля функций и качества конкретной системы или компонента программного продукта.

Представленный состав спецификации требований на этапе детального проектирования целесообразно использовать руководителям и системным аналитикам как компонент для уточнения технического задания и контракта с заказчиком на программный продукт, а также служить базой для формирования комплекса отчетных требований, утверждаемых и проверяемых заказчиком при приемке готового программного продукта. Состав стандартизованных характеристик качества программных комплексов и процессы выбора требований к ним в конкретных проектах должны быть отдельным, обязательным разделом в общей спецификации требований, итерационно формируемыми на этапах концепции и проектирования комплекса программ и контролируемыми при испытаниях программного продукта.

Лекция 5

ТРЕБОВАНИЯ К ПРОФЕССИОНАЛЬНОЙ КВАЛИФИКАЦИИ РУКОВОДИТЕЛЕЙ И СПЕЦИАЛИСТОВ, ОЦЕНИВАЮЩИХ ЭКОНОМИЧЕСКИЕ ХАРАКТЕРИСТИКИ ПРОИЗВОДСТВА КРУПНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Основные требования к профессиональной квалификации руководителей для оценивания экономических характеристик производства крупных программных продуктов

Интуитивные оценки руководителями и исполнителями – размеров, сложности и трудоемкости конкретных программных проектов, как правило, *отличаются существенными недостатками и ошибками* вследствие того, что [5, 29, 42]:

- человек, в основном, оптимистичен, и каждому представляется, что проект комплекса программ меньше по размеру и более простой, что ведет к первоначальным недооценкам его сложности и к конфликтным ситуациям при разработке;
- человек обычно не полностью использует предыдущий опыт о сложности функций аналогичных проектов и, особенно, о большом размере вспомогательных компонентов комплексов программ, которые также должны быть разработаны;
- отдельные специалисты, как правило, не знакомы со всем размером проекта и пожеланиями пользователей, что приводит к недооценке второстепенных функций и программных компонентов, к отсутствию реалистичного применения накопленных знаний при оценивании размера и сложности проекта.

Следствием этого являются *большие ошибки при планировании экономических характеристик – сроков, трудоемкости и стоимости создания программных продуктов*. Стихийная оценка

характеристик экономики производства сложных комплексов программ в большинстве случаев приводит к значительному запаздыванию завершения разработок и превышению предполагавшихся затрат. Типичны ситуации, когда отставание сроков внедрения промышленных систем управления и обработки информации полностью **зависит от неготовности для них программных продуктов.**

Приступая к разработке сложных программных проектов, квалифицированные заказчики и исполнители, прежде всего, должны **уметь понять целесообразно ли их создание** и оценить, какова будет возможная эффективность применения готового продукта, оправдываются ли затраты на его разработку и использование (рис. 5.1).

Квалифицированные руководители и специалисты должны уметь оценивать и учитывать экономические характеристики производства крупных программных продуктов:

- возможность больших ошибок при интуитивной оценке трудоемкости и длительности производства программных продуктов;
- количественные методы прогнозирования экономических характеристик производства программных продуктов;
- оценку экономической и технической целесообразности и актуальности предполагаемого производства программного продукта;
- опыт и статистику накопленных экономических характеристик производства программных продуктов.

Рис. 5.1

Поэтому такие технические проекты традиционно должны начинаться с анализа и разработки **экономического обоснования** предстоящего жизненного цикла и применения предполагаемого продукта. Заказчику проекта необходимо оценивать реальную потребность в его создании и возможную конкурентоспособность, а потенциальному разработчику предполагаемого продукта проводить оценку реализуемости проекта в условиях и ресурсах, предлагаемых заказчиком. **Руководители конкретных проектов зачастую не достаточно квалифицированы**, чтобы обоснованно определять, сколько времени и затрат труда потребуется на каждый этап программной части проекта системы, и не могут оценивать, насколько успешно выполняется план производства. Это, как правило, означает, что программная часть проекта системы с самого начала **выходит из-под**

экономического контроля и возможна катастрофа с реализацией и завершением проекта всей системы в требуемый срок с заданным качеством. Это привело к появлению новой области экономической науки и практики – **экономики производственных процессов и жизненного цикла программных продуктов**. Ее основной задачей является квалифицированное прогнозирование, эффективное управление, распределение ресурсов и экономное использование необходимых быстро возрастающих капиталовложений в производство сложных комплексов программ высокого качества и различного назначения.

Применение комплексов программ как производственной продукции существенно повысило **актуальность экономического обоснования и необходимость прогнозирования и измерения** их характеристик качества и процессов производства. Основной целью производства многих программных продуктов является повышение эффективности промышленных систем обработки информации и/или управления объектами, в которых применяются сложные комплексы программ. Такими системами могут быть средства автоматизированного управления прокатными станами, самолетами, системами вооружения или электростанциями, информационно-справочные системы административного управления, системы автоматизации проектирования и обучения и т.п. В ряде случаев программные продукты невозможно или очень трудно характеризовать непосредственной **экономической эффективностью**. В таких случаях при анализе программ невозможно определять прямую экономическую эффективность систем в зависимости от затрат ресурсов, и целесообразно из анализа исключать характеристики полной экономической эффективности программных продуктов и сопутствующие ей функциональные критерии качества. В результате исследование эффективности создания комплексов программ приходится проводить по величине затрат на производство программного продукта в предположении, что полностью обеспечена реализация заданных их функциональных характеристик с требуемым качеством. Важной особенностью крупных программных продуктов является их **отчуждаемость от первичных производителей**. Это обуславливает необходимость производить такие программы с соблюдением определенных стандартов, правил структурного построения, а также оформлять на такие программы достаточно полную документацию, позволяющую применять, сопровождать и развивать программные продукты.

Экономическое обоснование проектов квалифицированными заказчиками и производителями на начальном этапе их развития должно содержать оценки рисков реализации поставленных целей, обеспечивать возможность планирования и выполнения жизненного цикла программного продукта или указывать на недопустимо высокий риск его реализации и целесообразность прекращения разработки. Большую часть рисков и негативных последствий производства можно избежать, освоив и используя существующие **методы оценивания и прогнозирования производственных затрат**, а также управления проектами программных продуктов для их успешного завершения.

Для заказчика и разработчиков при заключении контракта необходимо умение квалифицированно выполнять достаточно достоверное прогнозирование требований к программному продукту и **экономическое обоснование необходимых ресурсов** по трудоемкости, стоимости, срокам и другим показателям. Заказчик заинтересован в получении продукта высокого качества при минимальных затратах, а разработчик желает получить максимальную оплату за созданный продукт и достаточные ресурсы на его производство. Противоположность интересов поставщика и потребителя при оценке экономических характеристик, стоимости и других ресурсов проекта требует поиска компромисса, при котором производитель программного продукта не продешевит, а заказчик не переплатит за конкретные выполненные работы и весь проект. Поэтому оба партнера **заинтересованы в квалифицированном экономическом прогнозировании** и обосновании проектирования и производства программного продукта.

Во многих случаях эффективность систем новой техники и программных продуктов в процессе производства приходится прогнозировать **в условиях неопределенности** целей, различных факторов и характеристик заказанных систем и продуктов. Обычно бывают недостаточно известны перспективы внедрения и применения объектов производства – новых программных продуктов. Трудно формализуемыми и оцениваемыми являются размеры (масштабы) и структура систем, взаимодействие основных подсистем, цели, функции и критерии оценки эффективности их функционирования. Значительные неопределенности содержатся также в экономических характеристиках производственных технологий, а также инструментальных средств автоматизации проектирования и изготовления комплексов программ.

При разработке комплексов программ сложно переплетаются содержание, этапы и распределение работ, возможен ряд возвратов на более ранние технологические этапы в процессе создания компонентов, они имеют не совсем определенные границы начала и завершения.

Методы и достоверность *экономического анализа производства* и жизненного цикла крупных комплексов программ можно разделить на *две части*, существенно различающиеся особенностями производственных процессов, экономическими характеристиками и влияющими на них факторами. *В первой части ЖЦ* производятся: системный анализ, проектирование, разработка, тестирование и испытания первой (базовой) версии программного продукта. Номенклатура работ, их трудоемкость, длительность и другие характеристики на этих этапах ЖЦ существенно зависят от свойств создаваемого продукта, требуемых показателей качества, внешней и технологической среды разработки. При этом *первоочередное значение* имеют размеры, архитектура, сложность функций, количество, состав и взаимосвязи компонентов комплекса программ, которые являются базой для оценивания экономических характеристик конкретного продукта. Изучение подобных зависимостей для различных прототипов программных продуктов позволяет достаточно достоверно прогнозировать состав и основные экономические характеристики производства, планы и графики работ для вновь создаваемых продуктов.

Вторая часть ЖЦ, отражающая применение, сопровождение и развитие версий программного продукта, связана не только с экономическими характеристиками продукта и среды производства. При этом экономические характеристики зависят *от величин и интенсивности изменений* версий, сложности и стоимости каждой модернизации программного продукта. Версии сложных программных продуктов обычно характеризуются *длительной непрерывной эксплуатацией*, продолжительность которой значительно превышает длительность производства первой версии. *В процессе сопровождения* программы могут подвергаться эпизодическим корректировкам, которые должны регистрироваться, накапливаться и передаваться пользователям экземпляров системы. Необходимо обеспечивать адекватность документации каждой версии эксплуатируемого продукта в любой момент времени.

Номенклатура работ на этапах сопровождения более или менее стабильная, а их трудоемкость и длительность могут сильно варьироваться и зависят от массовости и других факторов распространения и применения конкретного программного продукта. Успех продукта у пользователей и на рынке, а также **экономику процесса развития версий трудно предсказать**, и она связана не только с техническими и экономическими параметрами производства версий комплекса программ. Определяющими становились потребительские характеристики и качество продукта, а его экономические особенности с позиции производства отходят на второй план. В результате планирование трудоемкости, длительности и числа специалистов для этих этапов трудно обобщать, и приходится производить итерационно на **базе накопления опыта и анализа развития экономических характеристик конкретных типов и версий программных продуктов**.

Оценивание требуемых экономических характеристик производства программного продукта с учетом профессиональной квалификации специалистов

Экономические характеристики производства – стоимость, трудоемкость, длительность комплексов программ, в **основном определяет человеческий фактор** – количество, квалификация и организация коллектива специалистов. Эти экономические характеристики обычно являются исходными требованиями, выдаваемыми заказчиком как доступные, ограниченные ресурсы для разработчиков. Для обеспечения производства программного продукта в таких заданных условиях необходим коллектив специалистов, имеющих соответствующую профессиональную квалификацию. Это приводит к необходимости оценивания **требований к коллективу специалистов**, необходимого для реализации определенного программного продукта, исходя из **заданных заказчиком экономических характеристик**. Альтернативой обычно является оценка возможных экономических характеристик производства требуемого программного продукта в зависимости от человеческих факторов коллектива предприятия, инструментальной и внешней среды производства (рис. 5.2).

Квалифицированные руководители и специалисты должны уметь оценивать экономические характеристики производства крупных программных продуктов, включая:

- анализ, выделение и оценивание факторов, определяющих экономические характеристики производства программного продукта;
- анализ, сбор и обобщение рейтингов для расчета экономических характеристик производства предполагаемого программного продукта;
- корректировку требований к количеству и квалификации коллектива специалистов для удовлетворения требований к экономическим характеристикам программного продукта;
- корректировку требований к уровню зрелости технологического обеспечения производства программного продукта;
- достижение рационального баланса требований к количеству и квалификации коллектива специалистов и требований к экономическим характеристикам производства программного продукта;
- расчет трудоемкости, длительности и числа квалифицированных специалистов, необходимых для производства программного продукта;
- контролируемое отслеживание экономических характеристик производства и изменений комплекса программ с учетом квалификации коллектива специалистов, затрат ресурсов и ограничений сроков.

Рис. 5.2

Коллективы с наилучшими экономическими характеристиками могут служить ориентирами достижимых в ближайшие годы значений экономических характеристик производства для соответствующих классов программных продуктов. Однако неуклонно повышается сложность комплексов программ, что вызывает возрастание затрат творческого труда на единицу размера продукта. Исследования экономических характеристик реальных производственных процессов создания крупных программных продуктов позволили выделить ряд факторов и характеристик специалистов в наибольшей степени, отражающихся на эффективности производства комплексов программ [5, 29, 54].

Накопленный опыт производства и обобщение проведенных исследований позволили выделить и детализировать ***четыре основные группы факторов***, влияющих на экономические характеристики

при непосредственном проектировании и производстве программных продуктов – рис. 5.3:

- факторы, отражающие особенности создаваемого комплекса программ **как объекта** проектирования и производства и требования к его функциональным характеристикам и качеству;
- факторы, определяющие **организацию процесса** производства программного продукта и его обеспечение коллективом квалифицированных специалистов;
- факторы, характеризующие **технологическую среду** и оснащенность специалистов инструментальными средствами автоматизации процесса производства программного продукта;
- факторы, отражающие оснащенность процесса производства программного продукта и специалистов **аппаратурными вычислительными средствами**, на которых реализуется программный продукт и базируются инструментальные системы автоматизации разработки.

Рекомендации и требования к квалификации руководителей и специалистов в учебнике отражаются, в основном, **номенклатурой** свойств, оценивания и умения без детализации значений этих характеристик. В модели СОСМО II большинство факторов рекомендуется учитывать несколькими градациями **уровня качества (5 – 6) – рейтинга каждой характеристики**. В частности, показатели уровня профессиональной квалификации специалистов предлагается определять длительностью работы по этой специальности или описаниями уровней конкретных свойств, которые выбираются руководителем конкретного проекта комплекса программ. Каждому уровню в таблицах представлены коэффициенты (рейтинги) влияния квалификации на трудоемкость и длительность проектирования и производства программного продукта. Эти рейтинги далее используются в выражениях для расчета экономических характеристик и могут применяться в требованиях и рекомендациях при детализации необходимых свойств специалистов.

Перечисленные группы факторов (кроме первой), **зависят от организации и квалификации коллектива специалистов** и должны учитываться при формировании требований к экономическим характеристикам. Для определения этих требований **квалифицированные специалисты должны уметь использовать итерации**:

- предварительно определять экономические характеристики производства при требовании средней квалификации всех специалистов коллектива (средние значения рейтингов);
- улучшать экономические характеристики путем совершенствования требований к отдельным реально достижимым характеристикам коллектива специалистов (доступные изменения рейтингов).



Рис.5.3

Наиболее известные современные методы оценивания экономических характеристик производства программных продуктов разработаны в [54] и представлены в модели СОСОМО II. Для выполнения *оценок трудоемкости производства программных продуктов* (человеко-месяцы) в этой модели рекомендуются выражения:

$$C = A \times \Pi^E \times \prod_{i=1}^n M(i), \quad (5.1)$$

5

где: $A = 2,94$; $E = B + 0,01 \times \sum_{j=1}^5 F(j)$; $B = 0,91$.

В модели **СОСОМО II** влияние на трудоемкость производства программного продукта учитываются 22 фактора. Пять – **масштабные факторы**, характеризуются суммой $F(j)$ в значении **степени размера** Π комплекса программ, а 17 множителей $M(i)$ непосредственно влияют на изменение трудоемкости производства программного продукта – таблица 5.1.

Факторы масштабирования $F(j)$ в уравнении оценки трудозатрат в модели СОСОМО II варьируются и применяются с целью выделения и обобщения основных параметров при прогнозировании производства. При использовании представленных выражений для прогнозирования экономических характеристик конкретных проектов специалистам следует выбирать набор факторов (**калибровать модель**), имеющих значения коэффициентов изменения трудоемкости (КИТ) $F(j)$ в соответствии с характеристиками факторов реального производства конкретного продукта.

Номинальными (средними) в выражении (5.1) для предварительных оценок экономических характеристик все значения могут приниматься $M(i) = 1,00$, при которых соответствующий фактор не влияет на трудоемкость производства программного продукта. Для выбора значений $M(i)$ в соответствии с реальными **характеристиками конкретного продукта, среды разработки и коллектива специалистов** следует использовать таблицы рейтингов трудоемкости, представленные по выделенным четырем группам (см. рис.5.3) в [54, 35]. В модели СОСОМО II поддерживаются вероятностные диапазоны оценок рейтингов, представляющие одно стандартное отклонение на фоне наиболее благоприятных или средних оценок.

Таблица 5.1
Состав факторов модели СОСМО II

	<i>Содержание фактора</i>
F1 F2 F3 F4 F5	<p>Масштабные факторы</p> <p>Новизна проекта</p> <p>Согласованность с требованиями и интерфейсами</p> <p>Управление рисками и архитектурой проекта</p> <p>Слаженность работы коллектива</p> <p>Технологическая зрелость обеспечения производства</p>
M1 M2 M3 M4 M5	<p>Факторы, влияющие на затраты производства</p> <p>Требования и характеристики объекта производства</p> <p>Надежность функционирования</p> <p>Размер базы данных</p> <p>Сложность функций и структуры</p> <p>Требование повторного использования компонентов</p> <p>Полнота и соответствие документации проекта</p>
M9 M10 M11 M12 M13 M14	<p>Характеристики коллектива специалистов</p> <p>Квалификация аналитиков</p> <p>Квалификация программистов</p> <p>Стабильность коллектива</p> <p>Опыт работы по тематике проекта</p> <p>Опыт работы в инструментальной среде</p> <p>Опыт работы с языками программирования</p>
M15 M16 M17	<p>Технологическая среда производства</p> <p>Уровень инструментальной поддержки проекта</p> <p>Необходимость распределенной разработки проекта</p> <p>Ограничения длительности производства продукта</p>
M6 M7 M8	<p>Аппаратурно-вычислительная среда производства</p> <p>Ограниченность времени исполнения программ</p> <p>Ограниченность доступной оперативной памяти</p> <p>Изменчивость виртуальной среды разработки проекта</p>

При использовании представленных выражений для прогнозирования экономических характеристик конкретных проектов следует выбирать набор факторов (**калибровать модель**), имеющих наи-

большие значения коэффициентов изменения трудоемкости (КИТ) $F(j)$ и $M(i)$ в соответствии с *характеристиками конкретного продукта, среды разработки и коллектива специалистов*.

Для прогнозирования длительности производства программных продуктов в модели СОСОМО II детально рассмотрены исходные данные на основе анализа трудоемкости его производства C .

При этом рекомендуется учитывать те же наборы факторов, как на рис. 5.3:

$$T = G \times C^H . \quad (5.2)$$

где: $G = 3,67$;

$$H = D + 0,02 \times 0,01 \times \sum_{j=1}^5 F(j) = D + 0,02 \times (E - B); \quad D = 0,28.$$

Максимальные величины каждого из КИТ производства программных продуктов экспериментально оценены в [5, 29, 54] при предположении, что остальные параметры зафиксированы. В действительности многие факторы взаимно *коррелированы*. Так, например, высокой сложности комплекса программ обычно сопутствует требование высокой безопасности и надежности функционирования, а также длительная эксплуатация. Ряд факторов влияет одновременно на несколько составляющих. Воздействие в процессе производства на такие факторы и субъективный акцент на сокращение определенных видов затрат в некоторых случаях может оказываться нерентабельным с позиции снижения полных затрат на производство программного продукта. Стремление уменьшить затраты в период производства без учета последующего использования продукта, его компонентов и всего жизненного цикла может оказаться мало полезным, а в некоторых случаях привести к значительному увеличению совокупных затрат. Поэтому *необходим системный анализ* распределения и использования ресурсов на производство конкретных комплексов программ путем выбора квалифицированными специалистами значений КИТ (калибровка модели) с учетом всего их жизненного цикла.

Длительность разработки программных продуктов является важнейшей экономической характеристикой, поскольку часто она определяет общие сроки разработки систем, а значит быстроту реализации идей в различных областях и системах автоматизации. При определении коэффициентов за начало разработки комплекса программ принят момент создания технического задания (ТЗ), а за

окончание — завершение испытаний программного продукта в целом или момент предъявления его на испытания. Диапазону размеров современных программных продуктов в три-четыре порядка (до 10 млн. строк) соответствует приблизительно такой же диапазон (три-четыре порядка до тысячи человеко-лет) изменения трудоемкости и стоимости их разработок. Однако очевидна принципиальная **нерентабельность разработки** даже очень сложных продуктов более 5 лет. С другой стороны, программы даже в несколько тысяч строк по полному технологическому циклу с испытаниями и поставкой как продукции редко создаются за время меньшее, чем полгода-год. Таким образом, вариация длительностей производства программных продуктов меньше, чем вариация их трудоемкости, и не превышает десятикратный диапазон от нескольких месяцев до 4 – 5 лет. Длительности разработок **T** ограничены сверху и снизу, и одним из основных факторов, определяющих эти границы, является масштаб комплекса программ – **P**.

Зависимости **T** от размера программ **P** значительно различаются для классов комплексов программ. Это определяется различием сложности классов программ, применяемых языков программирования и единиц измерения размера ПС, следствием чего является различие значений размера созданных программ при одной и той же длительности и трудоемкости разработки. Чтобы сократить ошибки, связанные с неопределенным измерением размера программ, исследована **зависимость длительности разработки от ее трудоемкости**. При этом учитывалась только трудоемкость непосредственной разработки программ **C** без затрат на средства автоматизации разработки. Установлено, что длительность производства комплексов программ меньше подвергается изменениям при автоматизации разработки или другими методами, чем трудоемкость или производительность труда. Необходимость выполнения при производстве определенной совокупности этапов и операций в заданной технологической последовательности остается более или менее постоянной при различных воздействиях на процессы разработки.

Оценка требуемого среднего числа специалистов для конкретного проекта комплекса программ предварительно может быть рассчитана путем деления оценки величины трудоемкости разработки (5.1) на длительность его производства (5.2):

$$N = C / T . \quad (5.3)$$

Однако рациональное число специалистов, участвующих в проекте распределяется не равномерно по этапам работ. Поэтому целесообразно определять число и квалификацию необходимых специалистов с учетом этапов разработки комплексов программ.

Средняя производительность труда коллектива специалистов при разработке сложного полностью нового комплекса программ $P = \Pi/C$ может служить **ориентиром** для сравнения эффективности труда при создании различных продуктов и на разных предприятиях. Эта характеристика, конечно, различается в зависимости от классов, размеров и других параметров комплексов программ, однако диапазон этих различий не столь велик как изменения размера, требований к качеству и других параметров. Так, при диапазоне изменения размеров программ реального времени на четыре порядка средняя производительность труда изменяется только в два раза, что в ряде случаев существенно облегчает упрощенные оценки и прогнозирование экономических характеристик.

Большую роль в повышении экономической эффективности производства программных продуктов играет **повторное использование готовых программных компонентов** (ПИК) из других проектов. При этом значительно сокращаются этапы программирования и автономной отладки модулей и компонентов программ, а также в той или иной степени длительность других этапов. Поэтому зависимость T от доли ПИК оказывается нелинейной, и заметное сокращение длительности разработки проявляется только при создании базовой версии программного продукта практически полностью из готовых компонентов. Для оценки и учета сокращения трудоемкости и длительности производства за счет повторного использования готовых компонентов в модели СОСОМО II [54] рекомендуются соответствующие дополнительные выражения и процедуры. При прогнозировании экономических характеристик предлагается в качестве дополнительных данных рассчитывать и учитывать долю (%) затрат на программный продукт, которые могут быть сокращены за счет применения модифицируемых и/или полностью готовых компонентов.

В модели **СОСОМО II** большое влияние имеют **масштабные факторы прогнозирования** $F(j)$. Выбирать и учитывать следует те факторы, коэффициенты влияния которых на трудоемкость в конкретном проекте имеют достаточную величину, сбалансированную с точностью определения размера комплекса программ или превышают

ее. В группу **факторов – F (j)** следует включать те, которые способны изменять трудоемкость в несколько (до 3 – 5) раз – таблицы 5.2 – 5.3:

- новизну проекта комплекса программ;
- необходимую степень согласованности проекта с требованиями технического задания;
- наличие управления рисками и архитектурой проекта;
- уровень обобщенной слаженности и организованности коллективной разработки проекта;
- уровень обеспечения и оснащения технологии производства по оценке модели СММI.

В модели **СОСОМО II** выделены и оценены рейтинги наиболее значительных интегральных факторов программных продуктов, влияющих на трудоемкость разработки – новизна проекта и согласованность требований заказчика [9, 25, 29]. **Новизну проекта F1** руководителям рекомендуется учитывать степенью формализации целей и функций программного продукта, наличием опыта аналогичных разработок, а также необходимости создания новых компонентов, алгоритмов и архитектуры обработки данных. Характеристики этого фактора достаточно подробно комментированы в таблице 5.2, что позволяет корректно выбирать адекватные значения рейтингов в таблице 5.3.

Согласованность (жесткость) **F2** продукта определена как необходимость и степень обеспечения при производстве соответствия заданным требованиям и спецификациям заказчика к компонентам, внешним интерфейсам и всему комплексу программ. Для учета этого фактора руководителям производства продукта предложены рейтинги соответствующих коэффициентов изменения трудоемкости разработки (см. табл. 5.3).

Наличие управления рисками F3 и архитектурой проекта трудно формализовать содержательно и адекватными значениями рейтингов. Поэтому далее этот фактор не рассматривается.

В модели **СОСОМО II** значительное внимание уделено **влиянию организации и взаимодействия коллектива разработчиков F4** на трудоемкость создания сложных программных продуктов. Этот фактор подробно комментируется его составляющими в таблице 5.4.

Таблица 5.2

Масштабные факторы требований к объектам разработки СОСОМО II

Факторы	Характеристики факторов					
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий	Сверх высокий
Новизна проекта F1	Полностью новый	Во многом новый	Частично новый	В основном известный	Значительно известный	Полностью известный
Согласованность требований F2	Строгая согласованность	Допускаются компромиссы	Значительная	Относительная	Незначительная	При необходимости
Коллективность F4	Некоторое взаимодействие	Сложное взаимодействие	Зачастую коллективная работа	Высокая степень взаимодействия	Непрерывное взаимодействие	Активное взаимодействие
Зрелость производства F5	CMMI Уровень 1	CMMI Уровень 1	CMMI Уровень 2	CMMI Уровень 3	CMMI Уровень 4	CMMI Уровень 5

Таблица 5.3

Рейтинги масштабных факторов объектов разработки СОСОМО II

Факторы	Рейтинги факторов					
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий	Сверх высокий
Новизна проекта F1	6,20	4,96	3,72	2,48	1,24	0,00
Согласованность требований F2	5,07	4,05	3,04	2,03	1,01	0,00
Коллективность F4	5,48	4,38	3,29	2,19	1,10	0,00
Зрелость производства F5	7,8	6,24	4,68	3,12	1,56	0,00

В составе организационных характеристик коллектива рекомендуется учитывать согласованность целей специалистов, участвующих в проекте, их психологическую совместимость и способность к дружной коллективной работе, наличие опыта работы в данном коллективе и другие объективные и субъективные свойства участников проекта. При этом большое значение может иметь личная мотивация и психологические особенности поведения разных специалистов при комплексной работе над производством крупного программного продукта.

Эти характеристики обобщены в качественный показатель ***влияния коллективности – сложности взаимодействия специалистов*** в коллективе, которому сопоставлены рейтинги изменения трудоемкости производства продукта (строка в табл. 5.4). Наилучшим считается непрерывное корректное взаимодействие организованных специалистов с большим опытом работы в данном коллективе при полной согласованности их целей, планов и методов работы.

В остальных случаях в той или иной степени (даже в 3 – 5 раз) может возрастать трудоемкость производства продукта, что нельзя не учитывать при прогнозировании экономических характеристик и обосновании производства крупных продуктов.

В модели ***СОСОМО II*** ***оценивание экономических характеристик от уровня производства – F5*** рекомендуется проводить на основе ***методологии СММІ – системы и модели оценки зрелости*** комплекса, применяемых технологических процессов жизненного цикла [20, 42, 51] (подробнее см. лекцию 11 и рис. 11.5). Методология обеспечения качества программных продуктов поддержана рядом методических документов, а также формализована международными стандартами [28, 58, 64]. ***При экономическом обосновании программных продуктов*** для достижения впоследствии устойчивых результатов в процессе развития технологии и организации управления жизненным циклом комплексов программ рекомендуется использовать эволюционный путь, который позволяет совершенствовать, постепенно снижать затраты и повышать качество процессов производства и продуктов. Методология СММІ рекомендует большой комплекс процессов, который предприятие должно выполнять для приобретения, поставки, производства, использования и сопровождения крупных комплексов программ, и виды деятельности, характеризующие степень технологической зрелости этих процессов.

Таблица 5.4

**Характеристики и влияние коллективизма разработчиков
программных средств на трудоемкость**

Коллективизм	Значения характеристик				
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий
Согласованность целей коллектива	Минимальная	Незначительная	Относительная	Значительная	Полная
Способность членов коллектива адаптироваться к целям других	Малая	Незначительная	Относительная	Значительная	Полная
Опыт работы в составе данного коллектива	Нет	Малый	Незначительный	Значительный	Большой
Степень доверия и взаимодействия в коллективе	Нет	В малой степени	В некоторой степени	Значительная	Большая
Обобщенная коллективность работ	Некоторое взаимодействие в коллективе	Сложное взаимодействие	Зачастую коллективная работа	Высокая степень взаимодействия	Непрерывное взаимодействие
Обобщенный коэффициент влияния коллективности F4	5,48	4,38	3,29	2,19	1,56

В модели СОСМО II приводятся количественные рекомендации **коэффициентов влияния уровней зрелости СММ на трудоемкость**.

кость реализации сложных программных продуктов – таблицы 5.2 – 5.3. Рейтингам технологической зрелости производства в детальной модели СОСОМО II сопоставлены уровни СММI, для каждого из которых приводятся коэффициенты изменения трудоемкости разработки. Применение совершенных технологий и инструментальных средств высшего уровня зрелости может снизить трудоемкость производства сложных продуктов в 3 – 5 раз по сравнению с наиболее широко применяемыми технологиями второго – третьего уровня зрелости СММI. Независимую оценку уровня СММI, выбранной технологии производства сложного ПС, можно использовать как *ориентир при прогнозировании и экономическом обосновании производства* предполагаемого продукта.

Факторы $M(i)$ специалистам следует выбирать из совокупности перечисленных в таблице 5.1, которые в конкретном проекте могут повлиять на изменение трудоемкости разработки на 10 – 20%, соизмеримое с точностью оценок размера комплекса программ. Таких факторов практически может быть выделено до 10 из 17-и, и влияние каждого целесообразно рассматривать и учитывать при оценках, если он способен изменить трудоемкости разработки конкретного проекта на 5 – 10%. Анализ, выбор и оценивание коэффициентов влияния $M(i)$, этих дополнительных факторов довольно сложный процесс. Он оправдан, когда совместное влияние совокупности этих дополнительных факторов может заметно изменить оценки трудоемкости. В результате расчет трудоемкости несколько усложняется, однако процессы последующего *расчета длительности разработки и необходимого числа специалистов практически не изменяются*.

В *СОСОМО II* изложены экспертные оценки для учета влияния *квалификации и характеристик различных категорий специалистов* на экономические характеристики производства программного продукта [54], которые подробно рассмотрены ниже. Аналогичные оценки учета особенностей специалистов и их влияния на экономические характеристики базируются на отечественном опыте создания крупных программных продуктов и представлены в [26]. Эти характеристики сводятся в основном к длительности и опыту работы в определенной области, и к экспертной оценке квалификации специалистов. Выбор и использование их значений при прогнозировании экономических характеристик требует высокой квалификации экспертов и детального знания особенностей коллектива специалистов и среды

производства реального продукта. Для обеспечения возможности их использования при прогнозировании трудоемкости и длительности разработки сложных комплексов программ в модели СОСОМО II представлены соответствующие относительные рейтинги влияния ряда факторов проектов комплексов программ, и в частности, характеристик специалистов. Эти данные *отражают квалификацию специалистов* по степени их влияния на экономические характеристики производства программных продуктов и тем самым *уточняют требования к квалификации коллектива специалистов*. При применении этих рейтингов следует учитывать, что некоторые из них взаимосвязаны, и целесообразно анализировать их корреляцию, возможность объединения или исключения при оценке реальных проектов и коллективов специалистов – рис. 5.2.

В модели **СОСОМО II** особое внимание уделено *влиянию характеристик, организации и взаимодействия коллектива разработчиков F4* на трудоемкость создания сложных программных продуктов. Этот фактор подробно комментируется его составляющими в таблице 5.4. В составе *организационных характеристик коллектива рекомендуется учитывать*:

- согласованность целей специалистов, участвующих в проекте;
- их психологическую совместимость;
- способность членов коллектива адаптироваться к целям других специалистов, к дружной коллективной работе;
- наличие опыта работы в составе данного коллектива;
- степень доверия и взаимодействия в коллективе и некоторые субъективные свойства участников проекта.

При этом большое значение может иметь личная мотивация и *психологические особенности поведения разных специалистов* – обобщенная коллективность работ над производством сложного программного продукта. Эти характеристики обобщены в качественный показатель *влияния коллективности – сложности взаимодействия специалистов* в коллективе, которому сопоставлены рейтинги изменения трудоемкости производства продукта. Наилучшим считается непрерывное корректное взаимодействие организованных специалистов с большим опытом работы в данном коллективе при полной согласованности их целей, планов и методов работы. В остальных случаях в той или иной степени (даже в 3 – 5 раз) может

возрастать трудоемкость производства продукта, что нельзя не учитывать при прогнозировании экономических характеристик и обосновании производства крупных программных продуктов.

Среди ряда *характеристик коллектива разработчиков* наибольшее влияние на трудоемкость оказывает тематическая и технологическая квалификация специалистов, которые в таблицах 5.5 – 5.6 представлены шестью факторами.

Таблица 5.5.

Характеристики разработчиков программного продукта

Характеристика	Значение характеристики				
	Очень низкая	Низкая	Номинальная	Высокая	Очень высокая
Квалификация аналитиков М9	15%	35%	55%	75%	90%
Квалификация программистов М10	15%	35%	55%	75%	90%
Тематический опыт М12	2 мес.	6 мес.	1 год	3 года	6 лет
Инструментальный опыт М13	2 мес.	6 мес.	1 год	3 года	6 лет
Опыт работы с языками М14	2 мес.	6 мес.	1 год	3 года	6 лет
Стабильность коллектива М11	48%	24%	12%	6%	3%
Ограничения сроков разработки М17	75%	85%	100%	130%	160%

Каждый фактор в таблице 5.5 отражен качественными характеристиками, которым соответствуют численные значения рейтингов относительного влияния на экономические характеристики в выражениях 5.1 и 5.2. Совместно эти факторы могут изменять трудоемкость производства программных продуктов на 30 – 40%. Кроме того, в модели СОСОМО II выделен и обобщен на основе пяти характери-

стик коэффициент – комплексная коллективность участников проекта (см. табл. 5.4), влияние которого может достигать пятикратного изменения трудоемкости.

Таблица 5.6

Рейтинги характеристик разработчиков программного продукта

Характеристика	Рейтинги характеристик				
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий
Квалификация аналитиков М9	1,42	1,19	1,00	0,85	0,71
Квалификация программистов М10	1,34	1,15	1,00	0,88	0,76
Тематический опыт М12	1,22	1,10	1,00	0,88	0,81
Инструментальный опыт М13	1,19	1,09	1,00	0,91	0,85
Опыт работы с языками М14	1,20	1,09	1,00	0,91	0,84
Стабильность коллектива М11	1,29	1,12	1,00	0,90	0,81
Ограничения сроков разработки М17	1,43	1,14	1,00	1,00	1,00

Затраты и труд специалистов при реализации крупного продукта традиционно принято распределять по **двум категориям специалистов**: разрабатывающим компоненты и комплекс в целом и обеспечивающим производственную технологию и качество программного продукта.

Организационное разделение специалистов, осуществляющих **производство программного продукта** (первая категория), и **специалистов – технологов**, обеспечивающих, контролирующих и управляющих его качеством в процессе производства (вторая категория), должно обеспечивать эффективное достижение заданных характеристик, а также независимый, достоверный контроль затрат и качества результатов производства.

Специалисты первой категории непосредственно создают компоненты и комплекс программ в целом с заданными показателями качества (см. табл. 5.5 – факторы **M10, M12, M13, M14**). Однако не все перечисленные ниже категории специалистов отражены в представленных таблицах. В процессе производства их функции заключаются в тщательном соблюдении принятой в фирме технологии и в формировании всех предписанных руководствами исходных и отчетных документов.

При этом предполагается, что выбранная технология способна обеспечить необходимые значения конструктивных показателей качества, а достижение заданных функциональных характеристик гарантируется **тематической квалификацией** соответствующих специалистов и регулярным контролем этих характеристик в процессе производства. Система стандартизированного документирования частных работ должна обеспечить объективное отражение качества компонентов и процессов их создания на всех этапах ЖЦ комплекса программ [32, 42, 67].

Разделение труда специалистов этой категории в крупных коллективах приводит к необходимости их дифференциации **по профессиональной квалификации и областям деятельности**:

- **разработчики программных компонентов – программисты – M10** создают компоненты, удовлетворяющие спецификациям требований, реализуют возможности продукта, отслеживают и исправляют ошибки при производстве сложных программных компонентов, что требует детального знания языков программирования, визуального программирования, сетевых технологий и проектирования баз данных – **M13, M14** в таблице 5.6;

- **тестировщики** обеспечивают проверку функциональных спецификаций требований, средств обеспечения производительности, пользовательских интерфейсов, разрабатывают стратегию, планы и выполняют тестирование на соответствие требованиям для каждого компонента и комплекса программ [3, 4, 34];

- **системные интеграторы** сложных проблемно-ориентированных комплексов программ работают над продуктом в значительной степени отличными от программистов методами, на разных языках производства, используют различные средства автоматизации производства и имеют на выходе различные результаты: крупные компоненты и комплексы программ;

- **специалисты, управляющие сопровождением и конфигурацией**, отвечают за снижение затрат на модификацию и сопровождение программного продукта, обеспечение максимальной эффективности взаимодействия компонентов и производство версий программного продукта, принимают участие в обсуждениях интерфейсов и архитектуры продукта;
- **документаторы** процессов и объектов жизненного цикла комплекса программ обеспечивают подготовку и издание сводных технологических и эксплуатационных документов на программный продукт в соответствии с требованиями стандартов.

Успех и качество при производстве крупных программных продуктов зависит от слаженности работы и профессионализма коллектива специалистов на всех этапах и уровнях производства продуктов – **от стабильности коллектива – М11** (табл. 5.5 и 5.6). Особенno важна не индивидуальная характеристика квалификации каждого специалиста, а, прежде всего, **интегральный показатель квалификации «команды»**, реализующей некоторую, достаточно крупную функциональную задачу или весь программный продукт [8, 51].

Тематическая квалификация и опыт – М12 специалистов в конкретной прикладной области, для которой разрабатывается программный продукт, приближенно может оцениваться продолжительностью работы по данной тематике. При низкой тематической квалификации допускаются наиболее грубые системные ошибки, требующие больших затрат при доработке программ. Имеются примеры, когда из-за таких ошибок, допущенных на этапе системного анализа, приходилось в процессе производства изменять 70 – 90% программ. Целесообразность использования в качестве параметра квалификации, значений длительности работы в определенной прикладной области подтверждается достаточно высокой ее корреляцией с коэффициентом изменения трудоемкости. При этом квалификация системных аналитиков и непосредственных разработчиков программ в конкретной прикладной области особенно важна не столько как индивидуальная характеристика каждого специалиста, а прежде всего как интегральный показатель бригады, команды, реализующей достаточно крупную функциональную задачу. Приводимые в разных работах оценки показывают, что при изменении опыта работы в данной области от 1 до 10 лет производительность труда может повышаться в 1,5 – 2 раза.

Разработчики программного продукта должны иметь в своем составе квалифицированных *проблемно-ориентированных аналитиков и системных архитекторов* – М9, способных переводить функциональные требования заказчика в конкретные спецификации и технические требования к комплексу программ и его компонентам (см. табл. 5.5 и 5.6). Уровень квалификации аналитиков в СОСОМО II предлагается оценивать в процентах от высшей квалификации, что может снизить трудоемкость производства почти на 30% от nominalной, которой соответствует рейтинг 1,00. Они должны иметь, прежде всего, хорошую подготовку по системному анализу алгоритмов и комплексов программ, по методам оценки эффективности, организации и планированию крупных разработок и тестированию программ. Им необходима высокая квалификация по архитектурному построению, комплексной отладке и квалификационным испытаниям версий программных продуктов определенных классов, умение организовать коллектив для решения общей целевой задачи реализации проекта. Это позволит на ранних этапах исключать или сокращать дефекты, обусловленные различием представления ими целей и задач комплекса программ, а также показателей качества.

Технологическая квалификация – М13 программистов в использовании инструментальной системы автоматизации производства программных компонентов отражает опыт применения методов, средств и всего технологического процесса при создании данного типа программных комплексов. Этот опыт можно характеризовать длительностью работы с конкретной инструментальной системой автоматизации или ее версиями, базирующимися на единых технологических концепциях, опытом и длительностью работы с регламентированными технологиями, инструментальными комплексами автоматизации разработки, *языками проектирования, программирования и тестирования программ*. Особое значение имеет коллективный опыт организации и выполнения сложных проектов на базе современных автоматизированных технологий и инструментальных средств. Опыт применения конкретного комплекса автоматизации, языков проектирования и программирования может являться существенным фактором при выборе технологии для создания новых компонентов и обеспечении качества программных продуктов.

Оценка производительности коллектива существенно зависит от *стабильности состава и психологического климата* (см. М11 в

табл. 5.5 и 5.6) в коллективе специалистов и их способности к сотрудничеству и дружной совместной работе над единым продуктом. В данном факторе годы работы с конкретной технологической системой отражают не только опыт работы с инструментами, но и **сlaşженность коллектива** по проведению больших комплексных работ. Нарушение технологии, задержка при разработке отдельных модулей или групп программ может приводить к большим дополнительным затратам и значительной задержке производства продукта в целом. Это вызывает простой групп специалистов, соответствующее увеличение совокупных затрат и снижение производительности труда. Однако чаще всего коллективы специалистов имеют уже некоторый технологический опыт, и дальнейшее повышение их квалификации может дать относительно немного: 10 – 20% производительности. Небольшое влияние этого фактора обусловлено также нивелирующим влиянием трудозатрат вспомогательного и руководящего персонала, которые практически не зависят от технологической квалификации и инструментального опыта.

Программистская квалификация специалистов – **M10, M14** и опыт работы с языками программирования в наименьшей степени отражается на производительности труда. В данном факторе учитывается освоенность не только языка непосредственного программирования, но и всех компонентов средств, используемых при создании программ (спецификаций, диалога, отладки, комплексирования). После двух-трех лет работы в наибольшей степени проявляются индивидуальные особенности конкретных специалистов, их творческие способности, тщательность в работе, рациональное использование средств автоматизации. При производстве сложных комплексов программ после первых лет работы возрастание программистской квалификации может повысить производительность труда на 5 – 10%.

В этой же паре таблиц – 5.5 и 5.6 представлено влияние на трудоемкость нарушение директивного **ограничения сроков производства продукта M17** относительно типовых – номинальных. При ограничении сроков до уровня 75% от заданных, номинальных, сокращению трудоемкости может сопутствовать значительное снижение качества и увеличение рисков реализации проекта. В приведенных данных неявно предполагается, что руководство проекта заранее знает о требуемом уменьшении или возможном увеличении сроков производства и в состоянии вести планирование и управление проектом

наиболее выгодным с точки зрения минимизации трудоемкости или стоимости. При увеличении сроков производства это приводит к большим затратам времени меньшей группой специалистов на тщательные разработку и подтверждение реализации требований к продукту, на спецификации проекта, составление планов тестирования. Для уменьшения сроков разработки есть ряд путей, с помощью которых руководство может добиваться **некоторого ускорения разработки за счет увеличения трудоемкости и стоимости продукта**:

- обеспечить детальное структурирование комплекса программ на модули и спецификации интерфейсов для максимального параллелизма работы специалистов;
- приобрести технологические, инструментальные средства для более быстрого кодирования, контроля и тестирования и обучить разработчиков их использованию;
- обеспечить дополнительную подготовку программистов и группы тестирования к работе в тематической области функций продукта;
- привлечь дополнительный вспомогательный персонал;
- отложить на время не существенное документирование программного продукта.

Тем не менее, есть предел сокращению сроков производства от номинальных с помощью увеличения числа специалистов и приобретения дополнительного оборудования. Этот предел приходится примерно на 75% от оптимального, заданного и согласованного срока разработки. При максимально возможном сокращении сроков разработки до 75% от оптимальных, затраты возрастают на 30 – 40% (см. табл. 5.6).

Специалисты второй категории – технологии, обслуживающие и сопровождающие технологический инструментарий, который применяется специалистами первой категории, должны обеспечивать применение системы качества предприятия, контролировать и инспектировать процессы производства для минимизации затрат. Основные задачи второй категории специалистов должны быть сосредоточены на контроле **экономических характеристик** процессов и результатов выполнения работ, а также на принятии организационных и технологических мер для достижения их необходимого качества, обеспечивающего выполнение всех требований технического задания на программный продукт.

Технологи должны уметь выбирать, приобретать и осваивать экономически наиболее эффективный инструментарий для проектов, реализуемых конкретным предприятием с учетом особенностей создаваемых комплексов программ требуемого качества и рентабельности технологических средств. Они должны разрабатывать регламентированный технологический процесс и систему качества, поддерживающие весь ЖЦ комплексов программ и **обучать специалистов – разработчиков** квалифицированному применению соответствующих инструментальных средств и технологий.

Специалисты, управляющие обеспечением качества программных продуктов, должны владеть стандартами и методиками предприятия, поддерживающими регистрацию, контроль, документирование и воздействия на показатели качества и экономические характеристики на всех этапах ЖЦ комплекса программ. Они должны обеспечивать эксплуатацию системы качества проекта, выявление всех отклонений от заданных показателей качества продуктов и процессов, а также от предписанной технологии на этапах производства. В результате должны приниматься меры либо по устранению отклонений, либо по корректировке требований, если устранение отклонений требует больших затрат ресурсов.

После предварительных оценок экономических характеристик производства программного продукта их следует сравнивать: с заданными заказчиком сроками, реальными доступными технологическими средствами и ресурсами коллектива специалистов – рис. 5.4.

При **обобщении рассчитанных экономических характеристик**, некоторые значения могут не удовлетворить заказчика, руководителей и специалистов ведущих прогнозирование. На практике могут существовать ограничения реальной численности и квалификации коллектива, выделяемого для данного проекта комплекса программ или допустимой (директивной) длительности производства. Поэтому в методике прогнозирования должна допускаться **возможность пересчета** получаемых прогнозных значений с использованием уточненных и откорректированных значений основных факторов и требуемых характеристик специалистов. Полная стоимость и длительность разработки обычно подлежат согласованию с заказчиком. В процессе согласования уточняются сценарии проектирования и производства, и возможно изменение не только влияния некоторых факторов, но и требований технического задания на объект и харак-

теристики продукта. Это особенно необходимо, если превышаются допустимые значения стоимости или длительности разработки. При недостаточных экономических характеристиках производства программного продукта можно целеустремленно подбирать специалистов с нужной профессиональной квалификацией для улучшения экономики продукта.

С использованием модели СОСОМО для оценивания экономических характеристик производства программного продукта квалифицированные руководители и специалисты должны знать и уметь:

- получить исходные данные заказчика о размерах и характеристиках предполагаемого программного продукта;
- сформировать исходные данные о профессионализме доступного коллектива специалистов и уровне зрелости технологии производства;
- рассчитать при этих исходных данных по формулам (5.1 - 5.3) трудоемкость (стоимость), длительность и число специалистов, требующихся для производства программного продукта;
- сравнить полученные экономические характеристики с требуемой заказчиком трудоемкостью (стоимостью) и длительностью производства программного продукта;
- при превышении полученными оценками заданных требований заказчика принять меры по совершенствованию состава и профессионализма коллектива специалистов и/или повышению зрелости технологии;
- при невозможности удовлетворить требования заказчика к экономическим характеристикам, руководителям производства следует отказаться от контракта на программный продукт или согласовать заказчиком изменение требований на продукт.

Рис. 5.4

Согласованные с заказчиком основные экономические характеристики позволяют фиксировать базовый сценарий и план производства продукта, проводить расчеты распределений трудоемкости и длительности по этапам и специалистам. По этим данным может проводиться детальное календарное планирование и контролирование работ всего коллектива специалистов (см. лекцию 6). Используя графики распределения статистической относительной длительности этапов, можно оценить продолжительность основных

этапов разработки и требования к использованию специалистов определенной квалификации по этапам работ. Полученные данные могут служить основой последующего календарного планирования проектирования и производства программного продукта.

Оценивание стоимости привлечения квалифицированных специалистов для производства программных продуктов

Ориентиром решения экономических задач, связанных с набором кадров начального уровня на предприятии в области информатики, в условиях нынешнего рынка квалифицированных специалистов в России может использоваться пример в работе [45] – рис. 5.5.

Руководителям производства программных продуктов для привлечения квалифицированных специалистов необходимо уметь оценивать и сопоставлять затраты на них, учитывая:

- дефицит кадров является крупной проблемой современной программной инженерии;
- два основных варианта прием на работу: в производственный коллектив специалиста «с улицы», и обучение сотрудника в вузе для родственного предприятия;
- чтобы новый специалист начал успешно работал в реальном проекте надо потратить на его обучение 1,5 – 3 месяца, то есть три-шесть тысяч долларов;
- производственному предприятию для подготовки специалистов нужно выбирать вузы – партнеров со сходными основными интересами;
- необходимость использовать методы сокращения «текучести» и удержания квалифицированных специалистов на предприятиях.

Рис. 5.5

Выделены ***два основных варианта***: прием на работу в производственный коллектив и доучивание специалиста «с улицы», или воспитание и обучение сотрудника в вузе для родственного предприятия, которое его сопровождает и практикует на старших курсах вуза. Для этих вариантов детально рассмотрены основные факторы и составляющие реальных финансовых затрат на подготовку специалистов и эффективность их труда в коллективе производственного

предприятия. Показано, что наем и содержание высококвалифицированного специалиста дорого обходится предприятию. Если же он не может немедленно приступить к созидательной работе, то предприятие несет дополнительные существенные убытки.

Каждому предприятию нужно много квалифицированных младших программистов и тестировщиков, которые *наиболее подвержены текучке* и они требуют дополнительного образования.

Представлены оценки, во сколько такой специалист обходится предприятию при ежемесячной зарплате \$1000. С учетом накладных расходов, оборудования и помещения, различных налогов, необходимой прибыли предприятия, суммарные затраты на одного программиста или тестировщика увеличиваются вдвое до \$2000 в месяц.

За каждого найденного «на улице» специалиста агентство по найму требует 15 – 20% от его годовой зарплаты, т.е. – \$3-4 тысячи. Однако его нужно научить принятым в данном коллективе средствам производства, контроля, тестирования, формам еженедельной и итоговой отчетности, стандартам на представление программ и т.д. Нужно, чтобы новый специалист познакомился с теми частями комплекса программ, которые были написаны до него, но которые предстоит использовать. Поэтому *удержание кадров*, уже работающих на предприятии, становится важнее политики приобретения новых кадров.

Чтобы новый специалист начал успешно работать в определенном проекте, надо потратить на его обучение 1,5 – 3 месяца, то есть три-шесть тысяч долларов. Эта оценка является нижней, так как иногда приходится посыпать молодого специалиста на какие-нибудь курсы (еще \$1 – 2 тысячи), на новичка тратят время другие сотрудники (также \$1 – 2 тысячи), иногда процесс занимает и более 3 месяцев. Некоторые специалисты взяли себе за правило менять место работы каждые полгода-год, повышая свою зарплату, в результате предприятие несет при этом большие потери. Тем не менее получается оценка в \$8 – 14 тысяч начальных затрат на одного сотрудника-новичка. В нормальной ситуации после участия в 2 – 3 успешных проектах (1 – 2 года) он повышается в должности, а на его место надо искать нового специалиста. Из представленных [45] расчетов видно, что окупить даже минимальные расходы на наем новичка «с улицы» за два года не удается.

Сейчас многие крупные фирмы оказывают поддержку вузам в форме стипендий студентам и преподавателям, оплаты командировок

на свои научно-технические конференции, бесплатных инструментальных программных продуктов и своего оборудования. Простая и понятная идея подготовки кадров предприятием в сотрудничестве с вузом не так просто реализуема. Самое главное, такой подход хорошо работает не только в подготовке кадров, но и в их удержании на предприятии.

Вуз должен обеспечивать хорошую базовую фундаментальную подготовку по современной математике. Российские вузы традиционно сильны в математике и программировании малых компонентов, но вопросы управления проектами, обеспечения качества, коллективной разработки обычно остаются за рамками учебной программы. Российские и международные стандарты требуют проведения ежегодной научной работы студентов (курсовые работы), прохождения преддипломной практики на производстве, наконец, выполнения заключительной дипломной работы, причем стандарт по программной инженерии настойчиво рекомендует коллективные работы. Предприятие может выделить научных руководителей для руководства небольшими группами студентов по темам, интересным для предприятия, предоставить компьютеры и другие ресурсы, необходимые для выполнения работы, преподавателей по новейшим технологиям, с которыми кадровые преподаватели в вузах еще не успели познакомиться.

При этом перед предприятием встают **две основные проблемы**: предприятию нужно выбирать вузы – партнеры со сходными основными интересами, гарантия сохранения инвестиций родственна проблеме удержания кадров и решается схожими методами. Описанный в работе [45] опыт подсказывает, что если студент работал над курсовой работой на 3 и 4 курсах на предприятии, прошел там преддипломную практику и написал там же дипломную работу, он уже хорошо освоился на предприятии, знает условия труда и научно-технические направления предприятия, которые интересны ему. Они работают в компании в среднем 2 года, что в несколько раз меньше, чем средний стаж работы выпускников того же вуза.

Каждый **сотрудник предприятия, руководивший учебным проектом**, тратит на эту деятельность примерно 6 часов в неделю и получает за это дополнительно к зарплате 3 – 6 тысяч рублей в месяц. Всего в описываемом учебном году на подготовку студентов предприятие затратило около \$30 тысяч, было принято на стажировку 30

человек, т.е. порядок затрат – \$1000 на одного стажера [45]. Разумеется, стажер – это еще не готовый специалист, его, так же как и нового специалиста, надо доучивать, на него будут тратить свое время другие сотрудники предприятия, но профессиональный уровень получающегося, в конце концов, специалиста может быть высок.

Как удержать кадры – разумеется, для каждого наемного работника важна величина зарплаты, но проявились следующая совокупность приоритетов:

- большая стабильная компания с большой историей;
- тесная связь с вузом, научное производство;
- большой набор вертикалей, экспертиз, дающий сотруднику широкий выбор по интересам;
- высокое качество менеджмента, хорошо налаженные производственные процессы;
- конкурентно-способная зарплата;
- социальный пакет: оплата компанией добровольного медицинского страхования сотрудников; бесплатное обучение иностранным языкам; «белая» зарплата, оплата отпусков, больничных листов.

Все это позволяет уменьшить текучесть кадров до 8% [45]. Для удержания кадров по сумме перечисленных выше факторов предприятие должно казаться конкурентоспособным в глазах сотрудника с другими аналогичными предприятиями. Средние значения и ожидания от полного пакета компенсации (зарплаты плюс социальных благ) были примерно одинаковы для всех сотрудников одного уровня, откуда бы они ни пришли на предприятие.

Лекция 6

ТРЕБОВАНИЯ К ПРОФЕССИОНАЛЬНОЙ КВАЛИФИКАЦИИ РУКОВОДИТЕЛЕЙ, ПЛАНИРУЮЩИХ ДЕЯТЕЛЬНОСТЬ СПЕЦИАЛИСТОВ ПРИ ПРОИЗВОДСТВЕ КРУПНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Требования к профессиональной квалификации руководителей по планированию деятельности коллективов специалистов при производстве крупных программных продуктов

В предыдущей лекции 4 изложены рекомендации по установлению требований к функциям, размеру, качеству и безопасности программного продукта, а в лекции 5 – способы определения *прогнозируемых базовых экономических характеристик* – трудоемкости, длительности и числа необходимых квалифицированных специалистов для производства крупных программных продуктов. Эти данные следует использовать в качестве исходных для детального планирования производства программных продуктов. Варьированием количества и квалификации специалистов, а также ряда других параметров проекта можно планировать и изменять его базовые экономические характеристики для удовлетворения общих требований заказчика к функциональным характеристикам продукта. В результате могут быть получены и согласованы между заказчиком и производителями *основы для детального планирования производственных процессов* программного продукта – рис.6.1.

Квалифицированные руководители производства программных продуктов должны в соответствии с требованиями к базовым характеристикам уметь осуществлять детальное планирование, распределение этапов и трудоемкость работ, время для их выполнения, квалифицированных специалистов и других ресурсов. Они обязаны удовлетворять требования технического задания, спецификаций и кон-

тракта и обеспечивать функции и уровень качества продукта, соответствующий заданным требованиям заказчика.

***Квалифицированные руководители по планированию
деятельности коллективов специалистов
должны знать и уметь:***

- согласовывать с заказчиком исходные требования к функциям, размеру и качеству программного продукта;
- определять базовые экономические характеристики – трудоемкость, длительность и число необходимых специалистов для производства программного продукта;
- определять для согласования с заказчиком исходные данные общих планов производства программного продукта;
- подготовить общий базовый план и сроки производства программного продукта;
- определять набор планов по видам профессиональной деятельности специалистов, по компонентам, этапам и срокам;
- анализировать и определять технологическую среду, необходимые ресурсы и сроки для основных этапов производства программного продукта;
- анализировать, определять и выделять ответственных специалистов по каждому компоненту и этапу производства продукта;
- обобщать и использовать экспериментальные распределения затрат трудоемкости и длительности по видам, этапам работ и времени с начала проектирования предшествующих программных продуктов.

Рис. 6.1

Выделенный для проектирования руководитель должен отвечать за планирование и управление проектом, работами и задачами реализации планов производственных процессов, таких как заказ, разработка, поставка, эксплуатация, сопровождение и ряд вспомогательных процессов. Он должен уметь определить экономические возможности реализации спланированных процессов по срокам, проверяя наличие, соответствие и достаточность ресурсов, выделенных для выполнения и управления ***каждым процессом*** – квалифицированных специалистов, технологии и условий внешней среды, а также реальность ***сроков завершения производства*** программного продукта.

Руководителю следует установить и поддерживать в рабочем состоянии документированные процедуры, гарантирующие производство компонентов и программного продукта в соответствии с заданными требованиями и согласно плану. Такие планы должны регламентировать виды деятельности или содержать ссылки на стандарты и **определять ответственных лиц** за их осуществление. В планах необходимо определять, кому, как и каким образом следует управлять производством компонентов продукта, анализировать выполнение работ, а также установить виды и частоту отчетов специалистов для руководства, потребителей и других заинтересованных сторон, принимая во внимание все конкретные требования заказчика. В планах должны определяться производственные подразделения и состав специалистов, которые будут выполнять конкретные виды работ.

Процесс планирования должен начинаться с определения заказчиком и руководителем **исходных данных и проектных ограничений**:

- основных функций, характеристик, размеров и требуемого качества программного продукта;
- временных ограничений производства готового продукта;
- предельных возможностей численности и квалификации коллектива специалистов;
- экономических и бюджетных ограничений проекта.

Эти ограничения должны определяться параллельно с оцениванием проектных параметров, таких как размер и структура будущего продукта, а также предварительное распределение производственных процедур и функций среди специалистов. Затем **следует уметь** определить этапы разработки и то, какие результаты и документы – на компоненты, подсистемы или версии программного продукта должны быть получены по окончании этих этапов. Далее начинается итерационная часть планирования. Сначала разрабатывается **укрупненный план этапов работ** по созданию продукта. После этого проводится оценка возможности выполнения сроков работ и отмечаются расхождения между требуемым и планируемым временем завершения работ. По мере поступления новой информации о ходе выполнения проекта возможен пересмотр первоначальных оценок плана и базовых требований к всему продукту. Это, в свою очередь, может привести к изменению функций, размера продукта и графика работ.

Представленные в предыдущей лекции категории специалистов, участвующих в создании крупных программных продуктов, необходимо уметь использовать в составе больших скоординированных коллективов, объединенных решением *единой целевой задачей*. Для этого должна быть определена и зафиксирована роль каждого специалиста в коллективе, что, когда и какого качества производственные процессы следует выполнить во взаимодействии с остальными специалистами проекта. Координацию деятельности специалистов должен уметь выполнять *руководитель – менеджер проекта, для чего подготовить планы* выполнения всеми специалистами производственных процессов, которые должны содержать описания соответствующих задач и работ, а также обозначения создаваемых программных компонентов и продуктов. В *обобщенном плане руководителям должны уметь определить* [16, 37]:

- предварительные графики своевременного решения специалистами конкретных функциональных и вспомогательных задач;
- оценки трудозатрат, времени и ресурсов на их решение;
- распределение задач по специалистам, их квалификация и обязанности;
- используемые в производственных процессах критерии управления качеством реализации компонентов;
- условия и производственная инфраструктура выполнения специалистами запланированных процессов и функций.

Руководитель должен уметь *подготовить укрупненные планы по каждому виду профессиональной деятельности квалифицированных специалистов* и утвердить содержание следующих крупных *детализированных планов* по компонентам и видам деятельности:

- этапы производства функциональных компонентов и всего комплекса программ, который должен определять используемую модель жизненного цикла программного продукта;
- верификации и тестирования, которые определяют методы и средства, способные удовлетворить последовательные цели процессов устранения дефектов и контроля качества программного продукта;
- обеспечения качества компонентов и программного продукта, определяющего методы и средства, при помощи которых будет гарантировано их требуемое качество;

- реализации процессов интеграции компонентов в версии программного продукта;
- сопровождения и управления конфигурацией программного продукта, методы и средства, при помощи которых будут удовлетворяться цели процесса управления изменениями и корректировками комплекса программ;
- тиражирования, адаптации и внедрения версий программного продукта для конкретных пользователей;
- документирования процессов и результатов производства и выпуска технологической и эксплуатационной документации.

Кроме того, в составе перечисленных планов или автономно может быть полезной разработка ***вспомогательных планов***:

- обучения и подготовки пользователей для квалифицированной эксплуатации версий программного продукта;
- обслуживания пользователей в процессе эксплуатации программного продукта.

Каждый представленный план должен четко отражать ресурсы, необходимые для его реализации, ***разделение работ на этапы и временной график выполнения этих этапов***. В детализированном плане могут присутствовать ссылки на планы других видов, по которым они разрабатываются отдельно от этого плана. При этом в планах ***следует уметь***:

- определить производственные процессы, которые необходимо выполнить специалистам и возможность достижения целей плана в рамках существующих ресурсов и ограничений;
- оценить варианты достижения целей плана и определить на основе анализа рисков и возможностей, какая стратегия целесообразна;
- количественно оценить процессы и ресурсы, необходимые для выполнения работ, рассматривая варианты достижения целей плана и принимая во внимание существующие риски и возможности, чтобы весь жизненный цикл комплекса программ удовлетворял требованиям заказчика;
- выявить конкретных специалистов и группы, дающие требуемый вклад в план при производстве продукта, определить им конкретные зоны ответственности и обеспечить то, чтобы обязанности были поняты и приняты, профинансираны и достижимы;

- идентифицировать интерфейсы между элементами плана, а также с другими планами, организационными и техническими компонентами системы;
- обеспечить инструментарий для того, чтобы планы проекта были формально разработаны, реализованы, поддержаны и доступны лицам, вовлеченным в производство;
- использовать упорядоченные подходы для того, чтобы регулярно оценивать процессы выполнения планов, принимать меры для корректировки отклонений от каждого плана и предотвращения повторения проблем, выявленных в процессе производства.

Все планы должны иметь в своем составе *детальные графики, идентифицирующие*: этапы работ; входные, выходные данные и описания решаемых задач; необходимые ресурсы и сроки выполнения; взаимосвязи и зависимости этапов и работ. Должно быть установлено организационно-техническое *взаимодействие между различными группами специалистов*, которые вносят свой вклад в процессы и обеспечение качества производства, а необходимая информация должна документироваться и регулярно анализироваться. В планах работ поставщиков и субподрядчиков *надо уметь* определить *границы ответственности* за каждую часть программного продукта и за способ обмена технической информацией между всеми сторонами.

Планы должны *регулярно пересматриваться* в процессе реализации проекта. Для внесения изменений в план требуется специальная организация и регламентирующий документ, позволяющие отслеживать, фиксировать эти изменения и отображающие выполнение очередного этапа плана производства программного продукта. Эти отчеты должны подводить краткие итоги окончания отдельного, логически завершенного этапа плана проекта. Обычно при завершении основных больших этапов заказчику предоставляются результаты их выполнения – *контрольные проектные компоненты*. Это может быть документация, прототип программного продукта, законченные подсистемы и компоненты. Менеджер должен осуществлять текущий контроль за выполнением планов, подготавливая как внутренние отчеты о развитии каждого процесса, так и внешние обобщенные отчеты для заказчика в соответствии с условиями договора. Все обнаруженные дефекты и результаты их устранения должны быть документально оформлены, а также в установленные сроки подтверждены

полная реализация процессов и выполнение утвержденных планов. После создания всех запланированных программных компонентов и продуктов менеджер должен определить степень их соответствия критериям, установленным в договоре или организационной процедуре.

Особого внимания требует *умение руководству разработчиков сотрудничать с заказчиком*, своевременное предоставление ему нужной информации и решение оперативных вопросов для обеспечения качества программного продукта. Если представитель заказчика обладает соответствующей компетентностью, то он может представлять конечного пользователя продукта, а также административное руководство проектом и иметь полномочия заниматься контрактными вопросами изменения планов. К ним относятся также определение и уточнение требований спецификаций к функциям и показателям качества продукта, одобрение и утверждение предложений разработчика, а также заключение дополнительных соглашений с ним. Целесообразно уметь планировать и регулярно проводить разработчиком и заказчиком *совместных анализов реализации планов проекта*, либо проводить такие анализы в случае значительных проектных событий, *чтобы охватить*:

- состояние и развитие выполняемых работ по производству и модификации программного комплекса и компонентов;
- соответствие результатов производства, согласованной спецификации требований заказчика;
- результаты проверок текущего состояния проекта и готовности к приемочным испытаниям;
- состояние работ, касающихся подготовки и обучения конечных пользователей разрабатываемой системы.

Входные проектные данные и требования к планам создания комплекса программ, включая установленные законодательные и регламентирующие нормативные требования, должны оформляться документами, а их выбор проанализирован заказчиком на адекватность. Неполные, двусмысленные или противоречивые требования должны быть предметом урегулирования с лицами, ответственными за их предъявление. Специфиацию требований должен представить потребитель – заказчик. Однако по взаимному согласию ее может подготовить разработчик в тесном сотрудничестве с потребителем для предупреждений разногласий путем, например, уточнения опре-

делений терминов, объяснения предпосылок и обоснования требований. Спецификация требований к планам может быть представлена и согласована в составе спецификации всей системы. Если программный продукт нуждается во взаимодействии с другими программными или аппаратными продуктами, то в его требованиях это должно быть оговорено непосредственно или при помощи ссылок на планы и интерфейсы между разрабатываемыми и другими применяемыми продуктами.

Выходные проектные данные реализации планов должны быть документально оформлены и выражены так, чтобы их можно было проверить и подтвердить соответствие входным проектным требованиям. Выходные проектные данные должны содержать критерии приемки продукта заказчиком или ссылки на них, а также идентифицировать те характеристики планов проекта, которые являются критическими для безопасного и надежного функционирования и своевременного применения программного продукта. К составу **выходных проектных данных могут относиться**:

- описание и сроки представления результатов испытаний и применения программного продукта;
- комплект оформленного программного продукта и эксплуатационной документации для пользователей;
- комплект технологической документации для обеспечения возможности модификации и сопровождения версий программного продукта.

Цель анализа и планирования технологической среды жизненного цикла комплекса программ состоит в том, чтобы определить производственные методы, инструментальные средства, процедуры, языки программирования и аппаратные средства, которые будут использоваться для производства, верификации, управления и подготовки документации программного продукта. План должен включать стандарты, методы предотвращения ошибок и обеспечения отказоустойчивости, которые ограничивают возможность внесения ошибок, и такие методы тестирования, которые гарантируют их обнаружение. Цель методов обеспечения отказоустойчивости состоит в том, чтобы включить в проект такие средства тестирования, обеспечения качества и безопасности применения программного продукта, которые могут гарантировать, что он будет адекватно реагировать на ошибки

входных данных, предотвращать выдачу ошибочных данных и контролировать возможность проявления ошибок.

Для планирования производства сложных программных продуктов целесообразно *уметь обобщать и использовать экспериментальные статистические распределения основных экономических характеристик* – трудоемкости, длительности и числа специалистов по этапам работ и по реальному времени реализации компонентов *предшествующих проектов* (см. рис. 6.1). Относительные значения распределения этих величин на интервале реализации крупных продуктов несколько различаются в зависимости от размера и типа комплекса программ. Однако общие тенденции состоят в наименьших затратах на начальных и конечных этапах, и в наибольших суммарных затратах, на средних этапах производства *новых* программных компонентов и продуктов.

В ряде работ [5, 26, 54] опубликованы результаты измерения основных *экономических характеристик в зависимости от этапов производства* комплексов программ. В оценках совокупных затрат на производство полностью новых комплексов программ доминирует трудоемкость (стоимость) непосредственной разработки программных компонентов. Для этих распределений характерно *наличие максимума* в использовании трудовых ресурсов на средних этапах разработки – *на этапах программирования и автономной отладки компонентов*. Эти этапы полностью заняты производством компонентов, а на остальных этапах к компонентам относится только часть таких затрат. После выделения затрат на создание компонентов, остающиеся затраты можно отнести к процессам проектирования, а также интегрирования компонентов и испытаний комплексов программ.

Распределение трудоемкости на проектирование и производство комплексов программ *квалифицированному руководителю полезно уметь делить в планах* на шесть крупных этапов:

- 1 этап – предварительное проектирование;
- 2 этап – детальное проектирование;
- 3 этап – программирование компонентов;
- 4 этап – тестирование и автономная отладка компонентов;
- 5 этап – интеграция компонентов и комплексная отладка продукта;
- 6 этап – испытания и документирование программного продукта.

При предварительном анализе целесообразно выделять подмножества работ, соответствующие трем категориям квалификации специалистов: руководители разработки и системные аналитики; непосредственные разработчики программных компонентов и специалисты по их комплексированию; вспомогательный персонал, обеспечивающий процессы разработки и документирование комплекса программ. Первая и третья категории специалистов непосредственно не взаимодействуют с текстом программ при их производстве, однако их труд является неотъемлемой частью всего процесса проектирования и производства и в крупных проектах составляет *около половины затрат*. Экспериментальные распределения относительного числа занятых специалистов при производстве сложных комплексов программ одинакового размера обычно подобны и положения экстремумов в этих распределениях различаются относительно мало. При детальном планировании в конкретных графиках приходится выделять еще ряд категорий специалистов, перечень которых представлен в лекции 5.

При разработке планов относительное число необходимых высококвалифицированных специалистов не велико на начальных и конечных этапах. В составе каждого из этих крупных этапов должны выполняться основные работы, определяющиеся названием и целью этапа, а также *ряд общих видов вспомогательных работ*, присущих, в той или иной мере, практически каждому этапу проектирования и производства. Такими *видами работ* являются [5, 26]:

- анализ и корректировка требований к комплексу программ и его компонентам;
- проектирование и уточнение функций и структуры компонентов и комплекса в целом;
- программирование изменений компонентов и их взаимодействия;
- планирование и выполнение тестирования компонентов и комплекса программ;
- верификация и аттестация компонентов и комплекса программ после изменений;
- анализ, оценка и управление качеством программных компонентов и комплекса в целом;
- документирование результатов этапа производства, создание и корректировка технологических и эксплуатационных документов.

Трудоемкость анализа, изменения требований и проектирования комплекса программ, естественно, быстро убывает при приближении к завершающим этапам производства продукта. Каждый **вид вспомогательных работ требует** в среднем 5 – 15% от суммарной трудоемкости, а доминирующие работы этапа составляют до 50% (за 100% принята суммарная трудоемкость каждого из основных этапов). Однако на этапе интегрирования и испытаний работы по программированию, тестированию, верификации в сумме могут составлять до 30% общей трудоемкости этапа. На всех этапах должно осуществляться документирование процессов и результатов, на что в совокупности требуется около 20% трудоемкости этапа.

Важно уметь на каждом этапе выделять долю труда, которую можно отнести к тестированию и отладке компонентов, их сборке, комплексированию и испытаниям крупных функциональных частей комплекса программ. **На программирование и автономную отладку программных компонентов** требуется обычно около **половины** трудоемкости и до 40% общей длительности разработки. Возрастание относительных затрат всех видов при комплексной отладке в реальном времени обусловлено высокой трудоемкостью динамического тестирования и испытаний для этого класса программных продуктов. Относительная трудоемкость и длительность при испытаниях различных классов программных продуктов приблизительно одинаковая и составляет 8 – 10% от совокупных затрат на производство. Однако абсолютная трудоемкость для испытаний программ, не связанных с реальным временем, приблизительно в 5 раз меньше, чем программ, требующих динамических испытаний, а абсолютная длительность этого процесса сокращается всего в полтора раза.

Совокупные относительные затраты **на автономную и комплексную отладку, а также на испытания** всех классов крупных комплексов программ приблизительно одинаковые и составляют **по трудоемкости и по длительности около 30%**. Эти соотношения имеют тенденцию медленно снижаться за счет развития современных технологий, инструментальных средств автоматизации и повышения качества начальных этапов проектирования и производства. Необходимость проведения комплексирования и динамического тестирования в реальном времени может привести к меньшему улучшению экономических характеристик для класса программ реального времени при активном внедрении средств автоматизации разработки.

Требования к профессиональной квалификации руководителей, разрабатывающих графики производства программных продуктов

Руководитель должен *уметь оценивать* длительность этапов и всего проекта, определять виды и размеры ресурсов, необходимых для реализации отдельных этапов и типов работ, и представлять их в виде согласованных с заказчиком графиков последовательности проектирования и производства. Если данный проект подобен ранее реализованному, то график производства нового проекта можно взять за основу. Если проект является инновационным, первоначальные оценки длительности и требуемых ресурсов почти наверняка будут *слишком оптимистичными*, даже если планировщик попытается предусмотреть все возможные неожиданности. С этой точки зрения проекты производства программных продуктов не отличаются от больших инновационных технических проектов, в графиках которых неожиданно могут возникать проблемы и трудности. Именно поэтому *графики работ являются динамичными* и их необходимо постоянно контролировать и обновлять по мере поступления новой информации о ходе выполнения проекта.

Как только утверждается архитектура комплекса программ, определяются частные производственные задачи. В это же время могут более детально рассчитываться трудозатраты конкретных специалистов на компоненты, а также кто, когда и над чем будет работать. В *процессе составления графика* (рис. 6.2) весь комплекс работ, необходимых для реализации продукта, разбивается на отдельные этапы и оценивается время и затраты ресурсов, требующиеся для выполнения каждого этапа и вида работ.

Обычно многие этапы и работы выполняются параллельно. График производства должен это предусматривать и распределять специалистов и производственные ресурсы между ними оптимальным образом. Нехватка ресурсов для выполнения какого-либо критического этапа – частая причина задержки выполнения всего проекта. При расчете длительностей этапов производства компонентов комплекса программ руководитель должен уметь учитывать, что реальное выполнение любого этапа обычно не обходится без проблем и задержек. Специалисты могут допускать ошибки или задерживать результаты работы, техника может выйти из строя, аппаратные или про-

граммные средства поддержки процесса производства могут поступить с опозданием. Если проект инновационный и технически сложный, это становится дополнительным фактором появления непредвиденных проблем и увеличения длительности реализации продукта по сравнению с первоначальными оценками.

Квалифицированные руководители при разработке графиков производства программных продуктов должны знать и уметь:

- разработать и документировать проекты графиков производства программного продукта по этапам, срокам, компонентам и специалистам;
- распределять ресурсы и специалистов с учетом неопределенностей внешней среды, обстоятельств и возможного изменения требований заказчика к программному продукту;
- формализовать задание на составление календарного плана и графика производства с учетом ресурсов, сроков и квалификации специалистов;
- применять инструменты автоматизированного формирования графиков производства программного продукта – диаграмм Ганта;
- отслеживать реализацию и оптимизировать графики для обеспечения баланса использования времени и ресурсов производства продукта;
- выявлять и устранять отклонения от плана производства и оперативно проводить его корректировку по согласованию с заказчиком;
- оформлять текущие и обобщающие отчеты о состоянии, изменениях графика производства и результатах выполнения плана.

Рис. 6.2

Кроме временных затрат, руководитель должен рассчитывать ***производственные ресурсы***, необходимые для успешного выполнения каждого этапа или компонента. Особый вид ресурсов – это коллектизы и группы специалистов, привлеченных к выполнению проекта.

Существует ***эмпирическое правило***: первоначально оценивать временные затраты следует так, как будто не может случиться ничего непредвиденного, а затем значительно (на 30 – 50%) увеличивать эти оценки для учета возможных проблем. Трудно прогнозируемые проблемы существенно зависят от типа и параметров проекта, а также от квалификации и опыта членов коллектива специалистов.

Даже в том случае, если итерационные методы предусматривают и допускают постепенное **уточнение и расширение требований заказчика**, целесообразно назначить некоторую дату, после которой в проект не должны вноситься новые требования. Рекомендуется включать в график **резервные интервалы времени**. Это делается из-за того, что невозможно полностью учесть действие всех внешних факторов. К тому же люди практически всегда растягивают решение производственной задачи на все отведенное время. Если при этом в графике не предусмотрено буферного времени, то все временные рамки наползают друг на друга и начинают перекрываться. Одним из вариантов предупреждения возможных неожиданностей является отведение под этапы заведомо большего количества времени. Другой вариант заключается в создании **буферных интервалов в производстве**, во время которых не планируется решение каких-либо реальных задач.

Чаще всего для представления графиков работ используются относительно простые **диаграммы Ганта** [16, 35, 42]. Графики Ганта – это **методика планирования проектов**, которую целесообразно использовать для достижения нескольких целей, включая календарное планирование производства, финансовое планирование, последовательное использование специалистов и различных ресурсов. График Ганта представляет собой гистограмму, где каждая горизонтальная линейка обозначает отдельный вид производственной деятельности (рис. 6.3).

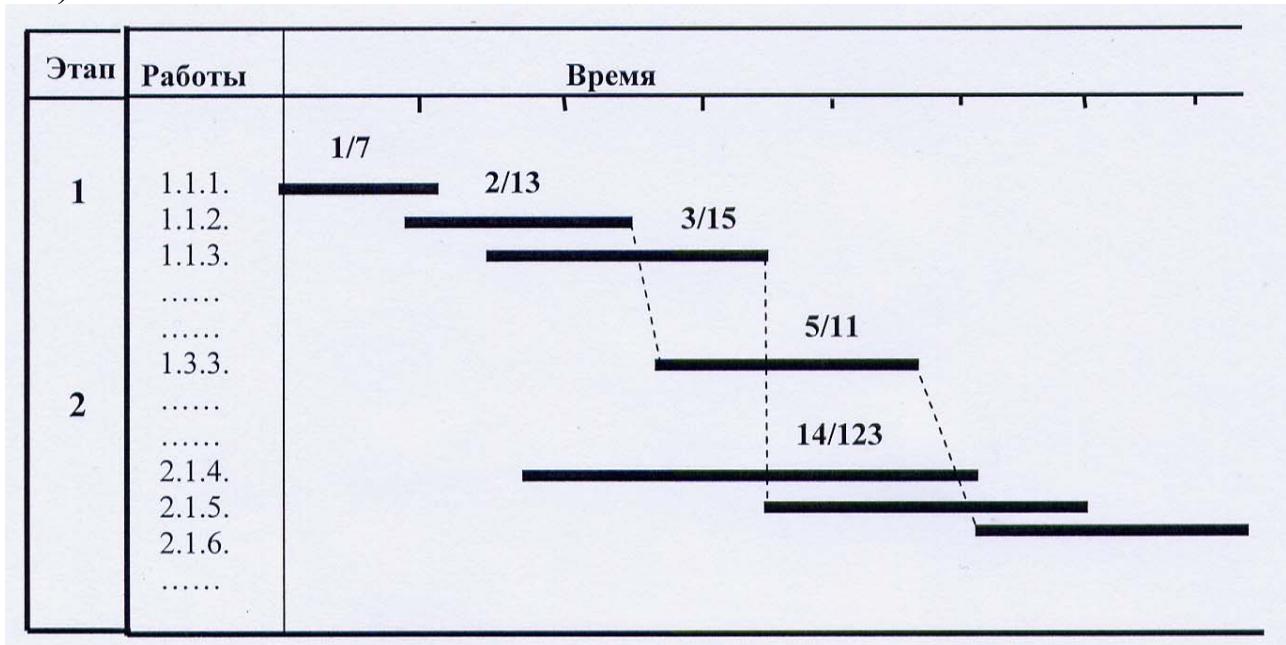


Рис. 6.3

Линейки частных работ изображаются относительно шкалы реального времени. Длина каждой линейки пропорциональна продолжительности времени, запланированного на выполнение определенной работы или использование ресурса. Слева должны быть перечислены производственные действия специалистов и содержание работ, а справа – указаны горизонтальные полоски, длина которых соответствует длительности выполнения каждого этапа работ в соответствии с временной шкалой. Визуально диаграмма Ганта представляет собой последовательность действий, выполняемых в пределах жизненного цикла комплекса программ. Трудоемкость и число необходимых специалистов может указываться числами над линейкой для каждого этапа (см. рис. 6.3 – в числителе трудоемкость работ; в знаменателе – число специалистов).

Календарное планирование производства комплекса программ можно начать с декомпозиции всего комплекса работ, этапов, компонентов и специалистов. Однако во время выполнения календарного планирования и его реализации могут проявиться новые виды деятельности, не предусмотренные во время первоначального планирования. Тогда руководитель должен вернуться назад и пересмотреть структуру декомпозиции работ и календарные планы для того, чтобы найти место вновь появившимся видам деятельности. При выполнении больших проектов, содержащих множество действий и другой отображаемой информации, диаграммы Ганта могут быть очень громоздкими. Это позволяет увидеть «общую картину» процессов производства, однако ориентироваться в подобных графиках очень сложно. Поэтому может быть полезным структурировать крупные графики на несколько более мелких по функциональным задачам проекта или достаточно автономным коллективам, и на обобщающем сводном графике производства предприятием всего программного продукта отражать компоненты без детализации.

График Ганта можно использовать для **распределения ресурсов и планирования обеспечения производства специалистами**. Если необходимо спланировать по срокам работу конкретных специалистов, то можно воспользоваться графиком Ганта, на котором каждая полоса (горизонтальная линейка) будет отражать одного из них или небольшую группу. На таком графике **специалисты являются ресурсами**, и на нем может быть показана величина ресурсов во время реализации производства компонентов программного продукта.

Можно пометить соответствующие части полос для обозначения количества времени, которое каждый специалист должен предположительно тратить на выполнение той или иной деятельности.

Несмотря на то, что графики Ганта наглядно показывают каждый вид деятельности и затрачиваемое на его выполнение время, они зачастую *не отражают зависимостей между решаемыми задачами*. Некоторые средства формирования рабочих графиков производства позволяют изображать связующие линии зависимостей между этапами (см. рис. 6.3 – пунктирные линии отражают зависимости начала работы только после завершения предшествующей работы, например, начало работы 2.1.5 только после завершения 1.1.3).

Независимо от того, как *выявляются отклонения от плана*, руководитель должен принимать решение об их устраниении. Заключительная часть графика может обозначать «*резерв времени*», и самый последний срок, к которому выполнение всего производства продукта должно быть завершено. Там, где производство ограничивается только человеческим трудом, обычно коррекция производится добавлением сотрудников или увеличением времени сверхурочной работы задействованных специалистов. Однако в проектировании крупных программных продуктов недостаток чаще всего ощущается не в рабочей силе как таковой, а в *интеллектуальных ресурсах конкретных специалистов*, поэтому с отклонениями от календарного плана здесь бороться сложнее. Важно учитывать, что механическое увеличение числа специалистов иногда не позволит уложиться в поставленные сроки, и помочь в этом могут *только профессиональные способности некоторых специалистов*. Таким образом, приемлемым вариантом может стать временный перевод выдающихся специалистов на работу над проблемной частью проекта, либо наем дополнительных специалистов – консультантов для устранения недостатков ресурсов.

Другой вариант обеспечения выполнения плана – подробное, критичное *изучение требований заказчика и исключение из них не самых актуальных – «чистка требований»*. Важнейшим принципом планирования производства комплексов программ является упорядочивание приоритетов требований в порядке возрастания важности для заказчика и/или пользователей. Зачастую можно согласовать с заказчиком и принять решение об исключении некоторых требований для того, чтобы уложиться в заданные сроки.

Может случиться так, что после рассмотрения всех вариантов выхода из цейтнота подходящее решение найдено не будет. Тогда правильным действием руководителя может быть **признание несостоимости первоначальных планов и графиков** и их полный пересмотр, исходя из нового понимания сложностей поставленных задач, профессиональной квалификации имеющегося коллектива специалистов и доступности ресурсов. Руководитель должен уметь рассматривать график как попытку прогнозирования жизненного цикла производства. Но также **важно учитывать риски**, связанные с дефектами и негативным прогнозированием, и быть готовым к возможному пересмотру планов и сроков.

Составляя любой рабочий график, руководителю следует учитывать **принцип неопределенности**. Спланированный рабочий график – это предположение о будущих действиях коллектива специалистов при оптимальных условиях. Возможными последствиями неопределенностей можно управлять с помощью предусмотренных в графике **временных резервов**, которые являются аналогами экономических и финансовых резервов, применяемых при построении бизнес-планов. Производство программного продукта обычно должно быть окончено к определенной дате, что интересует заказчика в первую очередь, с возможностью некоторой задержки. После разработки структуры комплекса программ, оценки продолжительности всех производственных этапов и составления рабочего графика необходимо предусмотреть достаточное количество буферов для непредвиденных обстоятельств с целью учета возможных неопределенностей.

Любое производство полно рисков, поэтому при объявлении предполагаемой даты завершения создания продукта полезно указывать **вероятность успеха**. Для того чтобы **эффективно управлять неопределенностью**, целесообразно использовать в графике начальные оценки длительности производственных этапов. В этом случае вероятность своевременного завершения каждого этапа будет составлять только 50%. После этого некоторую неопределенность можно внести во временные буферы завершающего действия, что при непосредственной работе над проектом позволит управлять неопределенностью сроков сразу для всех производственных действий. Если вероятность своевременного завершения каждого этапа составляет только 50%, то некоторые из них будут завершены позже установленного срока. Все разницы в сроках завершения этапов

должны быть **вычленены из буфера резервов** непредвиденных обстоятельств последнего этапа или всего производства в целом. Однако некоторые этапы могут быть завершены досрочно, поэтому выигрыш во времени также должен быть **добавлен** к буферу неопределенности проекта. Это означает, что размер буфера может оставаться практически неизменным.

Для планирования и мониторинга планов производства различных сложных продуктов создано несколько **инструментов технологических программных продуктов**. Они обеспечивают представление планов в различной графической форме – графиков Ганта, сетевых графиков и др. [16, 35, 42]. Большинство из них не имеют проблемного ориентирования и требуют перед применением подготовки состава, содержания и характеристик этапов и работ, значений времени и трудоемкости, необходимых ресурсов, состава и квалификации конкретных специалистов для наполнения содержания графиков. В **качестве примера**, достаточно простым и удобным для руководителей может использоваться программный продукт **Microsoft Project** для представления, изменения и управления сложными планами производства программных продуктов в виде графиков Ганта. Его основные функции отражены **четырьмя группами процедур**: задачи проекта; ресурсы проекта; отслеживание реализации и изменений проекта и отчеты. Эти группы позволяют манипулировать планами и конкретными работами с учетом их взаимосвязей, доступных ресурсов, локальных и общих целей планирования. Для очень крупных проектов возможно выделение определенных этапов или групп работ и построение для них локальных более детальных графиков Ганта [35].

Исходные данные о характеристиках компонентов проекта, необходимые для составления и описания плана, должны быть представлены руководителем в группе **Задачи проекта**:

- определение проекта – состава и взаимодействия основных задач;
- определение рабочего времени решения задач и ограничения параметров всего проекта;
- ввод задач и характеристик проекта и их размещение на графике с учетом связей и времени;
- организация этапов и последовательности решения задач;

- установка крайних сроков и ограничений для отдельных и/или групп задач;
- оценка возможных рисков проекта от проведенного размещения задач;
- публикация исходного состава и предварительного размещения задач на графике проекта.

Ресурсы проекта:

- выбор состава, квалификации специалистов и характеристик оборудования для решения задач проекта;
- задание типов резервирования определенных ресурсов;
- определение рабочих часов использования определенных ресурсов;
- назначение специалистов и оборудования на решение каждой задачи;
- добавление и уточнение дополнительных сведений о ресурсах и их связей;
- публикация сводных данных о доступных ресурсах проекта.

Отслеживание, изменение и оптимизация графика проекта:

- создание и сохранение первичного базового плана для сравнения с последующими версиями при изменении графика;
- организационная и техническая подготовка для отслеживания реального хода работ над графиком проекта;
- включение сведений о реальном ходе и экономических характеристиках выполнения задач проекта;
- контроль состояния директивных сроков выполнения и завершения плана производства продукта;
- внесение изменений в задачи, ресурсы и риски проекта, а также пересмотр планов и оптимизация графика;
- публикация данных о выполненных планах, компонентах графика проекта и их сравнение с требованиями заказчика.

Отчеты о проекте и его графиках:

- отражение текущего состояния производства продукта как отчета на определенный момент времени;
- анализ и контроль критических задач производства продукта и подготовка решений;
- контроль, сокращение или устранение рисков проекта;

- контроль выделения и использования времени, ресурсов и специалистов;
- публикация сводных данных о реализации и завершении графика производства продукта.

Итерационное планирование целесообразно проводить поэтапно, последовательно детализируя содержание производственных процессов с учетом функций, времени и характеристик конкретного комплекса программ. Суммарные затраты экономических ресурсов, уточняемые при их оценках по этапам и по перечням работ, *не должны выходить за пределы*, ограниченные договором с заказчиком. Если эти условия не выполняются, то *итерационно следует* пересматривать и изменять: или ресурсы, определенные заказчиком, или график и распределение ресурсов по этапам производства, или состав и содержание работ на производственных этапах. Во всех случаях следует контролировать обеспечение требований к функциям, качеству и характеристикам программного продукта, утвержденным заказчиком.

Необходимые временные, трудовые и другие ресурсы для выполнения и отражения работ на этих графиках, отличаются глубокой зависимостью от требований к характеристикам конкретного программного продукта и условий его производства. В то же время более определенными являются базовые перечни работ и сопутствующих им документов на каждом этапе для крупных комплексов программ управления и обработки информации. Для каждого этапа работ на графике Ганта полезно отображать значения трудоемкости (знаменатель) и необходимое число специалистов (числитель на рис. 6.3), а также по вызову желательно получать справку о составе и *содержании конкретных работ и отчетных документов*, начиная от результатов системного анализа и до завершения испытаний версии программного продукта.

Лекция 7

ТРЕБОВАНИЯ К ПРОФЕССИОНАЛЬНОЙ КВАЛИФИКАЦИИ СПЕЦИАЛИСТОВ, УПРАВЛЯЮЩИХ МОДИФИКАЦИЕЙ, СОПРОВОЖДЕНИЕМ И КОНФИГУРАЦИЕЙ КРУПНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Требования к профессиональной квалификации специалистов, управляющих сопровождением и модификацией крупных комплексов программ

По мере развития производства и применения сложных программных продуктов выявилось, что интегральные затраты на сопровождение и создание новых версий могут значительно превосходить затраты на разработку их первой версии. По мере накопления эксплуатируемых продуктов и их компонентов, все большее число специалистов переходит из области непосредственного создания новых программных компонентов в область системного проектирования, управления конфигурацией и *создания новых версий* программного продукта на базе готовых повторно используемых компонентов. Зачастую руководители и разработчики программного продукта не предусматривают этот процесс, что значительно снижает эффективность последующего совершенствования и применения версий созданного продукта.

Для обеспечения развития и модификации версий программного продукта необходим *квалифицированный руководитель, который должен организовать производственный процесс реализации версий – сопровождения продукта*. Для этого следует разработать план сопровождения и соответствующие процедуры, а также выделить конкретные ресурсы и специалистов. После поставки заказчику программного продукта сопроводитель в соответствии с договором и предложениями о модификации или отчетами о дефектах должен изменять – сопровождать соответствующие программы, данные и документы.

При организации **сопровождения и модификации** крупных программных продуктов следует учитывать важные **психологические факторы**, усложняющие отбор, обучение и деятельность **квалифицированных руководителей и специалистов** в этой области – рис. 7.1:

- эта деятельность требует высокой квалификации, больших, творческих и умственных затрат, связанных, прежде всего, с необходимостью одновременного, широкого охвата и анализа множества компонентов и их взаимосвязей, находящихся в различных состояниях завершенности модификаций или устранения дефектов;
- корректируемые компоненты зачастую разрабатывались в прошлом в разное время, различными специалистами, в различном стиле, с неодинаковой и не полной документацией, что усложняет освоение их содержания при внесении изменений и устранении дефектов;
- сложная, творческая сторона работ при сопровождении выявляется тем, что приходится овладевать и анализировать программы, разработанные ранее другими специалистами, которые зачастую оказывается проще не корректировать, а разработать заново;
- комплексы программ, прошедшие широкие испытания и эксплуатацию у заказчиков гарантируют достигнутое качество результатов функционирования, и любые в них изменения имеют высокий риск внесения дополнительных ошибок и ухудшения качества, что усложняет крупные модификации;
- выполняемые работы требуют особой, скоординированной тщательности корректировок и четкого регламентированного взаимодействия ряда специалистов, различающихся квалификацией и ответственностью;
- процессы и результаты сопровождения не отличаются наглядностью и внешним эффектом, проявлением их размера и сложности, вследствие чего не престижны среди рядовых специалистов и недооцениваются руководителями крупных проектов.

Руководителям следует формализовать **конкретный план сопровождения программного продукта**, выделив из общего состава процессов жизненного цикла, который адаптировать с учетом функций, размера и особенностей проекта. При подготовке производственного процесса **квалифицированные руководители** должны разработать, документировать и выполнить планы и процедуры для прове-

дения деятельности специалистов по сопровождению. Они должны устанавливать процедуры для получения, записи и сообщений о дефектах, проблемах и запросах на изменения от пользователей, и для обеспечения обратной связи к пользователям.

***Квалифицированные руководители и специалисты
для сопровождения крупных комплексов программ
должны знать и уметь:***

- учитывать психологические факторы, усложняющие отбор, обучение и деятельность людей при процессах сопровождения;
- составлять и реализовать планы и стратегии модификации, сопровождения и формирования версий программных продуктов;
- осуществлять анализ и устранять дефекты компонентов и их взаимосвязей, находящихся в различных состояниях завершенности модификаций;
- корректировать компоненты, которые разрабатывались в разное время, различными специалистами, с не полной документацией;
- учитывать, что любые изменения имеют высокий риск внесения дополнительных дефектов и ухудшения качества версии продукта;
- выполнять скоординированные корректировки рядом специалистов, различающихся квалификацией, ответственностью и иерархией права на изменения;
- оценивать совокупность затрат на модификацию программ, устранение дефектов и рентабельность выполнения или прекращения изменений версий программного продукта;
- оценивать и выделять в среде производителей или заказчиков ответственных за финансирование новых версий программного продукта;
- оценивать факторы, допускающие прекращение сопровождения и модификацию определенной версии или полностью программного продукта.

Рис. 7.1

Следует устанавливать функциональную связь с процессом управления документированием и конфигурацией для управления модификациями и версиями программного продукта. План сопровождения должен быть подготовлен квалифицированными руководителями во время первичного проектирования версии программного продукта и включать в себя процессы анализа предложений заказчика и пользователей по внесению изменений в программный продукт.

Общий план сопровождения должен определять:

- причины необходимости сопровождения программного продукта;
- состав исполнителей работ по сопровождению;
- роль, квалификацию и обязанности каждого специалиста, вовлеченного в процесс сопровождения;
- как должны быть выполнены основные производственные процессы и работы при модификациях продукта;
- какие имеются и необходимы ресурсы для сопровождения;
- методы и средства организации работ по управлению и выпуску новых версий продукта и синхронизации совокупности изменений в них;
- перечень всех проектных результатов, документов и продуктов, подлежащих поставке заказчику в процессе сопровождения;
- критерии завершения этапов соответствующей деятельности, работ и задач при реализации изменений;
- состав отчетных материалов по этапам, затратам и графикам проведения работ;
- состав отчетных материалов по проблемам, изменениям и устраниенным дефектам при сопровождении.

Стратегия сопровождения должна быть ориентирована на ***квалифицированных специалистов***, необходимых и доступных для обеспечения развития и модификаций программного продукта. Процесс разработки изменений включает в себя ряд работ, связанных с планированием, анализом и оценкой дефектов и модификаций:

корректировок, усовершенствований или адаптаций программного продукта к новой или измененной среде;

определением размера модификации, стоимости, затрат времени на модификацию;

рисков влияния изменений на производительность, надежность или безопасность версии программного продукта.

Процесс устранения дефектов начинается с анализа несоответствий требованиям независимо от их происхождения или источника, которые обнаружены в ходе выполнения разработки, эксплуатации, сопровождения или других процессов. Целью данного процесса является обеспечение своевременного, ответственного и документируемого анализа и регистрации всех обнаруженных дефектов и определения причин их возникновения. Подготовка процесса устране-

ния дефектов должна быть циклически замкнутой, обеспечивающей в соответствии с условиями договора:

- своевременное документирование и ввод всех обнаруженных дефектов в процесс их анализа и устранения;
- организацию работ квалифицированных специалистов над ликвидацией дефектов;
- соответствующие уведомления заинтересованных сторон о выявленных дефектах;
- определение, анализ и возможное устранение причин их возникновения;
- реализацию корректировок и внесение изменений в соответствующие компоненты;
- учет и документирование состояний дефектов и их изменений;
- сопровождение отчетов о дефектах и изменениях.

Процесс устранения дефектов должен содержать схему классификации и установления *их приоритетов*, для каждого должен быть определен соответствующий класс и приоритет для упрощения анализа причин его возникновения и устранения. В отчетах о дефектах должен быть приведен анализ причин их возникновения. Реализованные специалистами корректировки в соответствующие компоненты должны быть оценены по следующим критериям: какие дефекты выявлены; какие неблагоприятные причины их возникновения устранины; какие изменения правильно внесены в соответствующие программные продукты и процессы; какие дефекты дополнительно обнаружены.

Квалифицированные специалисты сопровождения должны уметь разрабатывать эффективные варианты выполнения модификаций версий программного продукта: документально оформлять проблему и цель изменения; запрос на модификацию; результаты анализа и варианты реализации изменений. Следует получать утверждение заказчиком или руководителем выбранного варианта модификации согласно договору. Реализация модификации должна содержать анализ и определение, какие документы, компоненты программного продукта и версии требуют изменений, что должно быть документировано.

Для того чтобы подтвердить актуальность представленных отчетов о дефектах, специалист сопровождения должен *дублировать и ве-*

рифицировать возникшие проблемы – дефекты, выполняя следующие производственные этапы:

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

До внесения изменений в систему и программный продукт в соответствии с договором с заказчиком руководитель или специалист сопровождения должен **уметь согласовать выбранный вариант корректировки**, выполнив следующее:

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

Четкая организация и автоматизация этого процесса позволяют избегать множества вторичных ошибок, обусловленных недостаточной координацией проводимых корректировок и формирования новых версий сложных программных продуктов коллективом специалистов. Этому должна способствовать утвержденная **иерархия принятия решений на изменения** компонентов и комплекса в целом должностными лицами производства (таблица 7.1), поддержанная методами и средствами защиты от несанкционированного доступа при выполнении корректировок специалистами различной квалификации и права доступа на разных уровнях проекта. Каждый вариант компонента или комплекса программ и изменений требований должен соответствовать правилам идентификации [12, 50, 63].

Таблица 7.1

Типы изменений в комплексах программ,
право на их выполнение и утверждение

Объекты изменения	Типы изменений объектов	Право на выполнение изменения имеют:	Право на утверждение изменения имеют:
Версии модулей и компонентов программ и документов	Устранение дефектов в программных модулях и компонентах	Программисты – разработчики модулей и компонентов	Руководители разработки требований к функциональным группам программ
Версии функциональных групп программ и документов	Корректировка функций и взаимодействия программных компонентов	Руководители разработки функциональных компонентов	Руководитель-архитектор требований к программному продукту
Версия программного продукта и комплекса документации	Модификация и улучшение функций и качества версии программного продукта	Руководитель-архитектор требований к программному продукту	Руководитель и заказчик проекта программного продукта
Версии требований пользователей к программному продукту	Адаптация требований к характеристикам внешней среды пользователей	Заказчик или сопровождающий версию программного продукта пользователей	Руководитель, сопровождающий версию программного продукта пользователей

Контроль за рассматриваемыми работами следует проводить посредством процесса совместного анализа руководителем проекта с заказчиком. В конце работ должен быть проведен **анализ риска** [33, 55, 56]. На основании выходных результатов анализа может быть пересмотрена предварительная оценка требуемых затрат ресурсов и с привлечением пользователей или заказчика принято решение о целесообразности перехода к работе по внесению изменений в базовую версию программного продукта. Результатами этой работы являются:

анализ влияния и стоимости изменений; рекомендуемый вариант и согласованные изменения; обновленные и исправленные документы.

Квалифицированные специалисты, управляющие обеспечением качества программных продуктов, должны овладеть стандартами и методиками предприятия, поддерживающими регистрацию, контроль, документирование и воздействия на показатели качества на всех этапах ЖЦ комплекса программ. Они должны быть обучены эксплуатации системы качества проекта, выявлению всех отклонений от стандартов, от заданных показателей качества объектов и процессов, а также от предписанной технологии на промежуточных и заключительных этапах производства.

Результаты испытаний корректировок должны быть документально оформлены и составлять: обновленные планы, документы и процедуры тестирования; измененные исходные программы; отчеты о квалификационном тестировании; характеристики отражающие **стоимость и качество внесенных изменений**. Обновленные документы должны включать подробный отчет о проведенном анализе; обновленные требования; обновленные планы, процедуры и отчеты о тестировании; обновленные учебные материалы. Проверка и приемка модификаций при эксплуатации обеспечивает подтверждение корректности изменений, внесенных в систему в соответствии с принятыми стандартами и по установленной технологии. Сопроводитель должен провести проверки каждого внесенного изменения совместно с заказчиком, утвердившим изменение в целях подтверждения целостности и работоспособности измененной системы и программного продукта (см. табл. 7.1).

Результатами данной работы являются: **новая версия программного продукта**, включающая в себя принятые изменения; отклоненные изменения; отчет о приемке версии; отчеты о проверках и аудитах; отчет о квалификационном тестировании программного продукта. Квалифицированные руководитель и специалисты должны **документально оформить** и представить заказчику:

- отчеты о дефектах и предложения о модификациях; результаты их анализа, трудоемкость и варианты реализации изменений;
- результаты приемочных испытаний, верификации, аттестации и измерений характеристик качества новой версии программного продукта;

- отчеты об обеспечении характеристик качества программного продукта и результаты приемо-сдаточного тестирования;
- комплект актуальных проектных документов и документов результатов сопровождения;
- официальные рекомендации с указаниями о целесообразных последующих модификациях и создании новых версий программного продукта.

Квалифицированные специалисты должны оценивать затраты на сопровождение и корректировки версий программных продуктов, включающие следующие основные *составляющие* [28]:

- затраты на тестирование для обнаружения и устранения ошибок и дефектов в каждой версии программного продукта;
- затраты на доработку и совершенствование программ, формирование и испытание новых модернизированных версий;
- затраты на конфигурационное управление, тиражирование каждой новой версии и ее внедрение в эксплуатируемых и новых системах.

Доля каждой составляющей в общих затратах на сопровождение может значительно изменяться в зависимости от особенностей сферы применения и жизненного цикла конкретного программного продукта. Для долгоживущих (~ 10 лет), многократно тиражируемых (1000 – 100000 экземпляров) комплексов доминирующими обычно являются затраты на модернизацию и доработку версий программного продукта. Затраты на модернизацию зависят от тиража косвенно, вследствие расширения условий применения конкретного комплекса и увеличения потока запросов пользователей на развитие программ. Так же косвенно влияет тираж на запросы для устранения выявленных ошибок.

Затраты на совершенствование и модернизацию комплексов программ близки по содержанию (но не по величине) к затратам на их первичную разработку. Модернизация обычно производится поэтапно. Для каждой новой версии изменяется (разрабатывается) только некоторая часть от всего объема программного продукта. Экспериментально установлено, что эта часть при вводе очередной версии обычно составляет 10 – 20% от размера всего комплекса [5]. Сложность связей компонентов приводит к тому, что удельные затраты на изменяемые программы при модернизации каждой версии могут быть

в 2 – 3 раза больше, чем затраты на создание компонентов программ **такого же объема** при разработке первой версии.

Финансирование сопровождения, тестирования и испытаний версий программного продукта целесообразно определять специальным разделом договора между специалистами – разработчиками и заказчиком на производство версии программного продукта [8, 16, 33]. В техническом задании и в контракте следует **уметь определять порядок квалификации видов и причин изменений** в программах при испытаниях, а также распределение ответственности за их инициализацию, реализацию и финансирование. Выявленные ошибки в компонентах и комплексе программ, которые искажают реализацию функций, согласованных руководителями проекта с заказчиком в контракте и требованиях спецификаций, а также отраженные в документации на версию, должны устраняться за счет разработчиков. Модификацию и расширение функций и характеристик компонентов или создание новых версий программного продукта, ранее не отраженных в требованиях технического задания и контракте с заказчиком, следует квалифицировать как дополнительную работу с соответствующим финансированием заказчиком.

Снятие программного продукта с эксплуатации и сопровождения должно быть подготовлено анализом, обосновывающим это решение. В анализе **квалифицированному руководителю** следует **уметь определить и экономически обосновать** возможность сохранения устаревшей версии комплекса программ, а также необходимость производства и применения новой версии программного продукта. Перед прекращением сопровождения следует определить влияние снятия программного продукта с сопровождения на пользователей, установить программный продукт, заменяющий снимаемый и определить обязанности по любым оставшимся вопросам последующей поддержки применения продукта. В содержание уведомления необходимо включить: описание заменяющего или модернизированного продукта с указанием даты его доступности для пользователей; объяснение того, почему прежний программный продукт нельзя больше поддерживать и применять; описание других доступных вариантов поддержки в случае прекращения сопровождения. Должны быть предусмотрены возможности доступа к архивным данным снятой с сопровождения версии программного продукта.

Требования к профессиональной квалификации специалистов, управляющих конфигурацией крупных комплексов программ

Основной задачей управления конфигурацией является документальное оформление и обеспечение полной наглядности текущей конфигурации продукта производства и выполнения требований к функциональным характеристикам комплекса программ. Другая задача заключается в том, чтобы все лица, работающие над комплексом программ, в любой момент его жизненного цикла использовали достоверную и точную информацию о состоянии проектирования и производства продукта и его реализации. Управление конфигурацией следует организовать так, чтобы каждый специалист, участвующий в производстве, знал свои обязанности и имел достаточно независимости и полномочий для выполнения поставленных задач [28, 50, 67]. Четкая, иерархическая структура документов комплексов программ позволяет использовать конфигурационное управление для решения задач *синтеза – конструирования и модификации конфигураций при производстве программного продукта*, состава и взаимодействия компонентов в процессе сборки и формирования его версий. Регистрация и учет истории этого процесса обеспечивает возможность его контроля и пошагового восстановления выполненных изменений (отката) для выявления вторичных дефектов, внесенных в процессе корректировок версий комплекса программ и его компонентов.

Политика, виды деятельности и квалификация специалистов и правила в области управления конфигурацией, характерные для конкретного проекта, должны быть *определенны руководителями в Руководстве по управлению конфигурацией проекта*. В этом Руководстве могут содержаться ссылки на процедуры управления конфигурацией, описанные в стандарте конкретного предприятия. Структура продукта должна определять взаимосвязь и положения объектов конфигурации при разбиении продукта на компоненты. *Объекты конфигурации должны* выбираться с помощью процесса разложения. Этот процесс, идущий сверху вниз, разделяет общую структуру продукта на логически связанные между собой и соподчиненные компоненты, выбираемые для управления конфигурацией. Выбор слишком большого количества объектов конфигурации отрицательно влияет на наглядность модификации продукции, затрудняет управле-

ние и увеличивает расходы. Выбор слишком малого количества объектов конфигурации или недостаточная дробность разложения создает трудности для технического обслуживания, а также ограничивает возможности модификаций и управления конфигурацией. Все необходимые функциональные и физические характеристики объекта конфигурации, включая сопряжения, изменения, отклонения и разрешения на отклонения, должны содержаться в четко идентифицированных документах. Для идентификации объектов конфигурации, их деталей, документов, сопряжения, изменений, отклонений и разрешений на отклонение следует установить и применять систему правил нумерации.

Определение конфигурации комплекса программ состоит в создании схемы обозначения программных компонентов и их версий (объектов программной конфигурации), которые контролируются при реализации проекта. Для каждого программного компонента и его версий должны быть определены: документация, в которой фиксируется состояние его конфигурации; эталонные версии и другие элементы обозначения. Для того чтобы управление конфигурацией было экономически эффективным, следует **уметь определять его организационную структуру и требования к квалификации специалистов** (рис. 7.2). Эта структура обычно ориентируется на проект и при необходимости адаптируется, чтобы отвечать потребностям различных этапов его жизненного цикла.

Управление конфигурацией является производственным процессом применения административных и технических процедур **квалифицированными руководителями и специалистами на всем протяжении жизненного цикла** крупных программных комплексов для:

- обозначения, определения и установления состояния и изменений версий программных продуктов и их компонентов в системе;
- управления изменениями и выпуском версий программных компонентов и продуктов;
- описания и сообщения о состояниях продуктов, компонентов и заявок на внесение изменений в них;
- обеспечения полноты, совместимости и корректности компонентов продуктов;
- управления хранением, обращением и поставкой программных продуктов.

***Квалифицированные руководители и специалисты
для управления конфигурацией крупных комплексов программ
должны знать и уметь:***

- сформировать Руководство по управлению конфигурацией компонентов и комплекса программ и обучить специалистов конкретного проекта его применению;
- организовать разбиение комплекса программ на объекты конфигурации, их идентификацию и учет версий компонентов;
- прослеживать корректность изменений компонентов и комплекса программ сверху вниз и снизу вверх;
- управлять запросами на изменения и корректировку версий компонентов и комплекса программ;
- проводить сборку и формирование версий конфигурации программного продукта;
- разработать совокупность поэтапных отчетов: о дефектах и ошибках, запросах на модификации, выполненных и утвержденных корректировках версий компонентов и продукта;
- структурировать и создавать базу данных и документацию о состоянии и изменениях конфигурации версий программного продукта;
- проводить регулярные проверки и аудит корректности конфигураций очередных версий программного продукта;
- осуществлять утверждение заказчиком и выпуск версий конфигурации программного продукта;
- тиражировать и архивировать при производстве версии программного продукта и документы в базе данных управления конфигурацией;
- предотвращать рост «энтропии» комплекса программ и увеличения сложности устранения дефектов и ошибок;
- оценивать ущерб для предприятия и его имиджа от рекламаций, дефектов и ошибок в поставляемых программных продуктах.

Рис. 7.2

Конфигурационное управление включает действия и средства, позволяющие устанавливать категории, статус и личности руководителей и специалистов, которые **правомочны ответственно определять целесообразность и эффективность изменений**, а также техническую реализуемость корректируемых версий с учетом ограничений бюджетов и сроков (см. таблицу 7.1). Решения об изменениях

должны приниматься на достаточно высоком уровне заказчика и ответственного руководства проектом, способным оценить их влияние на концептуальную целостность и качество всего программного продукта и системы.

Концепция конфигурационного управления конкретным проектом должна предусматривать возможность *анализа и отслеживание изменений иерархической структуры – конфигурации*, программных комплексов и их компонентов как сверху вниз, так и снизу вверх. Первая задача состоит в обеспечении пошаговой детализации сверху вниз возможных причин конкретных дефектов (проявлений *вторичных ошибок*) или неэффективности функционирования программы, для обнаружения их источника (*первичной ошибки*). Вторая задача при движении снизу вверх должна обеспечивать проверку корректности взаимодействия связанных корректировок и сохранения концептуальной целостности и качества комплекса программ и/или его компонентов. Работы специалистов по *прослеживанию изменений должны включать*:

- подтверждение того, что идентифицированы затронутые изменениями компоненты конфигурации;
- оценку воздействия изменений на требования качества, надежности и безопасности и на обеспечение обратной связи к процессу оценки качества функционирования системы;
- оценку дефектов или предлагаемых изменений, и решения о действиях, которые следует предпринять для их коррекции;
- обеспечение обратной связи от отчетов по дефектам и тестам, контроля изменений и корректирующих действий к затронутым изменениями процессам и объектам.

Конфигурационные базы должны быть установлены руководителями официальным *соглашением с заказчиком* на определенные моменты времени и использованы в качестве отправных точек для официального контроля за конфигурацией программного продукта. После первоначального выпуска документов на конфигурацию комплекса программ все изменения следует контролировать. Влияние изменения, требования заказчика и подвергнутая влиянию конфигурационная база должны определять степень формальности, соблюдаемой при работе с изменением, и могут стать основой любой системы классификации, используемой для распределения по категориям данного изменения.

Контроль конфигурации включает следующие **виды деятельности квалифицированных специалистов**, которые должны быть подробно описаны в документированной процедуре контроля за изменениями:

- документирование и обоснование изменения;
- обеспечивать целостность компонентов конфигурации и базовых версий, а также защиту их от некорректных изменений;
- оценивание последствий модификаций;
- внесение и проверка достоверности реализации изменений;
- при проведении работ по внесению изменений, модифицированы и обновлены документы жизненного цикла комплекса программ, на которые эти изменения могут влиять;
- утверждение изменений.

Для каждого изменения следует отслеживать аудиторские проверки, посредством которых анализируется каждое изменение, его причина и разрешение на его внесение в программный продукт. Должны быть выполнены контроль и аудиторская проверка всех доступных контролю программных компонентов, которые связаны с критическими функциями безопасности или защиты.

Учет состояний конфигурации включает подготовку квалифицированными специалистами протоколов управления и отчетов о состоянии, которые отражают хронологию изменения контролируемых программных объектов, состояние их конфигурации. Отчеты о состоянии должны учитывать количество изменений в данном проекте, последние версии программных компонентов, обозначения выпущенных версий, количество выпусков и сравнения программных объектов различных выпусков. Оценками конфигурации должны быть определены и подтверждены специалистами: функциональная законченность программных объектов с точки зрения реализации установленных к ним требований; физическая завершенность программных объектов с точки зрения реализации в продукте всех внесенных изменений.

Составление отчетов о статусе компонентов конфигурации должно начинаться специалистами с того момента, как будут получены первые данные о конфигурации. Отчеты о статусе конфигурации должны предоставлять информацию о всех идентификациях компонентов конфигурации и всех отклонениях от установленных конфигурационных баз. Это позволит прослеживать изменения конфигура-

ционных баз. Записи и отчеты должны стать побочным результатом деятельности по идентификации и контролю. Проверки конфигурации следует проводить до принятия конфигурационной базы с целью гарантии того, что продукт соответствует контрактным или установленным требованиям и что она точно отражена в документах по конфигурации. Проверка конфигурации может потребоваться для официальной приемки заказчиком конфигурации программного продукта.

После оценки изменения *уполномоченный специалист или группа лиц* должны проанализировать документы и решить вопрос об *утверждении или не утверждении изменения* (см. таблицу 7.1) Решение об утверждении должно быть документировано и разослано соответствующим лицам. Каждая единица конфигурации должна быть идентифицирована, документирована и выпущена ее официальная версия до того, как осуществлено производство программных продуктов для пользователей. Цель работ по архивированию и получению документов – обеспечить получение документов, которые необходимы для возможности копирования, повторной генерации, повторного тестирования и модификации программного продукта (см. рис. 7.2).

Регистрация и учет истории процесса корректировки конфигурации должен обеспечивать возможность ее контроля и пошагового восстановления выполненных изменений (*отката*) при выявлении вторичных дефектов, внесенных в процессе производства модификаций для очередной версии программного продукта. Такие дефекты обычно обусловлены одновременным, не скоординированным внесением групп изменений несколькими специалистами или потерей некоторых изменений в определенной версии программного продукта. Это возможно при одновременной разработке или корректировке различных версий компонентов, предназначенных для использования в различных, но определенных версиях сложных комплексов программ.

Для реализации на практике приведенных производственных процедур, требований и планов управления конфигурацией программных комплексов необходимо *уметь организовать мероприятия*, гарантирующие участникам проектов определенную *культуру и квалификацию специалистов, дисциплину разработки и выполнения модификаций*. Такая организационная и производственная сис-

тема должна обеспечивать специалистам разной квалификации и роли в проекте возможность взаимодействия при решении требуемых комплексных задач для накопления, хранения и обмена упорядоченной информацией о состоянии и изменениях компонентов производства программного продукта.

Отчеты о дефектах и корректирующих действиях, связанные с процессами жизненного цикла комплексов программ, целесообразно фиксировать в отдельных подсистемах конфигурационного управления:

- должно быть подготовлено сообщение о каждом дефекте и его реализации, отказовые ситуации или аномальное поведение программного продукта, а также о предпринятых корректирующих действиях;
- отчеты о дефектах и отказовых ситуациях должны предусматривать идентификацию затрагиваемых компонентов, состояние сообщений о дефектах, утверждение и закрытие сообщений о дефектах после корректировок;
- сообщения о дефектах и отказовых ситуациях, для которых требуется корректирующие действия в программном продукте или выходных данных процессов жизненного цикла, должны активизировать производственные работы по выполнению и контролю изменений.

Каждый дефект или модификация должны быть классифицированы по категориям и приоритетам, тесты и корректирующие действия должны быть оценены, чтобы определить были ли дефекты устранены, а изменения комплекса программ были правильно выполнены без внесения дополнительных дефектов. Результаты анализа и предложения необходимо передавать для конфигурационного управления в унифицированной форме **Отчетов специалистов о выявленных дефектах и предложениях по корректировке**, которые должны содержать:

- подробное описание сценария функционирования программного продукта и исходных данных, при которых выявлен отказ или дефект;
- описание проявления отказовой ситуации или дефекта и документы результатов его регистрации;
- предположение о причине, вызвавшей проявление отказа или дефекта;

- предложение по корректировке требований к программному продукту и его компонентов для устранения дефекта и/или совершенствования функциональной пригодности и применения программного продукта.

Для каждого изменения должны документироваться содержательная аннотация, а также общие характеристики и достигнутое на испытаниях качество очередной версии программного продукта. Для экономически эффективной реализации процессов сопровождения и конфигурационного управления комплексами программ они должны быть поддержаны **службой тиражирования, архивирования и гарантированного хранения** всей необходимой информации о документах, их корректировках, версиях, авторах и их правах на изменения. Эта служба должна обеспечивать поставку пользователям только утвержденных копий версий программного продукта и/или их компонентов с полным, адекватным комплектом эксплуатационных документов, а также извещений на частные изменения конкретных, ранее приобретенных версий. Для гарантированного сохранения состояния и результатов модификаций требований к программам и обеспечения возможности их анализа на любой стадии проекта, а также **отката по истории** выполненных корректировок требований, следует организовать четкую **систему хранения и копирования подлинников, дубликатов и копий** программного продукта и документов.

Технической основой управления конфигурацией комплексов программ при производстве являются **системы управления базами данных** (СУБД), адекватные целям и функциям проектов, структурированные по целям, назначению и содержанию данных в выделенных подсистемах [24, 35, 42]. Они должны обеспечивать возможность управления организационной и производственной деятельностью коллективов специалистов, универсальное хранилище в них необходимых данных с инструментами наполнения, корректировки, поиска и контроля информации, соответствующей их профессиональной деятельности. Должны быть упорядочены деловые **коммуникации между специалистами** разных категорий, управление динамическими процессами выполнения изменений и транспортировки корректировок между подсистемами в соответствии с целями их использования специалистами.

Руководителями должна быть разработана **архитектура системы технологического обеспечения** управления конфигурацией ком-

плекса программ при производстве, а также *Руководство по ее применению специалистами*. Выбранная СУБД должна быть адаптирована на управление основными взаимодействующими подсистемами базы данных с учетом класса и масштаба проекта. По мере развития жизненного цикла комплекса программ подсистемы базы данных управления конфигурацией должны поэтапно *заполняться реальными данными* от заказчика и специалистов соответствующих квалификаций и контролироваться руководителями производства. При этом следует управлять динамикой процессов реализации процедур модификации, регистрировать реальное использование ресурсов специалистов, текущее время и график выполнения процедур развития и оформления изменений в подсистемах базы данных.

Эта *информация в подсистемах базы данных управления конфигурацией* должна быть защищена от случайных и преднамеренных искажений путем организованного санкционирования, дублирования и контроля модификаций, истории их создания и изменения в процессах жизненного цикла комплекса программ. Необходимо специалистам гарантировать сохранность версий изменений с учетом их важности для результатов всего производства продукта. Особенno защищенным от искажений и разрушения следует сохранять *архив версий программных продуктов*, прошедших успешные испытания и утвержденные заказчиком. Для устранения дефектов, реализации корректировок и ошибок при развитии новых версий целесообразно выделять рабочую копию предшествовавшей версии и архив накопленных изменений, обеспечивающих возможность *«отката»* к предыдущей версии в случае разрушительных некорректных изменений в процессе разработки новой версии программного продукта или ее компонентов.

Такая *система обеспечения информацией процессов управления конфигурацией* квалифицированным руководителем производства может быть структурирована в соответствии с адаптированной версией жизненного цикла конкретного комплекса программ. В соответствии с *задачами специалистов при производстве* основные *подсистемы базы* данных информационного обеспечения управления конфигурацией должны быть ориентированы на определенные процессы, квалификацию специалистов и компоненты комплексов программ:

- на заказчика и руководителя проекта при формировании и хранении требований к программному продукту и его крупным функциональным компонентам;
- на компоненты, тесты и результаты тестирования подсистем комплекса программ программистами, тестировщиками и разработчиками функциональных программ;
- на дефекты, ошибки, запросы на модификации и корректировки компонентов и комплекса программ;
- на руководителей функциями, модификациями, корректировками и версиями программного продукта;
- на заказчика и руководителей требованиями, комплексами программ и эксплуатационной документацией версий программного продукта, утвержденных и доступных для поставки пользователям и для архивирования.

Для каждой подсистемы целесообразно выделять достаточно автономную базу данных компонентов с ограниченным доступом только для определенных категорий специалистов (см. таблицу 7.1). Эти фрагменты базы данных могут быть построены на стандартизированной основе СУБД поддержки производства, взаимодействовать с аналогичными по структуре компонентами базами данных. Они должны накапливать и содержать основные компоненты и документы результатов производства на соответствующем уровне жизненного цикла комплекса программ. Интерфейсы этого взаимодействия компонентов базы данных должны быть стандартизированы, по возможности *ограничены для некоторых категорий специалистов* по объему и доступности текущей и отчетной информации. Для каждого сложного проекта комплекса программ целесообразно уметь оформлять и утверждать *Руководство и схему базы данных, обеспечивающей управление конфигурацией фрагментов комплекса программ, его требований и тестов*, а также *категории ответственных лиц* за их поэтапную реализацию, контроль и сохранность информации всего программного продукта.

Экономическая эффективность процессов производства и совершенствования версий программного продукта в значительной степени определяется их квалифицированной организацией и информационным обеспечением базы данных конфигураций [28, 50]. Выбор технологии производства комплекса программ и СУБД в значительной степени зависит от перспективы его последующего сопрово-

ждения и совершенствования. Контроль среды поддержки жизненно-го цикла управлением конфигурацией должен обеспечивать гарантию того, что инструментальные средства, используемые для создания и модификации программ, идентифицируются, контролируются и могут быть получены из соответствующих источников.

Программные комплексы зачастую сначала проектируются с одним видением задач, однако когда возрастают требования заказчи-ка, окружение и мир меняются, а когда меняются люди, их взгляды может не разделять новый руководитель или заказчик проекта. Когда возникают эти фундаментальные события, целесообразно еще раз об-думательности вносятся мелкие изменения, чтобы комплекс программ отвечал требованиям, в результате **система становится более сложной**, менее понятной, с дополнительными ошибками. Все-гда есть неожиданные препятствия: большее, чем ожидалось число ошибок, плохое конструктивное решение, новое требование, ошибка, которая упорно не хочет устраниться и результат – нарушение графи-ка производства. Когда времени нет, все формальности отбрасываются, поскольку руководители и поставщики не могут ждать. Конечный эффект разработки, зависящей от объективных факторов сировой действительности и процессов изменения программного продукта приводит к снижению понятности специалистами функций и струк-туры программного продукта к **росту «энтропии» комплекса про-грамм** и к увеличению сложности устранения ошибок.

Любое изменение требований и состояния программного про-дукта может **потребовать пересмотра соглашений** с заказчиком о функциональности, качестве, сроках и бюджете, присущих разработ-ке всей системы. Группы специалистов управления конфигурацией часто становятся частью такого процесса и организуют форум для от-крытых дискуссий и поиска компромиссов с заказчиком по поводу ранее принятых решений, которые могут оказывать **воздействие на риски программного продукта**.

Недостаточная профессиональная квалификация специали-стов при изменениях программ может отражаться дорогими негатив-ными и даже катастрофическими последствиями от применения про-граммных продуктов. Влияние ошибок в сложных комплексах про-грамм можно оценивать с двух позиций [49]: **ущерба от проявления не устраниенных ошибок и дефектов** при применении программных

продуктов по назначению и затрат, необходимых для их обнаружения и исключения. Величина необходимых затрат на сокращение ошибок до допустимых пределов, более или менее, предсказуема и изменяется в относительно небольшом диапазоне на каждую ошибку. Однако ущерб от не выявленных ошибок программ в некоторых системах может отражаться огромными материальными потерями и даже гибелью людей или долго оставаться не замеченным.

Людям свойственны *человеческие слабости и психологические ошибки* [13, 43, 49]: они ошибаются, поскольку забывчивы и легко отвлекаются, а также слишком ленивы, чтобы напечатать еще несколько строк кода для проверки правильности входных данных для некоторых функций. Зачастую документация не соответствует реализации программ или создаются бесполезные документы. Если требования меняются постоянно, становится ясно, что сами *заказчики не понимают, что им нужно*, и данная проблема для них слишком сложна. Они вынуждены возвращаться и ставить новые требования, это означает, что они не компетентны и упустили несколько необходимых свойств. В процессе разработки короткие, простые пути и компоненты имеют свойство повышать сложность решений и уменьшать понятность комплекса. Недостаток знания предметной области руководителями и специалистами может оказаться существенным фактором, вносящим *вклад в повышение сложности продукта и ошибок* в принимаемые решения. Заказчики постоянно добавляют новые свойства и требования, в результате элегантная архитектура и конструкция разрушается и накладываются свойство на свойство, не думая о сохранении структуры комплекса и возможности ее контроля.

Ошибки часто проявляются, когда производятся изменения в программах, есть строгая корреляция между числом изменений и улучшений, внесенных в комплекс программ, и числом дефектов в нем. Когда обновляется программный компонент для устранения существующей ошибки, меняются ошибки, которые известны, на *новые не известные ошибки*. Поскольку ошибки разрастаются в изменяющейся среде, высок шанс того, что будет внесена ошибка при корректировке компонента. Жизненный опыт свидетельствует, что вероятность внесения одной или более *дополнительных ошибок* при попытке исправить дефект составляет *от 15 до 50%* в зависимости от ситуации. В сложных комплексах программ шанс внести серьезную

новую ошибку при исправлении первоначальной ошибки достаточно велик, поэтому не все первоначальные ошибки необходимо корректировать.

Существует сильная **корреляция между предыдущим и текущим уровнем дефектов** в ходе эволюции программного компонента. «**Ошибки привлекают ошибки**» – когда устраняется ошибка в компоненте, можно внести другую ошибку в тот же компонент. Через некоторое время такие компоненты становятся известными в качестве склонных к ошибкам. Поэтому иногда легче и дешевле применить тактику полной перестройки и замены компонента, удаляя все возможные ошибки путем уничтожения мест их обитания, полностью перестроив и заново реализовав весь компонент.

Каждая **ошибка уникальна и специфична**, но, в то же время, каждая ошибка может повторять аналогичные. В борьбе за то, чтобы сделать программы функциональными, работающими и полезными, их враги – ошибки появляются в разных видах (см. лекцию 8). Есть ошибки, которые возникают в виде неожиданных полезных свойств и ждут наиболее благоприятного момента, чтобы обмануть доверие. Некоторые ошибки легко предотвратить, предприняв определенные меры предосторожности. Другие легко создать, но трудно обнаружить.

Если обнаруживаются ошибки, то это означает, что **существуют причина, набор условий и квалификация специалиста**, которые ответственны за их проявление. Если воспроизвести эти условия, то можно проявить соответствующие ошибки. Ошибки, которые представляются невоспроизводимыми, только кажутся таковыми, так как в действительности недостаточно возможностей для имитации функциональной среды и определения входной информации для комплекса программ. Некоторыми **причинами невозможности воспроизвести ошибку** могут быть [49]:

- пользователи могут применять продукт не так, как предписано в документации и когда сообщают об ошибке, могут не включить некоторые важные факты, поэтому при работе по инструкциям для воспроизведения ошибки не удается воссоздать проблему;
- программный продукт может работать в среде реального времени и использовать данные, зависящие от времени, вследствие чего невозможно точно охватить все входные данные даже для того, чтобы подтвердить, что ошибки нет, поэтому без специального диагности-

ческого динамического компонента трудно воспроизвести ошибку;

- в многопоточной системе, где важны временные соотношения, трудно создать точную последовательность входных событий с такими же временными характеристиками и тем самым воспроизвести ошибку.

Однако перечисленное не означает, что ошибки невоспроизведимы. Иногда цена воспроизведения ошибки может быть слишком высока. В других случаях трудно понять и представить все взаимосвязи причин, чтобы воспроизвести конкретный случай. Однако без воспроизведения невозможно показать, что известна причина ошибки и что она удалена.

Ущерб от ошибок, который часто не замечается, влияет на **моральный дух** специалистов, связанных с программным продуктом. Никому не понравится, если его предприятие будут связывать с низкокачественным продуктом. Бесконечные ошибки в программном продукте – оскорбление нашей веры в собственный интеллект и квалификацию, понижает моральный дух коллектива специалистов – от руководителей проекта до программистов, испытателей и персонала, работающего с покупателями. Каждый, кто работает в программной индустрии, знает, что **некоторое число ошибок будет в его продукте**. Коллектив специалистов ожидает, что они будут проводить **большую часть времени в борьбе с ошибками** до выпуска очередной версии продукта. Ошибки могут становиться деморализующей силой, когда их число не уменьшается и особенно когда предполагается, что уже исправлена ошибка, а она возникает снова.

Люди устают от постоянной работы с одними и теми же проблемами и дефектами. Квалифицированные специалисты хотят работать с чем-то новым, разрабатывать самые современные, классные продукты, используя современные технологии и инструменты. Испытатели иногда наблюдают, как число выявляемых ошибок растет. Устанавливается чувство разочарования, в результате уход в другое предприятие или переход на другой проект становится очень привлекательной альтернативой.

Результатом **деморализации коллектива становится**: низкая продуктивность и высокая текучесть кадров [49]. Вместо того чтобы стараться работать в меру своих самых лучших способностей, специалисты коллектива будут искать пути, чтобы им вообще не приходилось сталкиваться с дефектами. Хотя такие пути могут оказаться

действенными на короткие сроки, в перспективе они будут стоить еще дороже: вносятся новые дефекты, а симптомы существующих проблем становятся скрытыми. С другой стороны, решение простой проблемы может тянуться многие дни, потому что никому не интересно этим заниматься.

Программная инженерия является уникальной в том, что заказчиков и пользователей приучили рассматривать **проблемы с программным продуктом как нормальный производственный процесс**. Кроме того, когда выпускаются дефектные продукты, пользователей заставляют платить за программные заплатки и устранение ошибок. Как и в других организациях, успех предприятия, занимающегося программным продуктом, зависит от многих факторов. Одна из ключевых сил в маркетинговой компании – **репутация марки (имида)**. Эти ассоциации не рождаются просто так, и крупная ошибка в продукте на стадии становления имени предприятия может причинить непоправимый ущерб. Другая ключевая сила на рынке программного продукта – это зависимость от производителя, когда покупатель привязывается при покупке продукта к определенному поставщику из-за зависимости от формата данных и знакомого интерфейса.

Барьер, который нужно преодолеть, чтобы выйти на рынок низок, и марка предприятия может помочь только войти на этот рынок. У покупателей нет причин оставаться лояльными к низкокачественным продуктам и предприятиям. Если покупатель расстался с поставщиком, шанс вернуть его назад незначительный, поскольку он не заслуживает доверия. Усилия, прилагаемые для привлечения покупателя, всегда выше, чем для его сохранения. **Плохой имидж** означает, что предприятие должно тратить больше для привлечения покупателей. Продукт, содержащий ошибки, имеет прямое влияние на конечный экономический баланс предприятия. Ошибочный программный продукт очень трудно обслуживать. Высоконадежный продукт может быть в 2 раза дороже, чем малонадежный, однако последний гораздо дороже в обслуживании.

Обширный **обзор примеров** около десятка критических программных ошибок с **большим ущербом** представлен в [49], из которого можно сделать вывод, что такие ошибки не так уж редко проявляются при эксплуатации и весьма разнообразны по типам и происхождению. В частности, чтобы успокоить своих покупателей после проявления ошибки в процессоре Pentium, фирма Intel была вынуж-

дена заменить все дефектные процессоры, что ей обошлось в \$400 млн. В другом примере из-за того, что кто-то забыл проверить перевод измерительных единиц в модуле, полет космического аппарата NASA Mars Climate Orbiter закончился образованием еще одного кратера на поверхности Марса. Система Therac-25 – наиболее часто упоминаемый в литературе пример, когда предельной ценой ошибки оказалась потеря жизни человека. Программа, разработанная для лечения пациентов, вызвала передозировку радиации. В результате программной ошибки погибли три пациента.

Лекция 8

ТРЕБОВАНИЯ К ПРОФЕССИОНАЛЬНОЙ КВАЛИФИКАЦИИ СПЕЦИАЛИСТОВ, ОБЕСПЕЧИВАЮЩИХ КАЧЕСТВО ПРОГРАММНЫХ ПРОДУКТОВ

Требования к качеству программных продуктов, определяющие необходимую профессиональную квалификацию специалистов при их производстве

Требования к необходимой квалификации руководителей и специалистов зависят от заданных заказчиком свойств и характеристик *качества программных продуктов*. Эти характеристики при производстве целесообразно основывать на международном стандарте ISO 9126. Высшие приоритеты в конкретных проектах, естественно, должны присваиваться свойствам и характеристикам качества, необходимым для достижения *стратегических целей* использования назначения и функций программного продукта. Остальные стандартизованные характеристики продукта, в той или иной степени, должны способствовать обеспечению *тактических целей* выбранных конструктивных требований качества с учетом доступных ресурсов. Поэтому им могут быть присвоены более низкие приоритеты и на выполнение соответствующих требований заказчика, могут *выделяться руководители и специалисты меньшей квалификации*. Кроме того, для их реализации могут сокращаться ресурсы, что, в частности, может отражаться на не полной реализации некоторых формализованных заказчиком требований. Стандартом рекомендуется, чтобы было предусмотрено измерение каждой характеристики с точностью и определенностью, достаточной для сравнений с требованиями технического задания и спецификаций, и чтобы измерения были объективны и воспроизводимы – рис 8.1.

Системная эффективность целевого применения программных продуктов определяется степенью удовлетворения потребностей определенных лиц – заказчиков и/или пользователей. В стандартах

эта эффективность отражается, прежде всего, основной, обобщенной характеристикой качества – функциональной пригодностью программного продукта, которая рассмотрена в лекции 3.

Квалифицированные руководители и специалисты для реализации требуемых характеристик качества программных продуктов должны знать и уметь:

- выбирать в производственные коллективы профессиональных специалистов, способных реализовать программный продукт с заданными характеристиками;
- устанавливать приоритеты характеристик качества, удовлетворяющие стратегическим целям программного продукта;
- сбалансировать требования к характеристикам качества программного продукта в условиях ограниченных ресурсов;
- обеспечивать и контролировать правильность – корректность компонентов и комплексов программ при проектировании и производстве продукта;
- обеспечивать и контролировать способность к корректному взаимодействию компонентов и комплексов программ при проектировании и производстве;
- обеспечивать и контролировать реализацию комплексной политики обеспечения безопасности применения программного продукта при производстве;
- обеспечивать реализацию требований и контроль достигнутой надежности функционирования программных продуктов;
- обеспечивать реализацию требований и контроль достигнутой производительности, пропускной способности, сопровождаемости, мобильности и эффективности использования вычислительных ресурсов продуктом;
- создавать программные продукты в соответствии со стандартными требованиями обеспечения практичности, сопровождаемости и мобильности.

Рис. 8.1

Решение этой задачи ***квалифицированными специалистами*** должно быть направлено на обеспечение требуемой высокой функциональной пригодности продукта ***путем сбалансированного улучшения оставльны, конструктивных характеристик качества в условиях ограниченных ресурсов***. Излишне высокие требования заказчика к отдельным атрибутам качества, требующие для реализации

больших дополнительных трудовых и вычислительных ресурсов, целесообразно снижать, если они слабо влияют на основные, функциональные характеристики. Ориентирами могут служить диапазоны изменения характеристик, границы количественных или качественных величин которых сверху и снизу могут быть выбраны **на основе следующих принципов**:

- предельные значения характеристик качества должны быть ограничены сверху допустимыми или рациональными затратами ресурсов на их достижение при производстве и совершенствовании версий программного продукта;
- наибольшие допустимые затраты ресурсов, например труда и времени для реализации характеристик качества, должны обеспечивать функциональную пригодность применения программного продукта на достаточно высоком уровне;
- допустимые наименьшие значения отдельных характеристик качества могут соответствовать значениям, при которых заметно начинает снижаться функциональная пригодность при применении программного продукта;
- ограниченные значения всех характеристик качества не должны негативно отражаться на возможных высоких значениях других приоритетных характеристик.

Достижение и улучшение каждой **конструктивной характеристики качества** программного продукта, требует **соответствующей профессиональной квалификации специалистов и затрат ресурсов** (трудоемкости, финансов, времени), которые в той или иной степени должны отражаться на основной характеристике – на функциональной пригодности. Эти конструктивные характеристики имеют значение для продукта постольку, поскольку они обеспечивают требуемое качество реализации основного его назначения и функций. При выборе конкретных конструктивных характеристик качества **квалифицированным заказчикам и руководителям** следует учитывать затраты ресурсов на их достижение и на результирующее повышение функциональной пригодности, желательно в **сопоставимых экономических единицах**, в тех же мерах и масштабах. Такое, даже приблизительное, качественное сравнение эффекта и затрат позволяет избегать при производстве многих не рентабельных завышений заказчиком требований к отдельным конструктивным характеристикам, которые не достаточно адекватно отражаются на улучшении функций

продукта. В зависимости от назначения и функций программного продукта почти каждая из **характеристик качества может стать доминирующей** или даже почти полностью определяющей функциональную пригодность программного продукта. Значительные трудности у руководителей и специалистов при формализации требований к характеристикам качества зачастую проявляются при анализе корректности, способности к взаимодействию и безопасности программного продукта.

Правильность – корректность - это способность программного продукта обеспечивать правильные или приемлемые результаты функционирования в соответствии с требованиями заказчика и пользователей. Эталонами для выбора требований к корректности при проектировании могут быть верифицированные и взаимоувязанные требования к функциям комплекса, компонентов и модулей программ, а также правила их структурного построения, организация взаимодействия и интерфейсов. Эти требования при проектировании и производстве должны быть обеспечены и прослежены квалифицированными специалистами сверху вниз до модулей, и использоваться как эталоны при установлении необходимой корректности соответствующих компонентов.

Требования к характеристике **корректность** могут представляться в виде описания двух основных свойств, которым должны соответствовать все программные компоненты и комплекс в целом. Первое требование состоит в выполнении определенной степени прослеживаемости сверху вниз реализации спецификации требований на программный продукт при последовательной детализации и верификации описаний программных компонентов вплоть до текстов и объектного кода программ. Второе требование **заключается в выборе квалифицированными руководителями** степени и стратегии покрытия тестами структуры и функций программных компонентов, совокупности маршрутов исполнения модулей и всего комплекса программ для последующего процесса верификации и тестирования, достаточного для функционирования программного продукта с требуемым качеством и точностью результатов при реальных ограничениях ресурсов.

Способность к взаимодействию состоит в свойстве комплекса программ и его компонентов корректно взаимодействовать с множеством определенных компонентов внутренней и внешней среды. При

выборе и *установлении специалистами* при производстве способности программных и информационных компонентов к взаимодействию ее можно оценивать объемом технологических изменений в комплексе программ, которые необходимо выполнять при дополнении или исключении некоторой функции или компонента, когда отсутствуют изменения операционной, аппаратной или пользовательской среды. При этом важно учитывать обеспечение возможности повторного использования апробированных компонентов и переноса их на различные платформы [44, 50, 52].

Защищенность – безопасность тесно связана с особенностями функциональной пригодности каждого программного продукта [30, 48, 62]. Разработка и формирование заказчиком и руководителем проекта требований к свойствам безопасности программного продукта должны осуществляться на основе потребностей эффективной реализации назначения и функций продукта при различных, реальных угрозах. В процессе проектирования *квалифицированными специалистами* должны быть выявлены потенциальные предумышленные и случайные *угрозы функционированию* и установлен необходимый уровень безопасности применения данного программного продукта. Наиболее полно функциональная безопасность комплексов программ характеризуется величиной ущерба, возможного при проявлении дестабилизирующих факторов и реализации конкретных угроз – рисков, а также средним временем между проявлениями непредумышленных угроз, нарушающих надежность и безопасность. Реализации угроз целесообразно характеризовать интервалами времени между их проявлениями, нарушающими безопасность применения программного продукта, или наработкой на *случайные катастрофические отказы*, которые отразились на безопасности.

Создание такой системы защиты должно предусматривать *планирование и реализацию квалифицированными руководителями целенаправленной политики комплексного обеспечения безопасности*. Потенциальная возможность и реальные проявления информационных диверсий, а также непредумышленных, негативных воздействий [48] заставляет в программные продукты включать средства для обеспечения безопасности применения пользователями всего комплекса функциональных программ в течение его жизненного цикла. Их реализация в критических системах может требовать значительных ресурсов памяти и производительности вычислительных

средств. Реальные ограничения ресурсов, используемых в процессах производства, **ограниченная квалификация специалистов**, изменения внешней среды и требований заказчика объективно приводят к отклонениям процесса функционирования продукта от предполагавшегося. В процессе формирования технического задания следует формулировать **основные положения методологии и план последовательного повышения безопасности** путем наращивания комплекса средств защиты, поэтапных испытаний компонентов и определения характеристик, допустимых для продолжения работ на следующих этапах.

Конструктивные характеристики качества крупных программных продуктов могут быть разделены на две группы: количественные и качественные, которые различаются возможностями конкретизации их **реализации и измерения квалифицированными специалистами**. Две группы стандартизованных характеристик качества программных продуктов – надежность и эффективность в наибольшей степени доступны количественным измерениям, и для их реализации обычно требуется **наиболее высокая квалификация руководителей и специалистов** (см. рис. 8.1)

Надежность: свойства программного продукта обеспечивать достаточно низкую вероятность потери работоспособности – отказа, в процессе его функционирования в реальном времени [5, 20, 62, 64]. Источниками ненадежности являются непроверенные требования и их дефекты, ошибки сочетания исходных данных, при которых функционирующие программы дают неверные результаты или отказы. В результате комплекс программ **не соответствует требованиям функциональной пригодности и работоспособности**. Допустимая длительность прерывания оперативной работы пользователей системы для полного восстановления нормального ее функционирования обычно может составлять несколько секунд или минут. Надежность программных продуктов наиболее полно характеризуется устойчивостью или способностью к безотказному функционированию, а также восстанавливаемостью работоспособного состояния после произошедших сбоев или отказов. Нижняя граница допустимой надежности обычно отражена значениям, при которых резко уменьшается функциональная пригодность, и использование данного типа программного продукта становится неудобным, опасным или нерентабельным. С другой стороны, наилучшие значения этих атрибутов практически **ог-**

раничены реальной квалификацией специалистов и теми ресурсами, которые могут быть выделены для их достижения при производстве и эксплуатации.

Эффективность: в стандарте ISO 9126 отражена двумя характеристиками – временной эффективностью и используемостью ресурсов ЭВМ, которые рекомендуется описывать, в основном количественными атрибутами, характеризующими динамику функционирования программного продукта. Основные требования к эффективности использования вычислительных ресурсов программным продуктом сосредоточены на наиболее критичных показателях производительности и длительности решения функциональных задач. Для систем реального времени особое значение имеют *требования эффективного использования* программным продуктом ресурсов ЭВМ *по производительности*. Основные требования к характеристикам эффективность использования вычислительных ресурсов, возможные дефекты и ошибки, сосредоточены на наиболее критичных показателях производительности и длительности решения функциональных задач. **Временная эффективность:** свойства комплекса программ, отражающие требуемое время обработки заданий, *время отклика* из ЭВМ в систему и/или внешнюю среду после получения типового задания и начала решения требуемой функциональной задачи; а также производительность решения задач с учетом количества используемых вычислительных ресурсов. **Пропускная способность** решения функциональных задач – производительность, число заданий, которое можно реализовать на данной ЭВМ в заданном интервале времени в зависимости от их содержания и числа действующих пользователей или воздействий из внешней среды. При излишне высокой интенсивности поступления исходных данных может нарушаться временной баланс между длительностью решения полной совокупности задач комплексом программ в реальном времени и производительностью ЭВМ при решении этих задач. Для реализации и измерения этих характеристик требуется *высокая профессиональная квалификация руководителя проекта и специалистов*, умеющих сформулировать соответствующие требования и создать условия для их реализации.

Три группы конструктивных характеристик качества программного продукта – практичность, сопровождаемость и мобильность трудно измерять количественно, и они доступны в основном

качественным формулировкам и оценкам их свойств квалифицированными специалистами.

Практичность – применимость: свойства программного продукта, отражающие сложность его понимания, изучения и использования для квалифицированных пользователей при применении в указанных условиях. Требования к практичности и ее характеристикам – понятности и простоте использования, зависят от назначения и функций программного продукта и могут формализоваться требованиями заказчика в виде набора свойств, необходимых для обеспечения удобной и комфортной эксплуатации продукта. Понятность зависит от качества документации и субъективных впечатлений от функций и характеристик продукта. Она должна обеспечиваться корректностью и полнотой описания исходной и результирующей информации, а также всех деталей функций программного продукта для пользователей. Изучаемость можно отражать трудоемкостью и продолжительностью изучения пользователями соответствующей квалификации, методов и инструкций применения программного продукта для полноценной эксплуатации. Для реализации этих свойств необходимы **профессиональные квалифицированные специалисты**, способные задать к ним требования при проектировании программного продукта и создать соответствующую эксплуатационную документацию.

Сопровождаемость: приспособленность программного продукта к модификации и изменению конфигурации определяется **квалификацией архитекторов и руководителей проектирования и производства** комплекса программ (см. лекцию 7). Модификации могут включать исправления, усовершенствования или адаптацию программного продукта к изменениям во внешней среде применения, а также в требованиях и функциональных спецификациях заказчика. Трудоемкость модификаций определяется архитектурой и внутренними метриками качества комплекса программ, которые отражаются на качестве при использовании, а также на сложности управления конфигурациями версий программного продукта. Требования к сопровождаемости количественно можно установить экономическими категориями допустимой трудоемкости и длительности реализации этих задач при некоторых средних условиях, обусловленных необходимостью устранения дефектов и усовершенствованиями функций программного продукта.

Мобильность: подготовленность программного продукта к переносу из одной аппаратно-операционной среды в другую (см. лекцию 4). Установление требований к мобильности может быть сведено **квалифицированными руководителями** к формализации трудоемкости и длительности процессов: адаптации к новым характеристикам пользователей и внешней среды, инсталляции версий программного продукта в среде пользователей и замены крупных компонентов по требованиям заказчиков или конкретных пользователей.

Каждая из перечисленных выше характеристик качества отражает особые свойства и требования к комплексу программ, вследствие чего для их реализации и измерения обычно необходимы **руководители и специалисты адекватной высокой профессиональной квалификации**, которую целесообразно осваивать с учетом особенностей и свойств создаваемых программных продуктов. Кроме того, в зависимости от назначения и категории программного продукта изменяются приоритеты и значения требуемых характеристик и соответственно **профессионализм и уровни квалификации специалистов**, необходимых для их реализации в продукте. Поэтому необходимо управление и координация их работы **квалифицированными руководителями** с целью достижения **необходимого сбалансированного уровня качества и допустимых дефектов – рисков** программного продукта.

Понятия и общие свойства дефектов и ошибок в крупных комплексах программ

При любых видах деятельности людям любой квалификации **свойственно допускать ошибки**. Это всегда необходимо учитывать заказчикам, руководителям и специалистам при проектировании и производстве таких особенно сложных изделий как крупные программные продукты. **Понятие ошибки в программе** подразумевает неправильность, погрешность или неумышленное искажение компонента, процесса или их характеристики качества, что может быть **причиной ущерба – риска** при функционировании и применении программного продукта. При этом предполагается, что **известно или можно предположить** правильное, эталонное состояние объекта или процесса, по отношению к которому может быть определено наличие искажения – дефекта или ошибки. Исходным эталоном для лю-

бого программного продукта являются спецификация требований заказчика или потенциального пользователя, предъявляемых к программам. Подобные документы устанавливают состав, содержание и значения результатов, которые должен получать пользователь при определенных условиях и исходных данных. Любое отклонение результатов функционирования программы от предъявляемых к ней требований и сформированных по ним эталонов-тестов, следует квалифицировать как *дефект – ошибку в программе*, наносящий некоторый ущерб. Различие между требуемыми и полученными результатами функционирования программ могут быть следствием ошибок не только в созданных программах, но и в первичных требованиях спецификаций и в документах, явившихся базой при создании эталонов-тестов. Тем самым проявляется объективная реальность, заключающаяся в *невозможности абсолютной корректности* и полноты исходных требований спецификаций и эталонов для сложных программных продуктов.

Источниками ошибок в программах являются специалисты – конкретные люди с их *индивидуальными особенностями, квалификацией, талантом и опытом* – таблица 8.1. Чтобы делать программы высокого качества, следует учитывать *психологию и свойства* этих людей и понимать, почему они делают определенные ошибки, и как их делают. Только тогда можно выбирать квалифицированным специалистам превентивные меры, создавать методы и средства, которые могут предотвращать или облегчать обнаружение и исправление ошибок [49].

В *понятии ошибки* собрана, *сфокусирована вся психология* [1, 14, 43] – рис. 8.2. Ошибка в системе предполагает ретроспективное исполнение в плане понимания того, что было выполнено ранее практически. Она требует владения приемами деления на отдельные шаги воспоминания о процессе, который шел непрерывным образом. При полном, подробном обдумывании процесса можно говорить о представлении процесса субъектом. Усилия субъекта были направлены на достижение цели, выполнение задачи, на положительное завершение намеченного. Чтобы узнать, что привело к ошибке, желательно знать: какими были движения, мысли и переживания человека в ходе выполнения действия. В определении понятия ошибки *решающее значение имеет время*:

- *прошлое* – ошибочное действие уже совершено, оно всегда

в прошлом, с ошибкой всегда имеют дело как со свершившимся фактом;

- *будущее* – тренировки и профилактические меры направлены в будущее, любое воспитание и обучение направлено на то, чтобы последующие действия не повторяли предыдущих ошибок и позволяли их выявлять.

Таблица 8.1

Типы первичных дефектов и ошибок комплекса программ и их источники

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

Квалифицированные специалисты для обнаружения, локализации и устранения дефектов и ошибок программных продуктов должны знать и уметь:

- учитывать психологические основы появления, формирования и обнаружения ошибок специалистов в сложных системах;
- использовать общие понятия и свойства дефектов и ошибок в сложных комплексах программ для их обнаружения и устранения;
- выделять и различать первичные и вторичные источники и факторы, определяющие дефекты и ошибки компонентов и комплексов программ;
- оценивать и различать измеряемые характеристики качества и риски вследствие дефектов и ошибок в комплексах программ;
- учитывать влияние профессионализма и квалификации специалистов на обнаружение определенных дефектов и ошибок в комплексах программ;
- составлять, систематизировать и обобщать квалифицированные отчеты о характеристиках выявленных и устранимых дефектов и ошибок;
- использовать статистику обнаруженных дефектов и ошибок для распределения ограниченных ресурсов с целью повышения качества программных продуктов;
- оценивать сложность компонентов и комплексов программ для прогнозирования возможного числа дефектов и ошибок, а также ресурсов, необходимых для их обнаружения и исключения;
- оценивать необходимую квалификацию и ограничения коллектива специалистов для успешного обнаружения и сокращения дефектов и ошибок до допустимого уровня в заданные сроки.

Рис. 8.2

При проектировании и производстве систем стараются **оценить вероятность ошибочных действий** персонала. Ошибки будут возникать в наиболее сложных пунктах трудового процесса там, где велика нагрузка и сложность. При ее оценке исходят из возможностей человека воспринимать, помнить, мыслить, из коммуникативных возможностей. Все эти возможности определяются количеством обрабатываемой информации, временными характеристиками обработки, состоянием человека.

Изучение труда профессионалов показывает, что он идет большей своей частью автоматически, а сознательное планирование и исполнение

ние профессионал ведет только в наиболее сложных участках. Именно там вероятнее всего ошибки и ошибочные действия.

Действие предполагает точное исполнение, которое зачастую почти невозможно. Чем сложнее задача, тем с большей вероятностью можно утверждать, что профессионал оказывается на грани, где переход от правильного действия к ошибочному, становится почти неуловимым. Понятие точного исполнения предполагает допуски – пространственные и временные. Пока идет действие, об ошибке говорить нельзя пока действие не закончилось. При анализе ошибок коллектива специалистов должны быть рассмотрены **социально-психологические аспекты трудового процесса**. В ходе поиска звена, связанного с ошибкой группы, следует оценить возможные причины:

- не отвечает ли за ошибку лидер с его стилем и методами руководства;
- не возникла ли ошибка из-за временной несогласованности процессов;
- из-за некоторых особенностей внутригруппового взаимодействия;
- из-за чрезмерного давления группы на индивида и т.п.

Ошибка – это факт, случай из практики. Концепция ошибки должна строиться на основе представлений о нормальном функционировании системы, которое определяется **позитивно**. Сама ошибка должна определяться только **негативно**. В определении понятия ошибки решающее значение имеет время: ошибочное действие уже совершено, оно всегда в прошлом. Столкнувшись с негативным результатом — ошибкой, ищут, что могло привести к нему. Для расследования причин ошибок необходимы соответствующие (в том числе и психологические) методики. Представление о **системе психических процессов** может стать основой для методики анализа ошибок в системе. В экспериментальной психологии установлены различные **типы ошибок** [1, 43]:

- **ошибки восприятия** – не успел обнаружить, не сумел различить, не узнал;
- **памяти** – забыл, не успел запомнить, не сумел удержать в памяти, сохранить, восстановить, воспроизвести;
- **мышления** – не понял, не успел схватить, не предусмотрел, не разобрался, не проанализировал, не объединил, не обобщил, не сопоставил, не выделил;

- **внимания** – не сумел сосредоточиться, собраться, переключиться, удержать, не успел охватить всего, устал.

В соответствии с положениями психологии мыслительный аппарат, управляющий действиями человека, выступит **как система**, в состав которой входит множество операций, разнообразных по качеству и по степени владения ими субъектом. **Понятие «субъект»** встает сразу же, как только сталкиваются с ошибкой системы на практике. Расследующие причины ошибки имеют дело с человеком, совершившим действие. Реконструируя ситуацию исполнения действия, имеют дело уже не с действием, а с человеком. Он не совершает действия, а участвует в реконструкции совершенного в прошлом. От него получают сведения о действии, но он же и скрывает,вольно или невольно, то, что происходило тогда, когда он совершал действие. Теперь же есть только человек и прошлое, уже совершенное действие.

Творя действие, субъект **чувствует действие «изнутри»**, чувствует движения, сопротивление объектов. Субъект запоминает эти чувствования, ощущения и сохраняет их в себе, в памяти, в структурах опыта, используя в дальнейших действиях. Субъекту даны различия между намеченными шагами действия и достигнутыми результатами. Эти его чувствования, они неповторимы, не воспроизводимы, но они его. В момент разбора совершенного действия эти особенности субъекта выступают особенно ярко. Субъект прежде исполнял действие, а теперь его заставляют вспоминать действие, думать о совершенном действии, когда уже ничего не исправишь, с сожалением понимая, что нужно было сделать иначе. Люди, в том числе и те, кто не присутствовал при исполнении самого действия, стараются **пройти по цепи звеньев вспять** и найти то звено, в котором произошел сбой – **ошибка**. Они исходят из некого представления о действии как о конструкции, которую можно расчленять.

В прошлом, исполняя действие, субъект был **устремлен на результат**, чувствовал тонкие оттенки ощущений, которые плавно переливались одно в другое, были слиты с контекстом ситуации и смыслом действия. Действие было дано субъекту как непрерывный процесс, как целое. Теперь люди говорят об этом, членя на части то, что было прежде живым целым. Субъекта не устраивает то, как проходит расчленение, каким способом членят на части целое, где проходят границы между частями, не устраивают сами части, не устраивает сам принцип членения. Люди, которые ведут расследование ошибки, пы-

таются построить собственную модель происходившего, дать свою версию.

Известно, сколь велико **влияние ошибок на формирование профессионального опыта** специалистов и совершенствование действий. Путь к профессиональному мастерству лежит через преодоление ошибок в системах. Опыт не может возникнуть из одних только знаний правил. Попытки исполнения действия в соответствии с правилами обязательно влекут за собой ошибки. Такого рода ошибки обязательны – они источник опыта любого человека.

После достижения определенного профессионального уровня профессиональное развитие прекращается – специалист достиг своего потолка – **уровня некомпетентности** [43]. Если сложность задач выше этого потолка – специалист не может действовать так же успешно, как при решении более простых задач. Субъект больше не в состоянии работать на достигнутом уровне сложности. Крайние, экстремальные ситуации чрезвычайно редки, поэтому не может быть речи о накоплении опыта действий в экстремальных ситуациях. По отношению к ним всякий выступает как новичок.

Здесь можно поставить вопрос о другой составляющей исполнения: **переживании вины за совершенную ошибку**. Оно становится базой для накопления опыта в течение длительного периода времени, а не просто после однократного исполнения. Деятельность, профессия должны представлять ценность для индивида. Окружающие люди должны представлять ценность для человека, тогда он прислушается к их мнению по поводу полученного результата. В противном случае индивид будет продолжать делать ошибки, несмотря на замечания и критику окружающих людей. В нормальных условиях в позитивном случае именно **социальный контроль** является основой формирования профессионального опыта.

Влияние ошибки на формирование профессионального опыта и совершенствование действий: через преодоление ошибок к мастерству. Высший профессионализм предполагает высокую требовательность специалиста к себе, критическое отношение к совершенным ошибкам, усиление контроля за действием после допущенной ошибки. Опыт – это не только высокое качество действий, но и пристальный контроль за исполнением. **Присутствие членов коллектива** делает контроль еще более пристальным: контроль за исполнением, коррекция и реакция, разбор ошибок. От группы исходит наказание за совершенную

ошибку, оценки и мнения окружающих имеют важнейшее значение для профессионального становления специалиста. Навыки выполнения профессиональной деятельности формируются под действием групповых норм.

Ошибка должна быть принята человеком, он должен признать ошибку и принять вину на себя. Людям свойственно не признаваться в ошибках, либо эти признания бывают ситуативными, поверхностными. Дело в том, что понятие *ошибка* имеет особый психологический характер, *оно чуждо субъекту*. Но поскольку само понятие ошибки является неоднозначным; поскольку велика роль случайности, стихии; поскольку ошибка предполагает сопоставление результата и процесса, который привел к ошибочному результату; поскольку ошибочный результат был достигнут вопреки стремлениям субъекта, вопреки его общим усилиям, которые были направлены на положительный результат – субъект сопротивляется.

В крупных комплексах программ статистика и распределение типов ошибок и выполняемых изменений для коллективов разных специалистов нивелируются и проявляются достаточно общие закономерности, которые могут использоваться *как ориентиры* при их выявлении. Каждому типу необходимых корректировок соответствует более или менее определенная категория специалистов, являющихся источником дефектов данного типа (см. таблицу 8.1). Такую корреляцию целесообразно учитывать как *общую качественную тенденцию* при анализе и поиске их причин. Этому могут помочь оценки типовых дефектов и корректировок путем их накопления и обобщения по опыту разработки определенных классов комплексов программ в конкретных предприятиях.

Важной особенностью дефектов и ошибок в комплексах программ обычно является *отсутствие полностью определенной программы-эталона*, которой должны соответствовать текст и результаты функционирования программного продукта. Поэтому установить наличие и локализовать дефект непосредственным сравнением с программой без ошибок в большинстве случаев невозможно. При отладке и тестировании специалистами обычно сначала обнаруживаются *вторичные* ошибки и/или *риски*, т.е. последствия и результаты проявления некоторых внутренних дефектов или некорректностей программ. Эти внутренние *дефекты* следует квалифицировать как *первичные* ошибки или причины обнаруженных аномалий результатов.

Последующая локализация и корректировка таких первичных ошибок должна приводить к устранению ошибок, первоначально обнаруживаемых в результатах функционирования программ.

Потери эффективности программ за счет их неполной корректности в первом приближении можно считать прямо пропорциональными (с коэффициентом) проявлениям вторичных ошибок в выходных результатах. Типичным является случай, когда одинаковые по величине и виду вторичные ошибки в различных результирующих данных существенно различаются по своему воздействию на общую эффективность применения программного продукта. Это влияние вторичных ошибок, в лучшем случае, можно оценить методами экспертного анализа при условии предварительной, четкой классификации видов возможных первичных ошибок в программах и выходных величинах. Таким образом, оценка последствий, отражающихся на вторичных ошибках и функционировании программ, может, в принципе, производиться *по значениям ущерба – риска в следствие не устраниенных их причин – первичных ошибок в программе*. Вторичные ошибки являются определяющими для эффективности функционирования программ, так как не каждая первичная ошибка вносит заметный вклад в выходные результаты. Вследствие этого ряд первичных ошибок может оставаться не обнаруженным и, по существу, не влияет на функциональные характеристики программного продукта (см. рис. 8.2).

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

всех вторичных проявлений, и внесения первичных ошибок при корректировках программ определяют скорость достижения заданной характеристики качества программного продукта.

На практике в процессе ЖЦ комплекса программ исходные требования поэтапно уточняются, модифицируются, расширяются и детализируются по согласованию между заказчиком и разработчиком. Базой таких уточнений являются *неформализованные представления, знания и квалификация специалистов-заказчиков, пользователей и разработчиков*, а также результаты промежуточных этапов проектирования. Однако установить не корректность таких эталонов еще труднее, чем обнаружить дефекты в готовых программах, так как принципиально отсутствуют формализованные данные, которые можно использовать как исходные. В процессе декомпозиции и верификации исходной спецификации требований на комплекс программ, возможно появление ошибок в спецификациях на группы программ и на отдельные модули. Это способствует расширению спектра возможных дефектов и вызывает необходимость создания гаммы методов и средств тестирования для выявления некорректностей в спецификациях на компоненты разных уровней.

Оценки качества программных продуктов при наличии дефектов и ошибок могут проводиться с двух позиций: с *позиции положительной* эффективности и непосредственной адекватности их характеристик назначению, целям создания, требованиям заказчика, а также с *негативной позиции* возможного при этом ущерба – риска от использования программного продукта или системы с ошибками. Риски характеризуют возможные негативные последствия дефектов или ущерб пользователям при применении и функционировании программного продукта или системы, и задача специалистов – разработчиков сводится к сокращению дефектов и ликвидации рисков [55, 56, 61].

К *понятию рисков* относятся негативные события и их величины, отражающие потери, убытки или ущерб от процессов или продуктов, *вызванные дефектами* при проектировании требований, недостатками обоснования проектов, а также при последующих этапах разработки, реализации и всего жизненного цикла комплексов программ. Риски проявляются как *негативные последствия дефектов функционирования и применения продуктов*, которые способны вызвать ущерб системе, внешней среде или пользователю. Они явля-

ются результатами отклонения характеристик объектов или процессов от заданных требований заказчика, согласованных с разработчиками.

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

Отчеты специалистов о дефектах и ошибках – основные инструменты для обнаружения дефектов в компонентах, продуктах и документах любых видов (см. лекцию 7). Воспроизведение дефекта обостряет его понимание и позволяет фиксировать последовательность шагов, которые заново приведут к проявлению ошибки. Лучшие отчеты об ошибках не должны являться ни загадочными комментариями, ни бесконечными и бесцельными документами. Следует пользоваться только теми словами, которые нужны, чтобы описывать последовательность действий и давать информацию, действительно имеющую отношение к ошибке, по которой готовится отчет для ее локализации. Для организации эффективного производства конкретного комплекса программ целесообразно унифицировать регистрацию и оценку опасности (приоритетов для устранения) дефектов и ошибок на основе шаблона типовых **отчетов специалистов о выявленных дефектах, ошибках и предложениях** по корректировке комплекса программ.

Изучение свойств и освоение характеристик типовых ошибок и дефектов непосредственно связано с достижаемой корректностью, безопасностью и надежностью функционирования программных продуктов и **помогает повышать профессиональную квалификацию специалистов, чтобы уметь:**

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

Статистика ошибок и дефектов в комплексах программ и их характеристики в конкретных типах проектов могут служить *ориентирами* для квалифицированных руководителей и специалистов при распределении ограниченных ресурсов в жизненном цикле комплексов программ и *предохранять от излишнего оптимизма* при оценке достигнутого качества программных продуктов. В крупных комплексах программ статистика и распределение типов выполняемых изменений для коллективов разных специалистов нивелируются и проявляются достаточно общие закономерности, которые целесообразно использовать как ориентиры при их систематизации, выявлении и устранении. Этому могут помочь оценки типовых дефектов, модификаций и корректировок путем их накопления и обобщения по опыту создания определенных классов и уровня сложности программных продуктов в конкретных предприятиях.

Сложность различных объектов и процессов значительно конкретизируется и становится измеримой, когда устанавливается *связь этого понятия с конкретными ресурсами*, необходимыми для решения соответствующей задачи и возможными размерами объектов, в которых возможно проявление дефектов. При производстве комплекса программ сложность проявляется на множестве этапов. Сама цель и задачи продукта могут быть настолько сложными, что делают трудным понимание и осознание руководителями и специалистами всех влияющих факторов и задач при проектировании. Это делает *сложной выработку четких требований, программ, документов и тестов*, а также правильного выбора и надежной реали-

зации решения с **допустимым минимумом дефектов и ошибок**. Средства, используемые для реализации решения, могут быть сложными, и это усложняет использование этих средств правильно и эффективно. Также может быть сложным общение между членами групп и всего коллектива специалистов, правильное и эффективное распространение информации, координация усилий и взаимодействия при создании крупного программного продукта, что может влиять на появление дефектов и ошибок.

Проблема сложна, когда существует больше факторов и переменных величин, чем может обработать отдельный специалист. Отбрасывание требований, намеренное или ненамеренное, – это не самый лучший путь уменьшить сложность, поскольку пропущенные требования вернутся обратно позже. Если требование было пропущено ненамеренно, это означает, что о нем просто забыли, и оно, в конечном счете, станет дефектом или ошибкой. К группе факторов, влияющих на **сложность выявления дефектов и ошибок** комплексов программ, относятся (см. лекцию 4):

- величина – размер программы, выраженная числом строк текста, функциональных точек или количеством программных модулей в комплексе;
- количество обрабатываемых переменных или размер и структура памяти, используемой для размещения базы данных;
- трудоемкость разработки комплекса программ;
- длительность разработки и реализации программного продукта;
- число специалистов, участвующих в жизненном цикле комплекса программ.

Некоторые из перечисленных факторов коррелированы между собой, причем, определяющими часто являются размер программы и объем базы данных. Остальные характеристики можно рассматривать как вторичные, однако они могут представлять самостоятельный интерес при анализе сложности и прогнозировании вероятного числа дефектов и ошибок в программе.

Масштаб – размер комплексов программ и их изменяемой части наиболее сильно влияет на количество дефектов и ошибок, а также на требования к **качеству** программного продукта. По мере увеличения размера и повышения требований к качеству, затраты на обнаружение и устранение ошибок увеличиваются все более высоки-

ми темпами. Одновременно расширяется диапазон неопределенности достигаемого качества. В зоне высокого качества программ возрастают сложности измерения этих характеристик, что может приводить к необходимости изменения затрат в несколько раз в зависимости от применяемых методов и результатов оценки качества.

Сложность обнаружения и устранения определенных типов дефектов и ошибок в комплексах программ обычно приходится связывать с *профессиональной квалификацией специалистов* выделяемых для соответствующих работ и компонентов. Показатели сложности при анализе возможных дефектов и ошибок можно разделить на *две большие группы*:

- *сложность появления ошибок при создании* компонентов и комплекса программ – статическая сложность, когда реализуются его требуемые функции, в программы вносятся основные дефекты и ошибки;
- *сложность проявления ошибок функционирования* программ и получения результатов – динамическая сложность, когда проявляются дефекты и ошибки, отражающиеся на функциональном назначении, рисках и качестве применения версии программного продукта.

Сложность технологического процесса производства программного продукта часто проявляется, когда группа специалистов, работает с одним и тем же набором модулей или компонентов. Это означает, что нужно осуществлять поддержку множества версий программ, которые выполняют подобные, но не точно повторяющиеся функции. Когда исходный модуль меняется в одной ветви, шанс нарушить работу программного компонента в другой ветви повышается, поскольку обычно не рассматриваются эти изменения в отношении к другой ветви. Сложность такой технологии повышается экспоненциально, когда группа людей работает над компонентами одного программного комплекса в одно и то же время.

Требования к профессиональной квалификации специалистов для учета причин и типов дефектов и ошибок в крупных комплексах программ

Одним из наиболее критических (и часто упускаемых) аспектов производства программных продуктов является *понимание специа-*

листами того, **почему возникают ошибки и откуда они берутся** (см. таблицу 8.1). Если можно определить, какой тип ошибки присутствует в системе, и установить связь с некоторыми действиями в ходе процесса разработки, можно улучшить этот процесс, предотвратить или удалить ошибки. Первичные ошибки в компонентах и комплексах программ можно анализировать с разной степенью детализации и в зависимости от различных факторов. Практический опыт показал [5, 26, 49], что **наиболее существенными факторами**, влияющими на характеристики обнаруживаемых ошибок являются:

- профессиональная квалификация специалистов и их роль в производстве программного продукта;

- методология, технология и уровень автоматизации системного и структурного проектирования комплексов программ, а также непосредственного программирования компонентов;

длительность с начала процесса проектирования и текущий этап производства и модификации комплекса программ;

класс комплекса программ, масштаб (размер) и типы основных компонентов, в которых могут содержаться дефекты и ошибки;

- методы, виды и уровень автоматизации тестирования, их адекватность требованиям и характеристикам компонентов и потенциально возможным в программах дефектам и ошибкам;

- виды и достоверность эталонов – требований, которые используются для обнаружения всех типов дефектов и ошибок.

Совокупность причин и типов дефектов и ошибок в сложных комплексах программ можно упорядочить и представить с учетом их зависимости от потенциальной опасности и возможной величины корректировок их последствий – рис. 8.3.

В верхней части перечня расположены дефекты и ошибки, последствия которых обычно требуют наибольших затрат ресурсов для реализации корректировок, они постепенно сокращаются при снижении по перечню.

Одной из основных причин ошибок в сложных комплексах программ являются **организационные дефекты при определении и использовании требований к программному продукту**, которые отличаются от остальных типов и условно могут быть выделены как самостоятельные (см. рис. 8.3). Ошибки и дефекты этого типа появляются из-за **недостаточного понимания заказчиком и коллективом специалистов целей и функций комплекса программ**, а также вследст-

вие отсутствия четкой его организации и поэтапного контроля требований качества продуктов.

Квалифицированные специалисты, следующие причины и типы дефектов и ошибок в сложных комплексах программ, должны знать, уметь выделять и учитывать:

- профессиональную квалификацию специалистов и их роль в производстве программного продукта;
- организационные дефекты при определении и использовании требований к программному продукту;
- ошибки оценки исходных характеристик системы и внешней среды для программного продукта;
- системные дефекты и ошибки адекватности и полноты требований к программному продукту;
- системные ошибки корректности формулирования и понимания требований к программному продукту;
- ошибки проектирования и разработки архитектуры программного комплекса;
- ошибки реализации требований к программному продукту;
- алгоритмические ошибки не корректной постановки задач, не полного учета условий среды компонентов и комплекса программ;
- ошибки реализации спецификаций требований к программным компонентам;
- программные ошибки модулей, компонентов и комплекса программ;
- ошибки документации компонентов и комплекса программ;
- технологические ошибки производства компонентов и программного продукта.

Рис. 8.3

Это порождается пренебрежением руководителей к организации всего технологического процесса формализации требований сложных программных продуктов и приводит к серьезной недооценке их возможных дефектов, а также трудоемкости и сложности их выявления. При отсутствии планомерной и методичной разработки и тестирования требований может оставаться не выявленным значительное количество ошибок, и, прежде всего, дефекты требований к взаимодействию отдельных функциональных компонентов между собой и с внешней средой. Для сокращения этого типа достаточно массовых ошибок, активную роль должны играть **лидеры – руководители и**

аналитики-системотехники, способные вести контроль и конфигурационное управление требованиями, изменениями и развитием версий и функциональных компонентов комплексов программ.

Ошибки оценки характеристик системы и внешней среды, принятых специалистами в процессе проектирования и производства комплекса программ за исходные, могут быть результатом аналитических расчетов, моделирования или исследования аналогичных систем. В ряде случаев может отсутствовать полная адекватность предполагаемых и реальных характеристик, что является **причиной сложных и трудно обнаруживаемых системных ошибок и дефектов развития проекта**. Ситуация с такими ошибками дополнитель но усложняется тем, что эксперименты по проверке взаимодействия программного продукта с реальной внешней средой во всей области изменения характеристик зачастую сложны и дороги, а в отдельных случаях при создании опасных ситуаций, недопустимы. В этих случаях приходится использовать моделирование и имитацию внешней среды с заведомым упрощением ее отдельных элементов и характеристик, хотя степень упрощения не всегда можно оценить с необходимой точностью.

Системные ошибки и дефекты адекватности требований к программному продукту определяются, прежде всего, неполной информацией о реальных процессах, происходящих в источниках и потребителях информации [23, 25, 51]. Кроме того, эти процессы зачастую зависят от самих алгоритмов и поэтому не могут быть достаточно определены и описаны заранее без исследования функционирования комплекса программ во взаимодействии с внешней средой. На начальных этапах производства не всегда удается точно и полно сформулировать целевую задачу всей системы, а также целевые задачи основных функциональных групп программ, и эти задачи уточняются в процессе проектирования. Характеристики внешних объектов, принятые в качестве исходных данных в процессе разработки алгоритмов могут являться результатом аналитических расчетов, моделирования или исследования аналогичных систем. Во многих случаях отсутствует полная адекватность условий получения предполагаемых и реальных характеристик внешней среды, что может являться причиной сложных и трудно обнаруживаемых ошибок. Это усугубляется тем, что часто невозможно заранее предусмотреть все разнообразие возможных внешних условий и реальных вариантов сценариев функ-

ционирования и применения версий программных продуктов. При автономной и в начале комплексной отладки версий программного продукта **относительная доля** системных ошибок может быть невелика (около 10%), но она существенно возрастает (до 35 – 40%) на завершающих этапах комплексной отладки новых версий продукта. В процессе сопровождения системные дефекты и ошибки обычно являются преобладающими (около 60 – 80% от всех ошибок). Следует также отметить большое количество команд, корректируемых при исправлении каждой такой ошибки (около 20 – 50 строк текста программ на одну ошибку) [26].

Системные ошибки корректности формулирования и понимания требований к программному продукту считаются наиболее критичным для общего успеха системы в целом или его версий. Ошибки требований являются наиболее трудными для обнаружения и наиболее сложными для исправления, которое может быть в 15 – 70 раз дороже, чем ошибок программирования [4, 42, 49]. Требование может быть пропущено в спецификации к системе и комплексу программ. Это ведет к неудовлетворенности пользователя, и комплекс программ считается заказчиком и пользователем дефектным. Пропуск части требований – наиболее обычная проблема среди ошибок требований. Ошибки требований могут также представлять собой конфликтующие требования в спецификации. Может проявляться неопределенность требований – такой способ формулирования требования, что даже если и не конфликтует с другим требованием, оно выражено недостаточно ясно, чтобы привести к единственному, конструктивному решению при производстве комплекса программ. Многочисленные исследования показали, что **ошибки корректности требований труднее всего обнаруживать и дороже всего исправлять**.

Ошибки проектирования и разработки архитектуры программного комплекса определяются процессами перевода неопределенных и общих положений, сделанных на этапе первичных спецификаций требований, в более точные технические описания сценариев того, как программный продукт и система должны работать [11, 49, 69]. Ошибки структуры легче обнаружить инспекциями, чем ошибки требований, но они в конечном итоге могут оказаться при корректировках весьма дорогостоящими. Главная причина того, что ошибки структуры дорого исправлять, состоит в том, что они могут

влиять на систему в целом. Если разработчик структуры комплекса, либо неверно прочитает требования, либо не поймет содержание требования также как заказчик или конечный пользователь, то появится дефект или ошибка проектирования структуры компонента или комплекса программ.

Ошибки реализации требований к программному продукту – программная ошибка или программный дефект, возможно, действительно ошибка требований или структуры, программная ошибка обычно дефект реализации. Ошибки реализации наиболее обычны и, в общем, наиболее легки для исправления в системе, что не делает проблему легче для программиста. Например, трудно столкнуться с проблемами временных соотношений в программном обеспечении аппаратных контроллеров, работая на обычном персональном компьютере. Однако такие ошибки программных контроллеров обычны в мире встроенных программ.

Алгоритмические ошибки программ трудно поддаются обнаружению методами статического автоматического контроля. Трудность их обнаружения и локализация определяется, прежде всего, отсутствием для многих логических программ строго формализованной постановки задачи, полной и точной спецификации, которую можно использовать в качестве эталона для сравнения результатов функционирования программ. К алгоритмическим ошибкам следует отнести, прежде всего, ошибки, обусловленные некорректной постановкой требований к функциональным задачам, когда в спецификациях не полностью оговорены все условия, необходимые для получения правильного результата. Ошибки, обусловленные не полным учетом всех условий решения задач, являются наиболее частыми в этой группе и составляют до 50 – 70% всех алгоритмических ошибок.

К алгоритмическим ошибкам следует отнести также ошибки интерфейса модулей и функциональных групп программ, когда информация, необходимая для функционирования некоторой части программы, оказывается не полностью подготовленной программами, предшествующими по времени включения, или неправильно передается информация и управление между взаимодействующими модулями. Этот вид ошибок составляет около 10% от общего количества, и можно квалифицировать как ошибки некорректной постановки задач. Алгоритмические ошибки проявляются в неполном учете диапазонов изменения переменных, в неправильной оценке точности используе-

мых и получаемых величин, в неправильном учете корреляции между различными переменными, в неадекватном представлении формализованных условий решения задачи в виде частных спецификаций или блок-схем, подлежащих программированию. Эти обстоятельства являются причиной того, что для исправления каждой алгоритмической ошибки приходится изменять в среднем около 20 строк текста, т.е. существенно больше, чем при программных ошибках.

Особую, весьма существенную, часть алгоритмических ошибок в системах реального времени, составляют просчеты в использовании доступных ресурсов вычислительной системы. Наиболее крупные просчеты обычно допускаются при оценке времени реализации различных групп программ реального времени, и при распределении производительности ЭВМ. Алгоритмические ошибки этого типа обусловлены технической сложностью расчета времени реализации программ и сравнительно невысокой достоверностью, определения вероятности различных маршрутов обработки информации.

Ошибки реализации спецификаций требований к компонентам – это программные дефекты, возможно, ошибки требований, структуры или программные ошибки компонентов. Ошибки реализации наиболее обычны и особенно разнообразны и, в общем, наиболее легки для исправления в системе, что не делает проблему легче для программистов. В отличие от ошибок требований и структурных ошибок, которые обычно специфичны для компонентов, определенные программисты часто совершают при кодировании программ разнообразные, но одни и те же типы ошибок, которые можно свести к следующим категориям.

Первую категорию составляют дефекты, которые приводят к отображению для пользователя сообщений об ошибках при точном следовании порядку выполнения требуемых функций. Хотя эти сообщения могут быть вполне законны, пользователи могут посчитать это ошибкой, поскольку они делали все правильно и, тем не менее, получили сообщение об ошибке. Вторая категория модификаций может содержать ошибки, связанные с дефектами в графическом интерфейсе пользователя. Третья категория может содержать пропущенные на стадии реализации функции, что всегда считается ошибкой, возможно, с большим риском. Многие тестировщики и пользователи бета-версий сообщают об ошибках, которые на самом деле являются желательными улучшениями.

Программные ошибки компонентов и комплекса программ по количеству и типам в первую очередь определяются степенью автоматизации программирования и полнотой статического контроля текстов программ. Количество программных ошибок зависит от квалификаций программистов, от общего размера комплекса программ, от глубины информационного взаимодействия модулей и от ряда других факторов. При производстве программные ошибки можно классифицировать по видам используемых операций на следующие **крупные группы**: ошибки типов операций; ошибки переменных; ошибки управления и циклов. На начальных этапах разработки и автономной отладки модулей, программные ошибки составляют около одной трети всех ошибок. Каждая программная ошибка влечет за собой необходимость изменения до 10 строк текста, что существенно меньше, чем при алгоритмических и системных ошибках.

Ошибки в документации состоят в том, что система делает что-то одним образом, а документация отражает сценарий, что она должна работать иначе. Во многих случаях права должна быть документация, поскольку она написана на основе оригинальной спецификации требований системы. Иногда документация пишется и включает допущения и комментарии о том, как, по мнению авторов документации, система должна работать. Ошибки документации можно разделить на **три категории – неясность, неполнота и неточность**. Несколько – это когда пользователю не дается достаточно информации, чтобы определить, как сделать процедуру должным образом. Неполная документация оставляет пользователя без информации о том, как правильно реализовать и завершить задачу. Такие ошибки ведут к тому, что пользователь неудовлетворен версией программного продукта, даже если программа в действительности может сделать все, что хочет пользователь. Неточная документация – это худший вид ошибок документации. Такие ошибки часто возникают, когда в систему вносятся изменения и о них не сообщается специалисту, пишущему документацию.

В то же время в жизненном цикле крупных комплексов программ **невозможно в начале проекта предусмотреть** все требования к функциям, характеристикам и качеству абсолютно точно, **без дефектов и ошибок**. Кроме того, крупные программные продукты обычно имеют длительный жизненный цикл с рядом версий, в течение которого они совершенствуются, расширяются, и повышается их

качество. Поэтому процессы документирования изменений и реализации *требований, тестов и эксплуатационных документов* должны организовываться на *основе методологии регулируемых итераций множества версий*.

Технологические ошибки отличия документации и программного продукта в памяти ЭВМ составляют иногда до 10% от общего числа ошибок, обнаруживаемых при тестировании и эксплуатации. Непосредственное участие человека в подготовке данных для ввода в ЭВМ и при анализе результатов функционирования программ по данным на дисплеях определяет в значительной степени их уровень достоверности и не позволяет полностью пренебрегать этим типом ошибок в программах. При ручной подготовке текстов машинных носителей при однократном фиксировании исходные данные имеют вероятность искажения около $10^{-3} - 10^{-4}$ на символ. Дублированной подготовкой и логическим контролем вероятность технологической ошибки может быть снижена до уровня $10^{-5} - 10^{-7}$ на символ.

Приведенные характеристики типов и причин дефектов и ошибок, и их свойства могут служить *ориентирами при прогнозировании возможного наличия не выявленных ошибок* в различных сложных программных продуктах. Следующим логическим шагом процесса их оценивания может быть усреднение для большого числа проектов фактических данных о количестве ошибок на конкретном предприятии, приходящихся на тысячу строк кода, которые обнаружены в различных продуктах. Тогда в следующем проекте будет иметься возможность использовать эти данные в качестве меры для прогнозирования количества ошибок, обнаружение которых следует ожидать при выполнении проекта с таким же уровнем качества, или с целью повышения производительности труда. Подобные оценки гарантируют *от избыточного оптимизма* при определении сроков и при разработке графиков разработки, сопровождения и реализации модификаций программ с заданным качеством. Непредсказуемость конкретных ошибок в программах приводит к целесообразности последовательного, методичного фиксирования и анализа возможности проявления любого типа дефектов и необходимости их исключения на наиболее ранних этапах ЖЦ комплексов программ при минимальных затратах.

Лекция 9

ТРЕБОВАНИЯ К ПРОФЕССИОНАЛЬНОЙ КВАЛИФИКАЦИИ СПЕЦИАЛИСТОВ, ТЕСТИРУЮЩИХ КОМПОНЕНТЫ И КРУПНЫЕ КОМПЛЕКСЫ ПРОГРАММ

Требования к профессиональной квалификации специалистов для успешного тестирования компонентов и комплексов программ

Специалисты, контролирующие и управляющие обеспечением качества крупных программных продуктов, должны владеть стандартами и методиками предприятия, поддерживающими регистрацию, контроль, документирование и воздействия на показатели качества на последовательных **этапах тестирования комплексов программ**. Они должны обеспечивать эксплуатацию системы качества проекта, выявление всех отклонений от заданных показателей качества объектов и процессов, а также от предписанной технологии на этапах сопровождения и управления конфигурацией (см. лекцию 7). Эти же специалисты должны анализировать возможные последствия выявленных отклонений от требований технического задания или спецификации на изменения. В результате должны приниматься меры либо по устранению отклонений, либо по **корректировке требований**, если устранение отклонений требует больших ресурсов. Специалист, ответственный за выполнение конкретных задач тестирования программного продукта **с высоким риском** [4, 28, 60] должен быть **аттестован**, если это необходимо, на основе соответствующего образования, подготовки и/или опыта работы. Может также стать необходимым включение в подготовку специалистов ознакомление со специфической (проблемно-ориентированной) областью, в которой будет использован данный программный продукт, и повышение профессиональной квалификации в этой области.

Основной целью квалифицированных специалистов при тестировании является обеспечение возможности программного

продукта функционировать надлежащим образом при любых условиях и данных и **соответствовать установленным и утвержденным заказчиком требованиям**. Тестирование предназначено для обнаружения дефектов и ошибок и последующего их устранения. Продукт должен удовлетворять ожиданиям заказчика и конечных пользователей, при этом должно быть выявлено и устранено как можно больше его дефектов. Кроме того, тестирование должно обеспечивать характеристики качества, достаточные для принятия решения о готовности программного продукта для передачи заказчику и применения по назначению (рис. 9.1).

Квалифицированные специалисты по тестированию компонентов и комплексов программ должны знать и уметь:

- участвовать и принимать решения при разработке требований к компонентам и комплексам программ для их эффективного тестирования;
- выделять и учитывать личные свойства и психологические характеристики специалистов при организации групп для эффективного тестирования компонентов и комплекса программ;
- осуществлять выбор, организацию и упорядочение рентабельных стратегий, процессов и методов тестирования комплекса программ;
- анализировать и выбирать приоритеты сценариев и значений тестов для эффективного тестирования комплекса программ;
- осуществлять верификацию и контроль наборов тестов на соответствие требованиям к компонентам и комплексу программ;
- применять критерии для оценивания достигнутого качества результатов тестирования компонентов и комплекса программ и принимать ответственные решения для его завершения;
- использовать статистические данные результатов тестирования для оценивания квалификации разработчиков компонентов и комплексов программ.

Рис. 9.1

Существенным является **участие профессиональных специалистов по тестированию в разработке требований к программному продукту** [4]. На этапе определения требований тестирование должно способствовать созданию ясных и непротиворечивых требований. Участие тестировщиков на этом этапе нужно еще и для того, чтобы обеспечить формулировку системных и всего комплекса требо-

ваний в пригодных для тестирования терминах. При определенном исходном состоянии системы и множестве входных параметров тестировщики должны иметь возможность предсказать состав и содержание выходных эталонных данных программного продукта и системы. **Тестировщики требований контракта на программный продукт** должны обеспечивать проверку реализации требований функциональных спецификаций, пользовательских интерфейсов, разрабатывать стратегию, выполнять и документировать тестирование для каждого компонента и версии программного продукта, должны быть административно независимыми от программистов и спецификаторов требований.

В процессе тестирования квалифицированные специалисты должны уметь проверить, что комплекс программ работает в соответствии со спецификацией требований и **удовлетворяет следующим общим критериям** [13]:

- при допустимых входных данных комплекс программ вырабатывает верный результат, соответствующий требованиям;
- при недопустимых входных данных комплекс программ отвергает входные данные и выдает соответствующее диагностическое сообщение;
- независимо от допустимости входных данных, программы при применении не останавливаются (не зависают) и не завершаются аварийно.

Личные качества и профессиональная квалификация тестировщиков крупных комплексов программ и отношения, установившиеся в коллективе, представляют собой значительный фактор для повышения эффективности разработки и безопасности программных продуктов. Организационная структура групп тестирования и распределение ответственности между специалистами за определенные уровни тестов обычно включаются в состав плана работ всего проекта. Необходимо определять и фиксировать в плане тестирования **распределение основных ролей и ответственности специалистов** между участниками процесса тестирования [4, 42, 60]. Ни один специалист не может быть носителем всех необходимых высоких качеств и квалификации, однако **профессиональная группа тестирования**, будучи единым коллективом специалистов, **должна владеть максимально возможным количеством следующих технологических и психологических качеств** [17]:

- обладание аналитическим складом ума;
- обладать незаурядными умственными способностями, легко и быстро осваивать новые технологии тестирования;
- внимательность к деталям и организованностью;
- творческим подходом и способностью к прогнозированию результатов своей деятельности;
- наличием опыта разработки эффективной стратегии тестирования;
- умением проводить работы по тестированию в необходимом и достаточном объеме;
- уметь описывать последовательность событий и конфигурацию системы, которые приводят к возникновению дефекта, это включает способность четко документировать процедуры и результаты тестирования требований, умение передать такую информацию разработчикам, другим тестировщикам и руководству;
- уметь критиковать и корректно воспринимать критику, так объяснять разработчикам компонентов суть дефектов, чтобы с его слов их можно было устраниТЬ;
- уметь противостоять внешнему давлению руководства, так как тестирование требований является завершающей стадией любого процесса разработки и, как правило, протекает в стрессовых обстоятельствах недостатка ресурсов и времени;
- быть терпеливым и готовым выполнять тесты столько раз, сколько нужно для того, чтобы выявить и локализовать дефект, после чего повторно выполнить регрессионные тесты, чтобы убедиться в корректном его устранении;
- видеть общую панораму развития проекта и уметь при необходимости сосредоточиваться на приоритетных деталях, иметь широкий и динамичный кругозор;
- быть экспертом в нескольких профессиональных областях – группе тестирования могут потребоваться специалисты: по базам данных; по управлению конфигурацией; по коммуникациям; по сетевым технологиям; по тестированию интерфейсов; по инструментальным средствам; по сценариям автоматизации тестирования; по моделям внешней среды.

Тестировщики должны уметь обнаруживать дефекты, быть твердыми и гибкими *при обсуждении спорных вопросов* с разработчиками компонентов и программистами. С учетом сложности тести-

рования в клиент-серверной или многопользовательской среде, тестирующие должны обладать большой совокупностью профессиональных знаний и навыков. Они должны иметь опыт работы с различными платформами, операционными системами, приложениями поддержки, интерфейсами других продуктов и систем, базами данных и языками разработки приложений.

Для сложных комплексов программ специалистам – тестирующим следует выделять и фиксировать пригодные для реализации тестирования *системные требования и сценарии использования функций* системы и программного продукта. При анализе требований к тестам квалифицированным специалистам необходимо *уметь решать* следующие основные *проблемы*:

- **что** нужно сделать при определении требований к тестам для проверки корректности конкретной версии программного продукта или его функциональных компонентов;
- **как** преобразовать архитектуру, системные требования к функциям и характеристикам, в сценарии и результаты использования программного продукта, а также в конкретные требования к совокупности тестов;
- **как** провести анализ и формирование документации на комплекс программ для формулировки и документирования требований к тестам и результатам тестирования.

Тестовые сценарии могут разрабатываться как для проверки *функциональных* требований и характеристик, так и для оценки *нефункциональных* (архитектурных) требований. При этом существуют такие тесты, когда количественные параметры и результаты тестов могут лишь качественно удовлетворять цели тестирования, например, простота использования, в большинстве случаев, не может быть явно описана количественными характеристиками. *Специалисты* должна освоить и применять следующие, *способы выбора и организации методов и процессов тестирования* [13]:

- тестирование, базирующееся *на интуиции и опыте специалистов*, наиболее широко используется и основывается на их знаниях, имевшихся ранее аналогий, оно может использоваться для идентификации тех дополнительных тестов, которые не охватываются более формализованными методами;
- тестирование *на основе потоков данных* – когда отслеживается полный жизненный цикл преобразования величин (переменных)

с момента их определения на всем протяжении использования, вплоть до уничтожения, может быть ориентировано на проверку задания начальных значений или всех переменных с точки зрения их применения;

- **функциональное тестирование** соответствия требованиям – проверка комплекса программ и/или системы, предъявляемым к ним требованиям, описанным на уровне спецификации функций и характеристик при применении;

- **системное тестирование** – охватывает целиком всю систему, обычно фокусируется на нефункциональных и/или динамических требованиях: безопасности, производительности, точности, надежности; тестируются интерфейсы к внешним приложениям, аппаратному обеспечению, операционной и внешней среде;

- тестирование **границых значений** строится с ориентацией на использование тех величин, которые определяют предельные характеристики тестируемых компонентов, расширением этого метода являются **тесты оценки живучести** системы, проводимые с величинами, выходящими за рамки специфицированных пределов значений в требованиях;

- **приемо-сдаточное тестирование** – испытания программного продукта, при котором проверяется поведение системы на предмет удовлетворения требований заказчика как с привлечением разработчиков системы, так и без них;

- **установочное тестирование** проводится с целью проверки процедур инсталляции программного продукта и/или системы в целевом окружении применения;

- **регрессионное тестирование** – повторное выборочное тестирование системы, комплекса или компонентов для проверки сделанных модификаций, оно не должно приводить к непредусмотренным эффектам, если система успешно проходила тесты до внесения модификаций, задача состоит в том, чтобы определить критерии допустимых масштабов изменений, после которых необходимы регрессионные тесты;

- **тестирование удобства эксплуатации и простоты применения** – проверка, насколько легко конечный пользователь может освоить программный продукт, включая не только функциональную составляющую, но и документацию;

- **тестирование, ориентированное на дефекты** – на конкретные, специфические категории ошибок, на обнаружение наиболее вероятных дефектов, предсказываемых, например, в результате анализа рисков.

Поскольку в крупных комплексах программ подвергнуть тестированию абсолютно все невозможно, **специалистам важно квалифицироваться на определенных приоритетах тестов и научиться выбирать, что и как обязательно** нужно тестировать. Если допустить полный перебор сценариев в тестировании (тестовое покрытие) будет избыточным и для отладки программного продукта потребуется значительное время, это может поставить под угрозу срок сдачи продукта заказчику. Если тестирование окажется недостаточным, то увеличится риск пропуска негативного эффекта при применении, устранение которого будет стоить дорого, особенно после сдачи программного продукта в эксплуатацию. Квалифицированный специалист должен уметь определить **нужный баланс** между этими крайностями на базе опыта и оценки успешности тестирования реализованных проектов (см. рис. 9.1).

Требования к тестам должны содержать подробный перечень того, что должно быть протестировано. При разработке требований к тестам специалисты по тестированию должны выполнить ряд шагов, чтобы получить представление о потребностях заказчика. Необходимо изучить системные требования к программному продукту, описание назначения и сценарии использования системы для того, чтобы понять ее цель. Кроме того, следует определить функции, наиболее **значимые для применения системы, и функции повышенного риска**.

Специалисты при подготовке квалифицированных тестов для комплексов программ должна освоить:

- разработку множеств тестируемых на основе их проектирования сверху вниз, используя цели тестов, анализ рисков, требования к функциям и характеристикам комплекса программ;
- контроль корректности множества тестов за счет проектирования снизу вверх на основе реального функционирования и использования системы, и/или сценариев взаимодействия с пользователями;
- использование пробного, исследовательского тестирования для заполнения возможных пробелов в покрытии тестами требований функций и характеристик комплекса программ;

- применение вариантов структурного анализа и инспекций для определения скрытых дефектов в тестовом покрытии архитектуры комплекса программ.

Для обеспечения высокого качества программного продукта параллельно с *верификацией требований к программному продукту* и разработкой корректировок специалистам следует разрабатывать и *верифицировать спецификации и сценарии тестов*, отражающие методы и конкретные процедуры проверки реализации этих требований. Тестовые спецификации могут использоваться для проверки согласованности, внутренней непротиворечивости и полноты реализации требований к программам. Для каждого требования к функциям комплекса программ, его архитектуре, функциональным компонентам специалистами по тестированию должна быть разработана спецификация требований к тестам, обеспечивающая проверку корректности, адекватности и возможности в последующем реализовать тестирование компонента на соответствие этому требованию. Такая взаимная проверка корректировок функций компонентов, отраженных в требованиях и в спецификациях тестов, обеспечивает повышение их качества, сокращение дефектов, ошибок, неоднозначностей и противоречий.

При разработке требований к тестам группа специалистов тестирования должна выполнить несколько предварительных шагов, в том числе получить четкое представление о *потребностях заказчика*. Необходимо изучить системные требования, сценарии использования системы и/или описание назначения системы для того, чтобы лучше понять цель ее разработки. Еще один шаг – определение *функций, наиболее значимых* для работы системы, и *функций повышенного риска*. Требования к тестам для комплексов программ можно также получать на основе представлений о логике архитектуры системы.

Квалифицированные специалисты тестирования функций и характеристик комплекса программ должны освоить и использовать стратегию следующих последовательных процессов [4, 13]:

- разработку автоматизированных и/или ручных тестов для покрытия всех функций, характеристик и рисков качества, которые определены как нуждающиеся в полном или сбалансированном *приоритетном тестировании*, с особым вниманием к факторам функ-

ционирования системы, наблюдаемым со стороны пользовательского интерфейса или доступного программного интерфейса системы;

- дополнять тестовые данные или условия тестовых сценариев для покрытия **критических вариантов** использования системы или сокращения известных типов дефектов, возможных в программном продукте;
- использовать исследовательское тестирование в областях, которые ранее не были покрыты тестами, и которые, с точки зрения интуиции или результатов тестирования, представляются областями **повышенного риска** наличия дефектов;
- насколько позволяют время и ресурсы или вызывает сомнение о возможных не выявленных рисках, использовать методы структурного покрытия для выявления не подвергавшихся тестированию областей и затем дополнять тесты для **покрытия критических пробелов** в тестовом покрытии;
- выполнять тестирование посредством **методов изучения реализации** программного продукта: тестирование потоков данных, тестирование путей и тестирование транзакций [3, 66];
- проводить **опытную эксплуатацию** и демонстрировать заказчику реальное применение – тестирование на основе того, что пользователи реально делают с комплексом программ и системой – бета-тестирование, изучение удобства использования и, в некоторых случаях, приемо-сдаточные испытания.

Важным аспектом **оценки профессиональной квалификации специалистов по тестированию** является умение принять ответственное решение о том, в каком объеме тестирование достаточно и **когда необходимо завершить процесс тестирования**. Тщательные измерения, такие как достигнутое покрытие тестами, или охват функциональности, очень полезны. Однако они не могут быть полным критерием достаточности тестирования. Принятие решения об окончании тестирования, включает умение оценки стоимости и рисков, связанных с потенциальными сбоями и нарушениями надежности и безопасности функционирования тестируемой программной системы. В то же время, стоимость самого тестирования также является одним из ограничений, на основе которых заказчик и руководители производства могут принимать решения о продолжении тех или иных связанных с проектом работ (в частности, тестирования) или об их прекращении.

Анализ и мониторинг характеристик, последствий и частоты выявленных дефектов при тестировании конкретного комплекса программ может служить *ориентиром для оценки индивидуальной профессиональной квалификации* и качества работы определенных специалистов, *разработчиков компонентов и комплекса программ*. Следствием такого анализа может быть выделение некоторых специалистов – разработчиков программ, отличающихся большим числом обнаруживаемых дефектов, для их замены или дополнительного обучения с целью сокращения числа первичных ошибок соответствующего типа. Накопление, классификация и обобщение характеристик дефектов определенных классов позволяет прогнозировать изменение выявления ошибок при тестировании, а также *необходимое распределение* состава и квалификации специалистов тестировщиков по компонентам и функциям комплекса программ.

Требования к профессиональной квалификации специалистов по организации, планированию, оценке и учету затрат на тестирование компонентов и крупных комплексов программ

Разработка тестов включает создание *квалифицированными специалистами – тестировщиками*, сопровождаемых, много-кратно применяемых, простых и надежных тестовых процедур, что может потребовать не меньше усилий, чем разработка программистами тестируемых текстов программ. Чтобы добиться максимального эффекта от автоматизации тестирования, тестировщики должны вести разработку тестов *параллельно* с созданием программистами текстов компонентов программного продукта. Схема структуры комплекса программ является графическим представлением основных работ, которые должны быть выполнены во время разработки тестов. Группе тестирования требуется проводить корректировку и уточнение структуры разработки тестов, чтобы отразить приоритеты конкретного программного продукта. Процессы разработки тестов для крупных программных продуктов, в зависимости от размера анализируемых компонентов, можно рассматривать на *двух уровнях* [4, 19, 34]:

- на уровне программных модулей и небольших их групп;

- на уровне целостных крупных комплексов программ и законченных функциональных компонентов, в которые интегрируются программные модули или группы программ.

Эти два уровня *принципиально различаются* применяемыми технологиями тестирования, которые, соответственно, относятся к методам программирования модулей и компонентов, и к *методам программной инженерии* крупных комплексов программ. Поскольку человеческий мозг способен оперировать ограниченным числом объектов одновременно, то чем сложнее фрагмент программы, тем более вероятно, что специалист допустит в нем ошибку. Сложность необходимых тестов для **контроля** функции, обычно такая же, как сложность *разработки* соответствующей функции. Покрытие их тестами для устранения дефектов и обеспечения качества – это последовательное покрытие модулей, компонентов или комплексов программ, которые могут определяться посредством анализа соответствия *пространства требований* к функциям и характеристикам программы и *пространства применяющихся тестов* для их контроля – рис. 9.2.

Измерение степени покрытия тестами текста модулей и небольших компонентов программ обычно производится, путем отслеживания управляющей логики исполнения программы в процессе тестирования. Инструменты по измерению тестового покрытия небольших компонентов позволяют оценивать процент проверенных строк кода, выполненных в ходе исполнения очередного набора тестов. Просматривая, что не было выполнено в ходе тестирования, руководители и тестировщики могут решать, нужно ли дополнять тесты для выявления и смягчения определенных рисков [3, 38, 61].

Анализ специалистами полноты покрытия тестами требований при исполнении комплекса программ должен определять, какие требования не были протестированы и какие части структуры программного продукта не были исполнены при тестировании. Тестирование структурного покрытия должно устанавливать, не пропущены ли элементы структуры программы, которые не проверены тестовыми процедурами, и покрыли ли тесты всю структуру комплекса программ. Тестовые варианты, основанные на требованиях, могут не полностью покрыть структуру программы. Анализ покрытия версии программного продукта тестовыми данными, основанными на требованиях, должен определить, насколько полно тестирование проверило реализацию всех требований, и показать потребность в дополнитель-

ных тестовых сценариях. При тестировании реализации требований к функциям и характеристикам программных компонентов полнота их покрытия тестами редко достигает 90% и хорошо, если составляет около 80% [4, 13, 17].

Квалифицированные специалисты для эффективного тестирования компонентов и крупных комплексов программ должны знать и уметь:

- организовать структуру и состав квалифицированного коллектива специалистов для тестирования компонентов и крупных комплексов программ;
- определять и сопоставлять пространства требований к функциям и характеристикам программ и пространства тестов для их контроля;
- выбирать и организовывать стратегию тестирования конкретного комплекса программ;
- разрабатывать планы тестирования комплекса программ в соответствии с установленной стратегией, доступными ресурсами и квалификацией коллектива специалистов;
- использовать методы и матрицы соответствия системных требований и тестовых сценариев, отслеживать реализацию плана тестирования требований к компонентам и комплексу программ;
- выполнять верификацию требований к тестам на соответствие требованиям, функциям и характеристикам компонентов и программного комплекса;
- выполнять корректировки дефектов и ошибок в результатах тестирования программ в соответствии с полномочиями специалистов;
- контролировать полноту выполненного плана тестирования и покрытия тестами требований к комплексу программ;
- планировать и осуществлять тестирование эксплуатационной и технологической документации комплекса программ;
- оценивать и контролировать ограничения использования всех видов ресурсов и квалифицированных специалистов для тестирования комплекса программ;
- оценивать рентабельные и допустимые сроки и ресурсы для завершения плана тестирования программного продукта;
- оформлять протоколы и акт завершения тестирования для предъявления программного продукта заказчику и пользователям.

Рис. 9.2

Стратегия тестирования комплекса программ – это совокупность выбранных методов и решений, вытекающих из целей и задач проекта и его тестирования, общие правила и принципы, способствующие достижению целей разработки программного продукта высокого качества. Руководитель группы специалистов, готовящий план тестирования, должен тщательно обдумывать идеи, выводы и решения таким образом, чтобы каждый специалист, использующий план, четко понимали, что от него требуется: стратегию тестирования, планирование и использование ресурсов, управление конфигурацией проекта и минимизацию рисков. При формировании стратегии тестирования специалистам целесообразно освоить и использовать **следующие рекомендации** [17].

Тестирование в первую очередь следует проводить для проверки реализации требований с наивысшим приоритетом, которые представляют для заказчика наибольшую важность, либо которые могут причинить заказчику наибольшие неприятности в случае проявления дефектов программного продукта. Если запланировано тестирование всех требований, и ресурсы это позволяют, естественно, проверить выполнение всех требований. В случае недостаточных ресурсов перед предъявлением продукта заказчику необходимо тщательно протестировать требования с наивысшими приоритетами. Целесообразно получить согласие заказчика на то, что требования, которые были проверены частично или не проверены вообще, не будут использоваться и поддерживаться вплоть до следующей версии продукта.

Тестирование сначала осуществлять для новых функциональных возможностей, которые изменились с целью исправления или совершенствования функций и характеристик. Если версия является очередным обновлением или эксплуатируемой версией, особое внимание следует уделить новым функциям и характеристикам. Любые изменения, внесенные в программы, могут исказить даже те части комплекса программ, которые непосредственно не затрагиваются изменениями. В этом случае как можно шире следует выполнять регрессионные тесты для всех функций и характеристик комплекса программ, какими бы ни были изменения.

Тестирование следует проводить тех компонентов и участков программ, в которых наиболее вероятно присутствие дефектов. Если обнаружен какой-то дефект, часто рядом может быть еще

несколько аналогичных дефектов. Если есть сведения от разработчиков, что часть компонентов породили проблемы во время модульного тестирования или проверки взаимодействия и функционирования компонентов, такие участки необходимо отметить, чтобы обратить на них внимание при комплексном тестировании или системных испытаниях.

Тестирование желательно начинать с функций и конфигураций, с которыми наиболее часто будет иметь дело конечный пользователь или система. Для этого в спецификации требований полезно включать методику оценки случаев использования некоторых функций и иметь доступ к функциональному разрезу конечного пользователя – математическому ожиданию использования каждой функции. Это дополнительный источник информации для установки приоритетов тестирования. Большая часть времени, отведенного на тестирование, должна затрачиваться на проверку наиболее часто используемых функций, конфигураций и операций.

Руководитель и группа наиболее квалифицированных тестировщиков должны анализировать и, в конечном счете, **утверждать план тестирования** комплекса программ, в котором описаны требования к тестированию и представлена **матрица соответствия системных требований и тестовых сценариев** [4]. Матрица отслеживания требований тестами позволяет проследить системные требования и сценарии использования системы, а также покрытие требований тестовыми процедурами. Матрица отслеживания требований должна определять каждое требование, которое проверяется группой тестирования, а также метод его верификации. Важно то, что матрица отображает тестовые процедуры на системные требования или сценарии использования системы и помогает убедиться в том, что системные требования или сценарии использования системы, проверенные при тестировании, успешно реализованы.

Матрица соответствия должна содержать информацию о требованиях, а также показывать взаимосвязь между требованиями и другими результатами проекта. **Планирование тестирования** включает как определение требований к тестам, так и разработку процессов управления этими требованиями. Механизмом согласования с заказчиком объема и глубины требований к тестам является план тестирования. Подпись заказчика под планом тестирования означает, что он

утверждает степень покрытия требований, проверяемых группой тестирования (см. рис. 9.2).

Планирование – компонент методологии тестирования, включает анализ специалистами всех работ и процессов, необходимых для его проведения. Оно предполагает, что процессы тестирования, методы, методики, специалисты, инструменты, план-график и оборудование **организованы и эффективно применяются**. Группа специалистов тестирования и выявления дефектов должна начинать подготовку плана тестирования с получения или создания шаблона плана тестирования, а затем его уточнять. Когда план тестирования комплекса программ разработан, обновлен и полностью описывает процессы тестирования, он становится руководящим инструментом для выполнения Программы тестирования. Он уточняется посредством корректировок целей, задач и стратегий тестирования, а также изменений требований к тестам, фиксирует параметры Программы тестирования, которые должны быть документированы.

Планирование работ по тестированию должно учитывать профессиональную квалификацию специалистов, ресурсы и работы, которые необходимо выполнить, чтобы своевременно **подготовить тестовую среду**. Тестировщики должны определить требования к аппаратному, программному и сетевому обеспечению с целью создания и поддержки адекватных изменений тестовой среды. Нужно планировать работы по приобретению, установке и настройке компонентов, моделей или динамических генераторов тестовой среды. Создание плана тестирования – **итеративный процесс**, требующий обратной связи с различными участниками проекта и их согласия с определенными в нем процессами, стратегиями тестирования и сроками выполнения работ. Руководитель должен **утвердить стратегию тестирования** и тестовые процедуры, которые должны быть подробно описаны в плане тестирования, и определять какие сценарии и тесты когда будут выполняться. Кроме того, предполагается, что заказчик согласен с тем, что план тестирования и связанные с ним тестовые сценарии достаточно проверяют покрытие тестами системных требований или сценариев использования системы. Подробное **изучение каждым специалистом – тестировщиками системных требований или сценариев применения системы** вместе с тщательным определением параметров плана тестирования и требований к

тестам необходимы для эффективного тестирования программного продукта.

План тестирования должен определять объем работ по тестированию [13]. Полезно выстраивать *структуру работ*, в которой на одном уровне определяются категории работ по тестированию, а на другом уровне – подробные описания и исполнители работ. Структура детализации работ используется в сочетании с хронометражем для определения времени выполнения каждого из этапов тестирования. Кроме того, план тестирования должен отражать и учитывать *оценки затрат* на тестирование. Оценка затрат может определять число сотрудников группы тестирования в часах или в количестве людей, если на выполнение определенного объема работ выделяется конкретный срок. По возможности в план тестирования помещаются такие оценки затрат, как планируемое число тестовых процедур и сценариев.

Спецификации тестов должны обеспечивать дополнительный контроль корректности требований и верификацию взаимодействия компонентов на соответствующем уровне описания компонентов и комплекса программ – рис. 9.3.

Независимая разработка спецификаций тестов *на основе* спецификаций требований создает базу для обнаружения, какие требования не тестировались или принципиально не могут быть проверены тестированием. Таким образом, *верификация спецификаций требований тестов* к функциям и характеристикам программных компонентов и комплекса могут использоваться с *двумя целями* [13, 34]:

- для разработки и проверки программ и интерфейсов взаимодействия программных компонентов разных уровней в комплексе программ;
- для создания требований к скоординированному комплексу тестов для проверки совокупности компонентов, обеспечивающих взаимную проверку реализации спецификаций требований комплексом программ.

В результате совокупности *спецификаций требований к тестам* могут применяться *как эталоны и вторая адекватная форма описания содержания компонентов и комплекса программ* для сквозной верификации спецификаций требований к тестам сверху вниз, а также сами подвергаться верификации на корректность соот-

ветствия исходным требованиям к компонентам текстов программ разного уровня [34].

Такие *параллельные взаимные проверки* спецификаций требований и текстов программ, и спецификаций тестов способствуют выявлению и исключению множества вторичных дефектов и ошибок комплексов программ. Впоследствии, эти спецификации тестов должны использоваться для непосредственного тестирования исполнения требований к программным компонентам соответствующего уровня. Кроме того, параллельная и независимая разработка, с одной стороны, спецификаций программ и спецификаций тестов, а также их реализации, с другой стороны, позволяет распараллеливать работы, что ведет к сокращению сроков создания компонентов и комплексов программ.



Рис. 9.3

Реализация этих *целей верификации и тестирования* может производиться разными методами и *независимыми специалистами* – программистами, интеграторами и тестировщиками, что позволяет использовать результаты их деятельности для сравнения одних и тех же описаний программ, представленных на языках программирования и на языках тестов. Особенности описаний и реализации программ, а также *мышления их разработчиков (программистов)* – на основе требований, функций, характеристик структуры и исполнения программ, существенно *отличаются* от представлений и методов описаний тех же *функций комплекса программ профессиональными тестировщиками* – создателями сценариев и эталонов требуемых результатов тестирования. Они акцентируют свою деятельность на конкретных процедурах проверки функционирования, возможных результатах и взаимодействии компонентов комплекса программ. Это позволяет выявлять вторичные дефекты и повышать качество путем *сопоставления двух методов* и результатов описания *одних и тех же* функций и характеристик программ за счет того, что мала вероятность одинаковых ошибок в сценариях и реализациях тестов и в описаниях требований к функциям и характеристикам программ [34].

Тестирование на базе *верифицированных требований к тестам* должно обеспечивать качество комплексов программ. Такие сценарии предназначены для поэтапной проверки внутренней непротиворечивости и полноты реализации требований к функциям и характеристикам качества комплексов программ. Требования к тестам должны проверяться на корректность системных взаимосвязей функциональных компонентов *сверху вниз* и на внутреннюю корректность взаимодействия между требованиями для компонентов и их пригодности для последующего тестирования и применения. Последовательная разработка и верификация спецификаций требований к программным компонентам должна обеспечивать корректность их логических и функциональных связей и применения. Однако этот процесс *не гарантирует* полноту и корректность реализации всех *требований*, так как обычно эти работы проводятся частично вручную и могут отсутствовать четкие эталоны для их проверки.

Для определения специалистами *успешности тестов* их результаты должны оцениваться и анализироваться. В большинстве случаев, успешность тестирования подразумевает, что тестируемый комплекс программ функционирует так, как ожидалось, и в процессе

работы не приводит к непредусмотренным последствиям. Однако любое непредусмотренное поведение может стать источником сбоев при изменении конфигурации или условий функционирования системы, поэтому требуют внимания, как минимум, с точки зрения идентификации причин таких дефектов. Перед устранением обнаруженного сбоя необходимо определить и **зарегистрировать те усилия**, которые необходимы для анализа проблемы и ее устранения. В тех случаях, когда результаты тестирования особенно важны, например, в силу критичности обнаруженного дефекта, может быть сформирована специальная группа специалистов для его анализа.

Должны быть документированы **профессиональная квалификация и навыки сотрудников**, необходимых для проведения тестирования. Состав группы тестирования с требуемым качеством и квалификацией может быть обозначен в требованиях к знаниям тестировщиков. **Руководитель** должен оценивать разницу между требуемой квалификацией и реальной подготовкой персонала, чтобы определить дополнительные направления для обучения. Планируемое обучение следует отразить в плане-графике, чтобы соответствующие работы по тестированию не предшествовали обучению. Также необходимо определить и документировать в плане **роли и ответственность сотрудников** группы тестирования с учетом особенностей комплекса программ.

План тестирования должен иметь определенные рамки, отражающие **ограничения по сотрудникам, по человеко-часам и по графику реализации**. В плане тестирования также целесообразно отражать допущения, предварительные условия и риски тестирования. Сюда включаются все события, действия или обстоятельства, которые могут помешать выполнению тестирования в срок. С разработкой плана должно быть связано определение функций, разработка которых имеет большое значение для успеха проекта и функций, разработка которых связана с наибольшим риском. Определение наивысшего риска дает возможность группе тестирования сосредоточить усилия на функциях высокой значимости с точки зрения достоверности результатов тестирования.

Требования к тестам должны заноситься специалистами в **базу данных и/или в матрицу отслеживания требований**. В базе данных или в матрице каждому требованию к тестам сопоставляется идентификационный номер компонента системной архитектуры про-

граммного продукта. Необходимо, чтобы группа специалистов по тестированию как можно раньше получила обратную информацию о матрице отслеживания изменения требований от конечных пользователей или заказчиков системы. Это способствует достижению согласия по поводу методов верификации, обеспечивающих проверку или контроль каждого требования. Принятие этого решения особенно важно, поскольку методы верификации отличаются сложностью и затратами времени. ***Раннее получение информации по матрице от заказчика*** позволяет группе специалистов тестирования увеличить время реакции на возможные изменения. Поскольку матрица отслеживания требований определяет выполняемые тестовые процедуры, одобрение матрицы заказчиком также означает его удовлетворение степенью покрытия тестами требований или сценариев использования системы. При проведении приемо-сдаточных испытаний ***заказчик и тестировщики*** анализируют матрицу, чтобы проверить покрытие тестами системных требований или сценариев использования системы.

К ключевым элементам планирования тестирования относится ***тестирование эксплуатационной и технологической документации*** [34]. Группа тестирования должна исчерпывающе документировать реализацию плана тестирования, а тестировщики обязаны подробно изучить содержание этих планов. Эта группа должна получить одобрение плана тестирования у ***конечного пользователя или заказчика***. Кроме того, предполагается, что заказчик согласен с тем, что план тестирования и связанные с ним тестовые сценарии достаточно проверяют покрытие тестами системных требований или сценариев использования системы.

Эффективный план тестирования, включающий в себя автоматизацию тестирования программного продукта, должен иметь свой ***собственный жизненный цикл*** развития и изменений, в который входят корректировки планирования стратегий и целей, определения требований к тестам, анализа, проектирования и кодирования тестов. ***Ресурсы тестирования ограничены***, а способов тестирования изменений систем много (см. рис. 9.2). Представление работ по тестированию в графической форме позволяет специалистам по тестированию оценивать границы и масштаб плана тестирования (см. лекцию 6). Завершив анализ, специалистам – тестировщикам следует переходить к созданию моделей реализации и оформлению плана тестиро-

вания. Структуру организации тестирования (архитектуру тестирования) можно представлять двумя способами. Один метод организации тестовых процедур, известный как архитектура тестирования **на базе архитектуры системы**, разбивает тестовые процедуры по функциям и компонентам системной архитектуры, по логическому принципу приоритетов в ее иерархии. Второй метод архитектура тестирования **на базе выбранных методов** связывает тестовые процедуры с различными методами тестирования, представленными в модели методов тестирования.

Тестировщики должны придерживаться утвержденного *графика проведения тестовых процедур*. По окончании выполнения теста следует производить анализ выходных данных тестирования и готовить документацию по результатам тестирования. Организация и планы комплексного, системного тестирования и приемо-сдаточных испытаний в совокупности представляют собой этапы, необходимые для тестирования системы в целом. Комплексное тестирование версии программного продукта направлено на *проверку его полного функционирования*. В процессе комплексного тестирования функциональные компоненты интегрируются и тестируются совместно на основе управляющей логики системы. Поскольку одни компоненты могут состоять из других, то часть комплексного тестирования может проводиться в ходе иерархического тестирования компонентов. В процессе системного тестирования специалист должен проверять интеграцию отдельных частей, в совокупности составляющих систему в целом.

Организационная структура и профессиональная квалификация коллектива специалистов при корректировках сложных комплексов программ должна учитывать: цели и функции тестирования; взаимодействующие организации; службы проектирования; систему обеспечения качества и средства, которые могут быть привлечены, охватывая, если необходимо, субподрядчиков и поставщиков. Организационная структура тестирования представляет наибольшее значение с точки зрения постоянного повышения зрелости технологии процессов (см. лекцию 11) и возможностей тестирования [34]. Группа тестирования, продолжающая существовать по окончании проекта, способна сохранять и усовершенствовать в новых проектах технологические процессы предприятия, процедуры и знание инструмента, а также использовать полученные знания. При организации сложного

проекта следует идентифицировать инстанцию, уполномоченную **утверждать версии программных продуктов и документов** для поставки заказчику и пользователям, и любые изменения к ним (см. таблицу 7.1).

Для сокращения и устранения дефектов разработки посредством адекватных контрмер необходима четкая организация коллектива специалистов и автоматизация процессов исправления дефектов, которые позволяют избегать множества вторичных ошибок, обусловленных недостаточной координацией проводимых корректировок и формирования новых версий сложных программных продуктов. Этому должна способствовать утвержденная дисциплина и иерархия принятия решений на координированные изменения компонентов и комплекса в целом должностными лицами, поддержанная методами и средствами защиты от несанкционированного доступа при выполнении корректировок документов специалистами различной квалификации и **права доступа к модификациям** компонентов на разных уровнях проекта.

Следует установить **полномочия специалистов** или групп для санкционирования и выполнения изменений документов и корректировок на каждом уровне разработчиков изменений (см. таблица 7.1). Изменения, которые воздействуют на программный продукт, уже находящийся на эксплуатации, должны быть предоставлены заказчику в соответствии с установленными контрактом формами и процедурами. На основе проведенного анализа и тестирования персонал должен разрабатывать варианты реализации изменений и документально оформлять: сообщение о каждом дефекте или заявку на внесение изменений; результаты их анализа и варианты реализации изменений, **оценку их влияния на функциональную пригодность**. Регистрация и учет истории этого процесса обеспечивает возможность его контроля и пошагового восстановления выполненных изменений (отката) документов для **выявления вторичных дефектов**, внесенных в процессе разработки очередной версии программного продукта.

Если предполагается, что программный продукт будет иметь длительный жизненный цикл или ожидаются значительные корректировки, то следует рассмотреть и учсть наиболее детальные требования к методике организации работ и к коллективу, ответственному за совершенствование тестирования и документирование комплекса программ. Если **изменения в программе или в данных невелики**, то тес-

тировщики обычно стремятся ограничиться компонентами, непосредственно связанными с выполненной корректировкой. Для этого выделяются для тестирования компоненты, которые связаны по информации или по управлению с теми, которые подверглись даже малым изменениям. Однако следует учитывать, что корректировки программных компонентов сами могут содержать ошибки и требуют тщательного тестирования не только тех частей, где внесены изменения.

В процессах производства и тестирования крупного программного продукта может участвовать большое число специалистов различных направлений и квалификации, которые при необходимости целесообразно объединять в единый коллектив – *службу тестирования на соответствие требованиям и управления конфигурацией программного продукта* (см. лекцию 7). Руководитель проекта является высшим должностным лицом, принимающим важнейшие решения по внесению изменений и корректировке требований и конфигурации сложных комплексов программ. Он взаимодействует с заказчиком и пользователями, определяющими модификации для согласования изменений требований к системе. Заказчик системы должен оценивать и утверждать наиболее крупные изменения, заметно влияющие на условия контракта, технические требования или стоимость программного продукта.

Недостатки или отсутствие достоверного обоснования необходимых ресурсов специалистов (особенно времени) для тестирования программных продуктов, могут являться причинами острых *конфликтов* между заказчиками и разработчиками. Поэтому целесообразно *активно привлекать заказчиков* к управлению тестированием, чтобы обеспечить своевременность производства программных продуктов в условиях ограниченных ресурсов. Необходимы согласия заказчиков при принятии основных решений о модификации функций и характеристик комплексов программ, и только они могут реально определить, как получить программный продукт с необходимыми функциями и высокого качества, выполненный в срок и в пределах выделенного бюджета.

Прогнозирование заказчиком и руководителями *затрат ресурсов на тестирование* крупных комплексов программ возможно более или менее корректно, на основе обобщения статистических данных ряда предшествующих проектов [5, 26, 54]. В данном случае, задача ограничена только ориентировочным перечнем основных со-

ставляющих затрат, которые целесообразно учитывать в процессе тестирования. Этот перечень может использоваться как ориентир при подготовке плана тестирования. Обычно наиболее важным для реализации проекта и зависящим от большинства его особенностей и факторов является **трудоемкость, непосредственно определяющая стоимость** тестирования создаваемого комплекса программ. Значения длительности разработки и числа специалистов взаимосвязаны и в некоторых пределах могут размениваться. Поэтому оценки этих показателей затрат можно варьировать, и **при недостаточном числе квалифицированных специалистов** естественно возрастает длительность разработки, хотя трудоемкость может остаться практически неизменной.

Оценка трудозатрат специалистов на тестирование является одной из наиболее важных и сложных компонентов обеспечения соответствия требованиям программных продуктов и квалификации специалистов. Затраты труда и времени, необходимых для квалификационного тестирования могут составлять существенную часть, **до половины стоимости продукта**, при этом важно для успеха, чтобы тестирование проводило достаточно большое число специалистов и у них было достаточно времени на качественное выполнение своих задач [4, 27].

При тестировании комплексов программ основным лимитирующим ресурсом обычно являются **допустимые трудозатраты специалистов**, а также **ограничения на сроки** разработки версии программного продукта, на параметры ЭВМ и технологию проектирования. Одним из наиболее важных компонентов планирования тестирования является оценка трудоемкости и времени, необходимых для его выполнения (см. лекцию 5). Ограничения реальных ресурсов на верификацию и тестирование определяют достижимое качество версий комплексов программных продуктов. Время, необходимое для выполнения той или иной задачи тестирования, зависит от количества специалистов, однако эта зависимость не обязательно линейная. Суммарная продолжительность работ по тестированию зависит от продолжительности решения отдельных функциональных задач, однако это не простое суммирование, поскольку некоторые задачи можно решать параллельно и одновременно с другими.

Затраты специалистами – тестировщиками на обнаружение и устранение дефектов и ошибок в программе определяются

двумя факторами: затратами на обнаружение источника каждой ошибки и затратами на устранение выявленных ошибок при формировании очередной версии. Чем меньше ошибок в программе, тем труднее они обнаруживаются, т.е. тем выше затраты на выявление каждой ошибки. Затраты на устранение ошибок и корректировку программ пропорциональны числу дефектов, выявляемых между очередными версиями программного продукта. Непрерывно требуются затраты для контроля состояния версий комплекса программ и обеспечения их сохранности. По опыту работ, широко тиражируемый комплекс программ объемом $\sim 10^5$ строк может требовать непрерывных усилий коллектива в составе десятка и более специалистов для устранения ошибок, корректировок версий и документации [4, 26, 41, 49].

Стратегии и измерения эффективности тестирования должны быть объединены руководителями и специалистами производства в единый процесс как **деятельности по обеспечению качества**. Хотя в большинстве современных методов разработки, тестирование выходит на передний план и является одной из базовых практик. Систематизированное тестирование и прогнозирование на основе полученных результатов часто подменяется отдельными работами в этой области, не позволяющими добиться необходимого качества программного продукта. Квалифицированные коллективы руководителей и специалистов должны **уметь завершать тестирование** программ протоколами успешного тестирования компонентов и комплекса программ и актом для предъявления заказчику программного продукта, **соответствующего требованиям**.

Лекция 10

ТРЕБОВАНИЯ К ПРОФЕССИОНАЛЬНОЙ КВАЛИФИКАЦИИ СПЕЦИАЛИСТОВ, ОСУЩЕСТВЛЯЮЩИХ ДИНАМИЧЕСКИЕ ИСПЫТАНИЯ КРУПНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Требования к профессиональной квалификации специалистов для подготовки динамических испытаний крупных программных продуктов

Тестирование для обнаружения и идентификации ошибок подразумевает успешность такой процедуры, если *дефект или ошибка найдены*. Это отличается от подхода в тестировании, когда тесты исполняются для демонстрации того, что программный продукт удовлетворяет предъявляемым требованиям и, соответственно, тест считается успешным, если *не найдено ошибок и дефектов*. Тестирование программных продуктов может использоваться для обнаружения и демонстрации наличия ошибок и дефектов, но никогда не гарантирует их отсутствие [4, 19]. Основная причина этого в том, что *полное, всеобъемлющее тестирование недостижимо* для реального сложного программного продукта. Когда квалифицированные специалисты оценивают требования к тестам и методам тестирования, надо четко определять, что подразумевается под их эффективностью, желательно, в количественных величинах. Возможные варианты интерпретации этого понятия, в частности, – число тестов, необходимых для обнаружения очередного дефекта или отношение количества всех обнаруженных дефектов к числу дефектов, найденных с применением заданного метода. Только обладая такого рода данными можно говорить о корректности *оценки требований к тестам*.

Характеристики качества функционирования программных средств зависят не только от их внутренних свойств, но и *от свойств внешней среды*, в которой они применяются. Для сокращения неопределенностей и прямых ошибок при оценивании качества про-

граммного продукта необходимо до начала динамических испытаний определить основные параметры внешней среды, при которых должен функционировать комплекс программ с требуемыми характеристиками при оценивании его качества и эксплуатации. Для этого заказчик и руководители проекта совместно должны структурировать, описать и согласовать **модель динамического функционирования внешней среды** и ее параметры в среднем, типовом режиме применения программного продукта, а также в наиболее вероятных и критических режимах, в которых должны обеспечиваться требуемые характеристики качества его функционирования. Такая **модель должна отражать и фиксировать характеристики**:

- внешних потоков информации, в том числе, их распределение по видам источников, характеристикам качества данных и возможности их дефектов;
- интенсивность и структуру типовых сообщений от оперативных пользователей и администраторов и их необходимую квалификацию, отражающуюся вероятностью ошибок и качеством выдаваемой информации;
- возможных негативных и несанкционированных воздействий от внешней среды при применении продукта;
- необходимые характеристики вычислительных средств, на которых предназначено функционировать комплексу программ с требуемым качеством.

Квалифицированные руководители и специалисты при генерации динамических тестов для испытаний крупных программных продуктов **должны уметь** формализовать функции и характеристики **тестовой внешней среды системы**. Руководитель динамического тестирования предварительно должен сформулировать требования к тестовым процедурам и применению генераторов динамических тестов. Требования должны применяться для разработки как ручных тестов, так и автоматизированных тестовых процедур и моделей генерации динамических тестов многократного применения. Главное в этих функциях – возможность их многократного использования, чтобы сводить к минимуму дублирование работ по созданию тестовых процедур и документов, повышать степень многократного их применения – рис. 10.1.

Квалифицированные специалисты для подготовки динамических испытаний крупных программных продуктов должны знать и уметь:

- определять характеристики динамических тестов, необходимых для обнаружения дефектов и ошибок, и для демонстрации, что продукт удовлетворяет требованиям заказчика;
- согласовывать требования к функциям и характеристикам внешней среды с заказчиком для динамических испытаний программного продукта;
- оценивать сложность необходимых динамических тестов и полноту тестового покрытия требований программного продукта при испытаниях;
- устранять дефекты и ошибки в требованиях к программному продукту и адекватных тестах для динамических испытаний;
- оценивать затраты ресурсов и квалифицированных специалистов, необходимых для разработки комплекса динамических программных имитаторов внешней среды;
- оценивать затраты квалифицированных специалистов для выполнения верификации и эксплуатации динамических тестов внешней среды.

Рис. 10.1

При испытаниях крупных комплексов программ, квалификация и число **необходимых специалистов для генерации динамических тестов** может быть соизмеримым с количеством участвующих при создании основных функций комплексов программ. Это определяется принципиальным **соответствием сложности необходимых наборов тестов** и тестового покрытия программ и **сложности функций**, реализуемых испытываемым программным продуктом.

Создание представительных совокупностей динамических тестов возможно путем использования реальных объектов внешней среды или с помощью программных имитаторов, адекватных этим объектам по результатам функционирования и генерируемой информации. При этом специалисты по испытаниям должны **уметь оценить** – какой метод и когда выгодней по затратам на генерацию тестов и по обеспечению необходимой степени покрытия тестами.

Имитаторы динамических тестов необходимы не только для оценивания достигнутых характеристик качества комплексов программ, но также для их квалификационного тестирования и испыта-

ний при создании версий программного продукта. Поэтому затраты на квалифицированных специалистов по моделированию программных имитаторов и их экономическую эффективность целесообразно рассматривать в производстве с учетом всего комплекса задач, которые они способны и должны решать. Анализ эффективности программной имитации внешней среды при разработке и испытаниях качества программного продукта целесообразно делить **на две части**: оценка факторов, определяющих эффективность средств имитации тестов и оценка экономического выигрыша при моделировании внешней среды на ЭВМ по сравнению с натурными испытаниями в реальных системах.

Программная имитация внешней среды на ЭВМ может обеспечивать широкие наборы тестов и достаточно полные тестовые покрытия комплексов программ и компонентов при испытаниях, в том числе за пределами характеристик реально существующих или доступных источников тестов, а также соответствующие критическим или опасным ситуациям функционирования объектов внешней среды. Для каждого параметра, отражающего внешнюю среду, отношение диапазона или числа тестов, возможных при программной имитации на ЭВМ по сравнению с натурными экспериментами, может служить оценкой величины возрастания достоверности определения качества при испытаниях программного продукта.

При подготовке динамических испытаний программного продукта **специалистам необходимо уметь учитывать** не только соотношение размеров областей изменения параметров тестов, но и распределение вероятностей значений каждого параметра в этих областях для реальных и перспективных объектов внешней среды. Некоторые значения динамических тестов не только трудно создать при натурных экспериментах, но они являются маловероятными в реальных условиях. Однако такие, даже маловероятные ситуации и значения тестов могут быть **критическими и/или особо важными** для функционирования всей системы, для которой испытывается программный продукт. Выбор и имитация подобных ситуаций позволяют отрабатывать и испытывать качество в критических маловероятных ситуациях, которые невозможно или опасно создавать на реальных объектах, но без их выполнения некоторые продукты недопустимо эксплуатировать в критических системах управления и обработки информации.

Требования к тестам и требования к программному продукту должны **отражать один и тот же объект**, но в разной форме. Поэтому сложность представительного описания комплекса тестов **соподразумима** со сложностью описания полной совокупности требований к функциям и характеристикам соответствующего комплекса программ **и может содержать столько же, но других дефектов и ошибок**. Жизненный цикл разработки требований к тестам и процессов испытаний программных комплексов должен проходить во времени параллельно динамике изменения требований и жизненному циклу реализации комплекса программ. Таким образом, представлению требований к программному продукту должно полностью соответствовать его равноправное содержание в форме требований к тестам для проверки их **взаимного соответствия**. Вследствие этого **трудоемкость и другие ресурсы** для разработки адекватных тестов должны соответствовать ресурсам и трудоемкости при создании и реализации корректных требований, определяющих полностью программный продукт. Однако на практике обычно это не учитывается, и выделяемые ресурсы на создание динамических тестов оказываются значительно меньше и ограничивают испытания реализации требований к крупным комплексам программ.

Выбор и формирование требований к характеристикам при динамических испытаниях программных продуктов должны начинаться с определения исходных данных. Для корректного выбора и установления требований к характеристикам качества, прежде всего, необходимо определить основные свойства и особенности программного комплекса. Чтобы при построении требований к программному продукту учитывать реальную ситуацию взаимодействия с заказчиком проекта при испытаниях, разработчикам целесообразно анализировать, учитывать и по возможности устранять некоторые традиционные **негативные человеческие факторы** [4,34]:

- образ, размер и границы крупного комплекса программ никогда заранее не могут быть определены точно;
- исходные потребности программного продукта полностью существуют только в головах «экспертов», заказчиков и специалистов, работающих в предприятии, и некоторые никогда не фиксируются в письменном виде;
- разработчики зачастую получают двусмысленную и неполную информацию от заказчика, поэтому при разработке им приходится

дится делать предположения и уточнения требований к тестам и испытаниям;

- взаимодействие между разработчиками и заказчиками иногда ограничивается внешним видом пользовательского интерфейса и не затрагивает того, что же действительно они собираются делать с помощью программного продукта;
- заказчики утверждают и подписывают требования к испытаниям с дефектами, а затем постоянно изменяют их.

Проекты комплексов программ, как правило, инициируются заказчиком с первичным **объемом требований к функциональным возможностям, превышающим** тот, который разработчик может реализовать, обеспечив приемлемое качество при заданных ресурсах. Тем не менее, квалифицированным специалистам необходимо ограничиваться, чтобы иметь возможность предоставить в срок и испытать достаточно целостный и качественный программный продукт. Поэтому следует выделять требования, которые обязательно должны быть реализованы и испытаны в версии программного продукта при имеющихся ресурсах проекта с **минимальными дефектами реализации требований**. Привлечение заказчика к итерационному решению проблемы требований с минимумом дефектов, для управления масштабом и функциями комплекса программ повышает взаимные обязательства сторон, способствует росту взаимопонимания и доверия между заказчиком и разработчиками.

Формализация критериев качества требований к результатам испытаний позволяет сокращать дефекты и прояснить некоторые **нечеткие требования**. Для них в одних случаях можно определить согласованный критерий качества, а в других случаях получается критерий, по которому не достигнуто соглашение с заказчиком. Не каждое требование в его содержании может иметь критерий качества, который можно использовать для проверки того, что оно не содержит ошибок и удовлетворяет заказчика. Подспорьем в поиске неосознаваемых и некорректных требований является создание моделей и/или прототипов, с помощью которых можно показать заинтересованным сторонам различные точки зрения на некоторые требования и их возможные дефекты при испытаниях.

Чтобы протестировать **требования на значимость и отсутствие дефектов**, следует сопоставлять требования и сформулированные цели испытаний программного продукта:

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

Требования к профессиональной квалификации руководителей и специалистов по организации динамических испытаний крупных программных продуктов

Любые *испытания ограничены допустимым количеством и объемом проверок*, а также длительностью работы коллектива испытателей, поэтому не могут гарантировать абсолютную проверку качества продукта. Для повышения достоверности определения и улучшения оценивания характеристик после внутренних испытаний, комплекс программ целесообразно передавать квалифицированным пользователям на *опытную эксплуатацию в типовых условиях*. Это позволяет более глубоко оценить эксплуатационные характеристики созданного комплекса и устранить некоторые дефекты и ошибки. Опытную эксплуатацию целесообразно проводить квалифицированным специалистам проекта с участием испытателей – заказчиков и некоторых пользователей, назначаемых заказчиком. Результаты и характеристики качества опытной эксплуатации после испытаний главного конструктора могут учитываться при проведении заказчиком квалификационных испытаний для их сокращения (рис.10.2).

Руководителям испытаний необходимо уметь проверять, что разрабатываемый комплекс программ удовлетворяет каждому из зафиксированных и утвержденных требований на программный продукт. Нужно выделять каждое такое требование, чтобы *проследить его развитие и устранение дефектов* в ходе подробного анализа, производства и, наконец, испытаний. Каждый этап разработки комплекса программ формирует, уточняет и реорганизует требования, чтобы сделать их как можно ближе к назначению нового программного продукта. Во избежание потерь и искажения требований **нужно уметь** преобразовывать исходные требования в решения для того, чтобы использовать их при испытаниях.

Квалифицированные руководители и специалисты для организации динамических испытаний крупных программных продуктов должны знать и уметь:

- составить и утвердить у заказчика требования на выполнение динамических испытаний программного продукта и системы;
- разработать и утвердить Программу и план динамических испытаний программного продукта на соответствие утвержденным требованиям;
- разработать методики динамических испытаний на соответствие требованиям к программному продукту и Программе испытаний;
- определить, испытать и паспортизировать средства имитации тестов внешней среды;
- выполнить динамические испытания и оформить протоколы по разделам Программы и требований на программный продукт;
- разработать комплект адекватной документации на результаты динамических испытаний программного продукта;
- оценить затраты трудоемкости, времени и ресурсов на проведение полноценных динамических испытаний программного продукта.

Рис. 10.2

Оценивание качества программного продукта при квалификационных приемо-сдаточных испытаниях проводится комиссией заказчика, в которой участвует руководитель (главный конструктор) проекта и некоторые квалифицированные специалисты или аттестованная сертификационная лаборатория. ***Комиссия при испытаниях должна руководствоваться следующими документами:***

- утвержденными заказчиком и согласованными с разработчиком контрактом, техническим заданием и спецификациями требований на программный продукт;
- действующими государственными и ведомственными стандартами на жизненный цикл и испытания комплексов программ на технологическую и эксплуатационную документацию, а также стандартами де-факто, согласованными с заказчиком для использования;
- Программой испытаний по всем требованиям контракта, технического задания и спецификаций;

- методиками испытаний, охватывающими каждый раздел требований технического задания, спецификаций и Программы испытаний;
- комплектом адекватной эксплуатационной и технологической документации на комплекс программ.

Программа испытаний является планом проведения серии экспериментов и должна разрабатываться квалифицированными руководителями и специалистами с позиции **допустимой минимизации объема тестирования в процессе проведения испытаний** для проверки выполнения требований технического задания и соответствия предъявленной документации. Программа испытаний, методики их проведения и оценки результатов, разработанные совместно заказчиком и разработчиком, должны быть согласованы и утверждены. Они должны содержать уточнения и детализацию требований технического задания и спецификаций для данного комплекса программ, а также гарантировать корректную проверку всех заданных характеристик качества. **Программа испытаний должна содержать следующие четко сформулированные разделы:**

- объект испытаний, его назначение и перечень основных документов, определивших его разработку;
- цель испытаний с указанием всех требований технического задания, характеристик и атрибутов качества, подлежащих проверке, и ограничений на проведение испытаний;
- собственно Программу испытаний, содержащую проверку комплектности спроектированного комплекса программ в соответствие с техническим заданием и план проведения испытаний для проверки по всем разделам технического задания и дополнительных требований, formalизованных отдельными решениями разработчиков и заказчика;
- методики испытаний, однозначно определяющие все понятия проверяемых характеристик качества, условия и сценарии тестирования, инструментальные средства, используемые для испытаний;
- методики обработки и оценки результатов испытаний по каждому разделу Программы испытаний.

Методика испытаний должна содержать: описание организации процесса тестирования, тестовые варианты, сценарии и процедуры, которые используются при динамических испытаниях. Каждый динамический тест должен иметь уникальный для данного проекта идентификатор;

должны быть представлены инструкции для проведения тестирования; описание аппаратуры и внешняя среда для реализации испытаний. Кроме того, должны быть приведены ссылки на проверяемые требования, указаны условия выполнения (конфигурация аппаратуры и компонентов комплекса программ), входные данные, эталонные и ожидаемые результаты, критерии оценки качества результатов, процедура проведения динамических испытаний для каждого тестового варианта, допущения и ограничения.

План испытаний программного продукта должен описывать порядок квалификационного тестирования, тестовую внешнюю среду, которая будет использоваться при тестировании, идентифицировать выполняемые динамические тесты и указывать план-график действий при испытаниях. Большой объем разнородных данных, получаемых при испытаниях крупномасштабных программных продуктов, и разнообразие возможных способов их обработки, интерпретации и оценки приводят к тому, что важнейшими факторами становятся *методики обработки и оценки результатов, а также протоколы проверки по пунктам Программы испытаний*. Результаты испытаний фиксируются *в протоколах*, которые обычно содержат следующие разделы:

- назначение тестирования и раздел требований технического задания, по которому проводились испытания;
- указания разделов методик в соответствии, с которыми проводились испытания, обработка и оценка результатов;
- условия и сценарии проведения динамического тестирования и характеристики исходных данных;
- обобщенные результаты испытаний с оценкой их на соответствие требованиям технического задания и другим руководящим документам, а также технической документации;
- описание обнаруженных дефектов и ошибок и рекомендуемых улучшений в испытываемом программном продукте;
- выводы о результатах испытаний и о соответствии созданного программного продукта определенному разделу требований технического задания и исходных спецификаций.

Представленная *организация испытаний крупных программных продуктов* ориентирована на наличие конкретного заказчика комплекса программ и ограниченное число пользователей, контролируемых заказчиком. Для обеспечения высокого качества крупных

программных продуктов реального времени необходимы соответствующие *проблемно-ориентированные интегрированные системы автоматизации динамического тестирования*, способные достаточно полно заменить испытания программ с реальными объектами внешней среды. При этом высокая стоимость и риск испытаний с реальными объектами почти всегда оправдывают значительные затраты на такие интегрированные системы, если предстоят испытания критических программных продуктов с высокими требованиями к качеству функционирования программного продукта, с длительным жизненным циклом и множеством развивающихся версий.

Ограниченные ресурсы для реализации функциональных требований при испытаниях могут *отражаться на дефектах* конструктивных характеристик: на надежности, безопасности, пропускной способности, качестве взаимодействия с внешней средой и с пользователями, качестве документации и других эксплуатационных факто-рах. Таким образом, *требования к характеристикам качества комплекса программ и допустимых рисков* должны быть проанализированы и согласованы для установления в договоре между заказчиками и разработчиками исходных компромиссных значений, пригодных для эффективного производства продукта с требуемыми функциями. Для решения этих задач необходимо управление характеристиками качества и управление рисками с целью обеспечении квалифицированными специалистами требуемой заказчиком функциональных характеристик. При этом следует учитывать, что каждый вид ресурсов в реальных условиях ограничен и может варьироваться только в некотором диапазоне [34, 59, 60].

Время или требование допустимой длительности производства продукта является невосполнимым ресурсом. Этот ресурс все больше определяет требования к качеству комплексов программ и их дефекты в процессе их производства. Жесткие *требования заказчиков к срокам* реализации продуктов, естественно, ограничивают разработчиков и испытателей. Увеличение числа привлекаемых для этого специалистов только в некоторых пределах позволяет ускорять разработку. Радикальный способ увеличения реальных затрат на разработку и испытания в ограниченное время состоит в систематизации, планировании и автоматизации технологии на всех этапах жизненного цикла комплекса программ.

В комплексах программ, могут проявляться *негативные свойства и дефекты требований*, которые проявляются ущербом –

риском при производстве и/или применении программного продукта. К таким свойствам – рискам при создании программных продуктов должны формироваться требования, ограничивающие их до допустимых пределов. Для обеспечения требуемых характеристик качества функционирования программного продукта необходима **организация контрмер** для сокращения дефектов и управления рисками под руководством координатора взаимодействия заказчика и производителей, прежде всего, при обеспечении функциональной пригодности продукта [30, 42, 51]. Для этого при управлении проектом руководителям **следует уметь: идентифицировать в требованиях возможные угрозы и риски**, имеющие как внешние, так и внутренние причины; проводить их количественную оценку; разработку откликов и контрмер для сокращения рисков и контроль реализации откликов.

Требования обеспечения минимальных допустимых рисков функциональной пригодности имеют доминирующее значение и обычно должны быть, в первую очередь, подчинены сокращению рисков функционирования системы и программ. Поэтому анализ рисков и возможных угроз целесообразно проводить систематизировано, начиная с установления требований к допустимым рискам функциональной пригодности [55, 61].

Квалифицированным руководителям необходимо уметь определять и выделять инструменты тестирования, которые будут применяться для выполнения проекта. Требования к тестам для комплексов программ можно также получить на основе представлений о логике архитектуры системы. Разрабатывая требования к тестам на основе системных требований или сценариев использования системы, испытатели должны создавать, по крайней мере, **одно требование к тестам на каждое системное требование**. Системные требования или сценарии использования системы, разбитые на составные части на уровне спецификации программного продукта или архитектуры, обычно проверяются во время комплексного тестирования.

Управление требованиями к тестам включает в себя хранение требований, отслеживание связей, оценку рисков требований к тестам, выстраивание последовательности требований к тестам и определение методов верификации тестов. Отслеживание связей состоит в построении матриц для отображения тестовых процедур на требования к тестам и **возможных дефектов тестовых процедур** (см. лек-

цию 9). Квалифицированные испытатели **должны уметь** описать в плане способ управления требованиями к тестам.

Анализ и формирование **требований к тестам и выявление возможных их дефектов** должны проводиться квалифицированными руководителями при определении целей, задач и стратегий испытаний, а также при выборе инструментов тестирования. Для сложных комплексов программ важно иметь пригодные для подготовки тестирования **системные требования и сценарии использования функций** системы и программного продукта. Эти требования необходимо анализировать и определять в терминах требований к тестам, устанавливающим их содержание и корректность.

Поскольку в крупных комплексах программ подвергнуть тестированию абсолютно все невозможно, **специалистам важно уметь выбирать, что обязательно** нужно испытать и какие **возможные дефекты обязательно** устраниТЬ. Если допустить полный перебор сценариев в тестировании, тестовое покрытие будет избыточным, и для отладки программного продукта потребуется значительное время, что может поставить под угрозу срок сдачи продукта заказчику. Если тестирование окажется недостаточным, то увеличится риск пропуска дефекта или ошибки, устранение которого будет стоить дорого, особенно после сдачи программного продукта в эксплуатацию.

Для обеспечения высокого качества программных продуктов стратегию создания **требований к тестам** целесообразно строить, **основываясь на требованиях к функциям и характеристикам** крупных комплексов программ. Требования к тестам можно дополнить на основе анализа логики архитектуры системы структурным методом. Он может применяться в зависимости от условий договора или особых требований к безопасности или ограниченным рискам системы. Например, покрытие полной структуры решений для некоторых функций может быть необходимо в аэрокосмических и других критических системах с повышенными требованиями к надежности и безопасности программных продуктов.

Оценка трудозатрат при формировании требований к динамическим тестам является одной из наиболее важных и сложных компонентов обеспечения соответствия требованиям программных продуктов. Получение **оценки** трудозатрат на выполнение испытаний крупного комплекса программ руководители **должны уметь представить** следующими **этапами** [4, 19] (см. лекцию 5):

- определение *перечня и состава задач испытаний*, которые должны быть выполнены, эта оценка начинается с определения работ, которые необходимо выполнить для того, чтобы тестирование программного продукта считалось состоявшимся и достаточным;
- оценка *трудозатрат* на решение отдельных задач и всего процесса испытаний в соответствии с требованиями, представляющих собой объем работ, необходимых для выполнения соответствующей задачи;
- определение *времени*, требуемого для решения каждой задачи и длительности всего тестирования – суммарная продолжительность работ по испытаниям зависит от продолжительности решения отдельных функциональных задач, однако это не простое суммирование, поскольку некоторые задачи можно решать параллельно и одновременно с другими;
- оценка *рисков невыполнения графика испытаний* и формулировка планов снижения требований к тестам – следует оценивать возможные проблемы и риски, которые могут возникнуть при решении задач в запланированные промежутки времени, и предусмотреть средства решения этих проблем.

Требования к профессиональной квалификации специалистов для выполнения динамических испытаний крупных программных продуктов

Испытания надежности функционирования крупных программных продуктов состоит в определении *соответствия требованиям, функций и характеристик* программного продукта при различной интенсивности потоков информации, адекватных нормальным условиям применения программного продукта, а также критическим по составу и интенсивности для выявления предельных условий его работоспособности. Такие *условия динамических испытаний отражаются на интегральных характеристиках качества*, на снижении надежности и/или безопасности, а также на повышении рисков применения программного продукта, вследствие дефектов и ошибок программных продуктов. Локализация и устранение таких динамических дефектов обычно осуществляется вне реального времени путем применения детерминированных сценариев и тестовых процедур, а иногда за счет изменения требований заказчика –

рис. 10.3. При этом предполагается, что в контракте, техническом задании или спецификации требований **зафиксированы и утверждены заказчиком** определенные значения этих характеристик и их приоритеты.

Квалифицированные специалисты при реализации динамических испытаний характеристик крупных программных продуктов должны знать и уметь:

- выполнить динамические испытания и определение реальной надежности функционирования системы и программного продукта в штатном, форсированном режиме и при перегрузках;
- выполнить динамические испытания и определение действительной безопасности функционирования системы и программного продукта в штатном режиме при случайных и предумышленных угрозах и при перегрузках;
- испытать производительность системы с программным продуктом при динамическом использовании ресурсов ЭВМ, при перегрузках, предельной пропускной способности и при допустимом различном времени отклика;
- испытать и оценить риски функционирования программного продукта и корректности выбора эффективных контрмер для их сокращения;
- провести испытания эксплуатационной документации на практичесность и соответствие требованиям заказчика к функциям и характеристикам применения пользователями программного продукта;
- оформить и утвердить у заказчика Акт о полном завершении динамических испытаний и выполнении требований на программный продукт и систему.

Рис. 10.3

Измерения должны **проводиться квалифицированными специалистами** при функционировании готового программного продукта для сопоставления с заданными требованиями и для оценивания рисков соответствия этим требованиям [4, 34, 51]. Входные параметры динамических тестов следует задавать на основе вероятностного распределения соответствующих характеристик или их наборов при эксплуатации программного продукта, исходя из частоты возможных сценариев работы пользователей или системы.

Прямые экспериментальные методы измерения интегральных характеристик надежности программных продуктов в ряде случаев весьма трудно реализовать квалифицированным специалистам при нормальных штатных условиях функционирования крупных комплексов программ из-за больших значений времени наработки на отказ (сотни и тысячи часов), которые необходимо достигать при разработке и фиксировать при испытаниях.

При заключительных приемо-сдаточных и сертификационных испытаниях **для достаточно достоверного определения надежности** специалисты **должны уметь** организовать многочасовые и многосугодичные прогоны динамического функционирования программного продукта в реальной и/или имитированной внешней среде в условиях широкого варьирования исходных данных с акцентом на стрессовые ситуации, стимулирующие проявления **угроз надежности**.

Форсированные (стрессовые) испытания для измерения надежности программных продуктов могут осуществляться повышением интенсивности искажений исходных данных и расширением варьирования их значений, а также специальным увеличением интенсивности потоков информации и загрузки программ на ЭВМ выше нормальной. Особым видом форсированных испытаний является целенаправленное тестирование эффективности средств оперативного контроля и восстановления программ, данных и вычислительного процесса **для оценивания восстанавливаемости**. При этом не обязательно зачастую сразу устанавливать причины искажения, главная задача сводится к **максимально быстрому восстановлению** нормального функционирования и ограничению его негативных последствий.

Функциональная безопасность программных продуктов и систем зависит от отказовых ситуаций, негативно отражающихся на работоспособности и реализации их основных функций, причинами которых могут быть дефекты и аномалии в аппаратуре, комплексах программ, данных или вычислительных процессах или предумышленные угрозы функционированию программного продукта. Основными источниками отказовых ситуаций могут быть некорректные исходные требования, сбои и отказы в аппаратуре, дефекты или ошибки в программах и данных функциональных задач, проявляющиеся при их динамическом исполнении в соответствии с назначением. При таких воздействиях внешняя функциональная работоспособность сис-

тем может разрушаться не полностью, однако невозможна полноценное выполнение заданных функций и *требований к программному продукту*. В реальных сложных системах, связанных с безопасностью, возможны катастрофические последствия и отказы функционирования с большим ущербом, под воздействием лиц, заинтересованных в предумышленных нарушениях работоспособности системы или программного продукта [13, 30, 51, 55].

Понятия, методы испытаний и характеристики *функциональной безопасности программных продуктов и систем близки к аналогичным показателям для надежности*. При более или менее одинаковых источниках угроз и их проявлениях эти понятия можно разделить *по величине негативных последствий* и ущерба при возникновении отказовых ситуаций. Чем сложнее системы и чем выше к ним требования безопасности, тем неопределеннее функции и характеристики тестирования требований для обеспечения их безопасности. Неопределенности начинаются с требований заказчиков, которые при формулировке технического задания и спецификаций не полностью формализуют и принципиально не могут обеспечить достоверное содержание всего адекватного набора условий, характеристик и значений требований безопасности, которые должны быть при завершении проекта и предъявлении конечного программного продукта заказчику. Различия свойств и квалификации персонала, применяющего систему, дополнительно увеличивают неопределенность значений безопасности и трудности ее прогнозирования при испытаниях с учетом множества *субъективных факторов различных специалистов, участвующих в эксплуатации*.

Существующие технологии тестирования способствуют повышению функциональной безопасности, снижению потенциального ущерба и рисков, однако практически всегда остается открытым вопрос, насколько применяемые методы оправдывают затраты на реализацию требований заказчика. Испытания, эксплуатация и сертификация способствуют снижению неопределенности оценок эффективности и итерационному приближению к практически приемлемому уровню безопасности программных продуктов. Роль негативных воздействий и их разрушительные последствия быстро возрастают в связи с ростом сложности современных систем на базе ЭВМ и ответственности решаемых ими задач.

Квалифицированные руководители должны завершать производство программных продуктов **комплексными испытаниями и удостоверением достигнутой безопасности** систем с программным продуктом, предусматривающие возможность улучшения их характеристик путем соответствующих корректировок и совершенствования программ. Повышение функциональной безопасности целесообразно также путем анализа выявленных дефектов и **оперативного восстановления вычислительного процесса, программ и данных (рестарта)** после обнаружения аномалий и отказов функционирования программного продукта. Этому может способствовать накопление, мониторинг и хранение данных о выявленных внешних угрозах, дефектах, сбоях и отказах в процессе исполнения программ и обработки данных.

Цель испытаний производительности – продемонстрировать заказчику, что система функционирует в соответствии с требованиями, содержащимися в спецификациях на производительность и касающиеся приемлемого **времени отклика** при обработке заданного количества транзакций. Средства, обеспечивающее испытания производительности и квалификация специалистов, должны позволять оценивать влияние перегрузок. **Оценивание перегрузок** – это процесс испытания работоспособности вычислительных машин при выполнении динамических сценариев и большого потока данных с целью выяснения того, когда и где программный продукт выйдет из строя, работая под высокой нагрузкой. Эти допустимые пределы должны быть определены в системных требованиях к программному продукту, где также должна определяться реакция системы на перегрузки. При излишне высокой интенсивности поступления исходных данных может **нарушаться временной баланс** между длительностью решения требуемой совокупности задач программным продуктом в реальном масштабе времени и производительностью ЭВМ при решении этих задач. Для выявления подобных ситуаций и определения характеристик программного продукта в условиях недостаточности ресурсов ЭВМ проводятся испытания при высокой, но допустимой, в соответствие с требованиями интенсивности поступления исходных данных.

В общем случае для испытаний **качества динамического функционирования программ в зависимости от загрузки** необходимо оценивать вероятность каждой комбинации тестовых данных и изменять соответствующую ей длительность исполнения компонентов. На

практике в ряде случаев ограничиваются некоторыми средними или наиболее вероятными значениями тестовых данных, а также одним или несколькими сочетаниями исходных данных, при которых ожидаются предельные значения потоков заданий и длительностей исполнения программ, способные негативно отразиться на качестве динамического функционирования программного продукта.

По результатам испытаний могут быть решены задачи динамического оценивания *ресурсной эффективности программного продукта*, что позволяет анализировать факторы, определяющие необходимую пропускную способность ЭВМ, и разрабатывать меры для приведения ее в соответствие с потребностями. Если предварительно в процессе проектирования производительность системы не оценивалась или определялась слишком грубо, то *велик риск*, что доработки будут большими или может понадобиться заменить ЭВМ на более быстродействующую. Это обусловлено, как правило, *«оптимизмом»* разработчиков, что приводит к *занижению интуитивных оценок длительностей решения функциональных задач* и возможных предельных интенсивностей потоков внешней информации. Длительная регистрация и накопление значений ресурсной эффективности способствуют выявлению ситуаций, при которых проявляются некоторые дефекты функциональной пригодности.

Оценивание величины производительности рекомендуется для определения: загрузки операторов-пользователей, пропускной способности по числу задач в единицу времени, временной шкалы событий обработки заданий и данных в системе. Эти результаты необходимо сравнивать с требованиями заказчика и пользователей для оценивания рабочих нагрузок и достаточности производительности программного продукта в конкретной системе и внешней среде. Эти условия *следует уметь детализировать до уровня*, позволяющего формализовать *требования к допустимым предельным значениям интенсивности решения функциональных задач*: в среднем, нормальном режиме работы программного продукта с наивысшим качеством функциональной пригодности; в режиме предельной загрузки, реализующейся с определенной вероятностью; в режиме кратковременной, аварийной перегрузки, способной критически отражаться на функциональной пригодности программного продукта.

Основная задача состоит в определении вероятностей и рисков, с которыми может нарушаться соответствие – баланс между потребно-

стями в производительности для решения всей требуемой совокупности задач и реальными возможностями ЭВМ и других компонентов системы. Проблема обусловлена сложностью оптимального распределения в динамике ограниченных ресурсов ЭВМ (особенно производительности) по многим функциональным задачам, необходимостью проектирования программ с учетом этих ограничений и возможными переделками программ для того, чтобы соблюсти ресурсные ограничения. В зависимости от характеристик потоков заданий и предполагаемых длительностей их реализации, могут распределяться приоритеты на их решение, и тем самым, повышаться эффективность использования ограниченной производительности вычислительной системы для определенного программного продукта.

Эти задачи могут быть *решены экспериментально* в процессе испытаний завершенного разработкой программного продукта, однако при этом *велик риск*, что производительность ЭВМ окажется недостаточной для решения заданной совокупности задач в реальном времени, что отразится на качестве использования программного продукта и системы. Поэтому при испытаниях требуется принимать специальные меры для создания *тестов, адекватных реальному применению продукта*, а также контролируемых, наиболее тяжелых по загрузке условий функционирования комплекса программ и внешней среды.

Для корректного испытания *пределной пропускной способности* системы с данным программным продуктом *необходимы тесты*, позволяющие измерять следующие характеристики функциональных программных компонентов и комплекса:

- экстремальные значения длительностей их исполнения и маршруты, на которых эти значения достигаются;
- среднее значение длительности исполнения каждой функциональной группы программ на всем возможном множестве маршрутов и его дисперсию;
- распределение вероятностей и значений длительности исполнения функциональных групп программ.

Риски программных продуктов выявляются при испытаниях, для чего квалифицированные специалисты *должны уметь*: анализировать, выявлять и идентифицировать их источники, противоречия требований характеристик и ресурсов для их реализации, предлагать

заказчику и руководителю **рациональные и возможные контрмеры**, обеспечивающие сокращение рисков до допустимых пределов:

- определять возможные последствия, уровни потенциальных опасностей – угроз, и приоритетов каждого класса и категории рисков комплекса программ;
- планировать методы и необходимые ресурсы для реализации эффективных контрмер при сокращении каждой категории опасных, приоритетных рисков;
- распределять ответственность специалистов за реализацию сокращения конкретных рисков.

Для **выбора при испытаниях критического уровня допустимых рисков** необходимо уметь учитывать назначение, свойства и особенности проблемно-ориентированной системы, возможную последовательность потенциально опасных событий, любые смягчающие факторы и характеристики, а также природу и частоту возможных негативных последствий идентифицированных угроз в программном продукте и в системе. Для выработки плана испытаний рисков и применения контрмер при их сокращении должна быть определена и документально установлена методика применения последовательного анализа угроз, уязвимостей и возможного изменения проявлений рисков. **Умение испытывать и сокращать риски** является важнейшим в деятельности квалифицированного руководителя крупного программного продукта.

Испытания эксплуатационной документации должны гарантировать проверку **отчуждаемости программного продукта** от их первичных специалистов – разработчиков и возможность его освоения и эффективного применения достаточно квалифицированными **специалистами – пользователями**. Эти документы должны обеспечивать эффективное применение программного продукта и точно отражать его назначение, функции, характеристики и требования для использования. Для этого эксплуатационные документы необходимо **уметь испытывать** на полное соответствие выполнения всей **совокупности требований на программный продукт**, согласованных между разработчиками и заказчиком, исключать возможность некорректного использования продуктов за пределами условий эксплуатации, при которых документами гарантируются требуемые показатели качества их функционирования. Документы должны обеспечивать психологически комфортное и корректное применение комплекса программ пользователями, на основе

ясного и непротиворечивого изложения в документах технологических процедур и операций для его штатного функционирования и получения требуемых результатов. В результате документы следует использовать квалифицированными специалистами – испытателями как отдельный, независимый при производстве **третий эталон и вид тестов, для квалификационного испытания** реализации требований к функциям и характеристикам программного продукта.

Важной задачей испытаний является проверка **комплексного плана разработки документации**, реализация которого обеспечивает достоверное документирование программного продукта. Испытание документов должно зафиксировать текущее состояние системы и версии программного продукта, назначение, требования, возможности и ограничения в зависимости от режима или конкретного состояния эксплуатации, информацию о взаимодействии пользователей, поставщика, разработчика и предприятия, осуществляющих поддержку программного продукта, во время эксплуатационного периода. **Испытания эксплуатационной документации на практичесность** должны обеспечивать соответствие и актуальность документов конкретному программному продукту. Оценки **практичности** зависят не только от собственных характеристик продуктов, но также от организации и адекватности документирования процессов их эксплуатации. При этом предполагается, что в контракте, техническом задании или спецификации зафиксированы и утверждены требования к основным параметрам и качеству методов и средств документирования и поддержки использования продукта.

Наиболее полным и разносторонним испытаниям должна подвергаться первая базовая версия программного продукта. При **испытаниях очередных модернизированных версий продукта** возможны значительные сокращения объемов тестирования апробированных повторно используемых компонентов. Для выявления и регистрации дефектов в процессе эксплуатации серийных образцов в каждом из них должен быть предусмотрен некоторый минимум контрольных средств для проверки функционирования и обнаружения искажений результатов. Эти средства должны позволять фиксировать условия неправильной работы комплекса программ и характер проявления дефектов при его применении.

Завершаются испытания продукта должны предъявлением заказчику на утверждение **комплекта документов, содержащих результаты комплексных испытаний** версии программного продукта:

- откорректированные тексты программ и данных на языке программирования и в объектном коде, а также полные спецификации требований на программные компоненты и продукт в целом после полного завершения испытаний;
- Программа испытаний по всем требованиям технического задания;
- комплект методик испытаний и обработки результатов по всем разделам Программы испытаний;
- тесты, сценарии и аттестованные генераторы динамических тестовых данных, использованные для испытаний программных и информационных компонентов и версии в целом;
- результаты и протоколы квалификационного тестирования, функциональные и конструктивные характеристики программного продукта в реальной внешней среде;
- отчет о подтверждении заданного качества, полные характеристики достигнутого качества функционирования, а также степени покрытия тестами требований заказчика;
- план, методики и средства автоматизации обучения заказчика и пользователей применению испытанной версии программного продукта;
- комплект эксплуатационной документации, описание и руководство пользователя в соответствии с условиями контракта;
- руководство по инсталляции, генерации пользовательской версии и загрузке базы данных в соответствии с условиями и характеристиками внешней среды;
- отчет о экономических показателях завершенного проекта версии программного продукта, выполнении планов и использованных ресурсах;
- акт о завершении испытаний и готовности к поставке и/или предъявлению для сертификационных испытаний версии продукта.

Все документы и результаты испытаний **обобщаются в акте**, в котором делается **заключение о соответствии продукта требованиям** заказчика и о завершении работы с положительным или отрицательным итогом. При выполнении всех требований технического

задания заказчик обязан принять программный продукт, и производство считается завершенным.

Требования к профессиональной квалификации специалистов для прогнозирования характеристик ошибок в крупных динамических комплексах программ

Путем анализа и обобщения экспериментальных данных реальных разработок динамических комплексов программ предложено **несколько математических моделей**, описывающих основные закономерности изменения суммарного числа обнаруживаемых вторичных ошибок в программах в зависимости от длительности их тестирования [59, 62, 64]. Установлено, что для этого может использоваться корреляция между **тремя характеристиками ошибок** в сложных комплексах программ:

- суммарным количеством ошибок в комплексе программ или количеством неверных команд в программе по отношению к общему количеству строк текста комплекса программ;
- количеством ошибок в комплексе или компоненте программ, выявленных в единицу времени в процессе тестирования и испытаний при постоянных усилиях;
- количеством дефектов или искажений результатов, проявляющихся для потребителей при функционировании компонента или комплекса программ вследствие не выявленных ошибок в программах.

Частота проявления вторичных ошибок при функционировании динамических комплексов программ и частота их обнаружения при тестировании зависят от общего количества первичных дефектов в программе или от вероятности ошибки в строке текста программы. Квалифицированные специалисты **должны уметь регистрировать** наиболее доступное для измерения число вторичных дефектов и ошибок в программе, выявляемых **в единицу времени** в процессе тестирования. Возможна также непосредственная регистрация отказовых ситуаций и наиболее крупных дефектов результатов, выявляемых средствами оперативного контроля в процессе нормального (при эксплуатации) функционирования программного продукта. Эти три показателя связаны некоторыми **коэффициентами пропорциональности**, значения которых зависят, в частности, от интервала времени,

на котором производится их сопоставление, от быстродействия ЭВМ, от эффективности средств автоматизации тестирования и от некоторых других параметров. Однако при фиксированной технологии разработки и функционирования конкретного программного продукта эти коэффициенты имеют вполне определенные значения. Для другой подобной системы и технологии коэффициенты могут несколько измениться, однако можно предположить, что оценки, проведенные для нескольких конкретных систем, позволят специалистам прогнозировать эти характеристики – Рис. 10.4.

Квалифицированные специалисты для прогнозирования характеристик ошибок в крупных динамических комплексах программ должны знать и уметь:

- регистрировать число вторичных дефектов и ошибок в программах, выявляемых в единицу времени в процессе тестирования и испытаний;
- анализировать и обобщать экспериментальные данные процессов и времени обнаружения дефектов и ошибок при реальных разработках динамических комплексов программ;
- прогнозировать интегральное число выявляемых ошибок в зависимости от длительности тестирования и испытаний комплекса программ;
- оценивать предельную интенсивность обнаружения отказов и наработку на обнаруженную ошибку, при которой прекращается практическое улучшение характеристик программного продукта при испытаниях;
- оценивать частоту проявления ошибок на тысячу строк кода, которую можно считать допустимой для динамического программного продукта и системы реального времени.

Рис. 10.4

Теоретические исследования подтвердили эти связи и позволили создать ***методы статистического прогнозирования*** интегрального числа выявляемых ошибок в зависимости от длительности тестирования комплекса программ. Исследования характеристик ошибок в комплексах программ показали, что такие модели целесообразно ***применять как ориентиры при производстве и сопровождении крупных динамических программных продуктов*** для обеспечения их высокого качества. Модели имеют вероятностный характер, и дос-

троверность прогнозов в значительной степени зависит от точности исходных данных и глубины прогнозирования по времени. В этих условиях модели могут использоваться квалифицированными специалистами **для приближенной динамической оценки**:

- потенциально возможной надежности и безопасности функционирования комплексов программ в процессе испытаний и эксплуатации;
- числа случайных дефектов и ошибок, оставшихся не выявленными в анализируемом комплексе программ;
- длительности тестирования, требующегося для обнаружения следующей ошибки в функционирующем комплексе программ;
- времени, необходимого для выявления имеющихся ошибок и дефектов в программном продукте с заданной вероятностью.

После тестирования и отладки в течение некоторого времени интенсивность обнаружения дефектов при самых жестких внешних условиях испытаний снижается настолько, что коллектив, ведущий разработку и тестирование, попадает в зону **нечувствительности к ошибкам и возможным отказам функционирования**. У специалистов создается представление о полном отсутствии случайных проявлений дефектов, о невозможности и бесцельности их поиска, поэтому усилия на тестирование сокращаются и интенсивность обнаружения ошибок еще больше снижается. Этой предельной интенсивности обнаружения отказов соответствует наработка на обнаруженную ошибку, при которой **прекращается улучшение характеристик программного продукта** на этапах тестирования или испытаний [4, 13, 34].

Описаны несколько математических моделей, основой которых являются **различные гипотезы о характере проявления вторичных ошибок в программах** [62, 64]. Эти гипотезы в той или иной степени апробированы, при обработке данных реальных разработок. Они позволили построить модели распределения ошибок и дефектов в программах и установить связь между интенсивностью обнаружения вторичных ошибок при тестировании, интенсивностью проявления ошибок при нормальном функционировании комплекса программ и числом первичных дефектов в зависимости от длительности тестирования. Неравномерность выбора маршрутов исполнения программы при нормальной эксплуатации, разное влияние конкретных типов ошибок в программах на проявление их при функционировании, а

также сравнительно небольшие значения числа ошибок, особенно на заключительных этапах квалификационного тестирования, приводят к тому, что флюктуации интервалов времени между обнаружением очередных ошибок могут быть весьма значительными. Модели дают удовлетворительные результаты при относительно высоких уровнях интенсивности проявления ошибок, т.е. при невысокой надежности программного продукта.

Для прогнозирования характеристик ошибок в крупных динамических комплексах программ в процессе тестирования и испытаний **квалифицированные специалисты должны уметь:**

- регистрировать характеристики и время обнаружения дефектов и ошибок;
- рассчитывать интенсивность проявления дефектов в единицу времени;
- обрабатывать и обобщать статистические данные времени обнаружения дефектов и определять коэффициенты в математической модели их прогнозирования;
- рассчитывать возможное время проявления следующего дефекта комплекса программ на основе модели прогнозирования;
- оценивать возможное число случайных дефектов и ошибок, оставшихся не выявленными в тестируемом компоненте или комплексе программ;
- рассчитывать потенциально возможную надежность и безопасность функционирования комплексов программ в процессе испытаний и эксплуатации.

В [42, 60] приведен **пример модели** на базе специальной инструментальной программы для автоматизированной оценки и мониторинга числа ошибок в комплексах и компонентах программ в зависимости от длительности тестирования. Ценность модели состоит в том, что она упрощает прогнозирование возможных ошибок в системах с использованием программных продуктов. За счет прогнозирования количества остающихся ошибок модель осуществляет мониторинг и помогает контролировать качество программных продуктов. Тестировщикам программных комплексов модель предлагает графическое представление числа обнаруженных ошибок и прогноз оставшихся ошибок. Руководителю подразделения обеспечения качества проекта программа позволяет прогнозировать длительность и

количество циклов тестирования, которые потребуются для удовлетворения требований заданного качества программного продукта.

В этом примере при анализе ошибок конкретного крупного проекта [42] было принято, что завершилась инспекция начального запрограммированного кода крупного динамического комплекса программ на соответствие рабочей проектной спецификации, в ходе которой было обнаружено 3,48 ошибок на тысячу строк кода. Наибольшее совпадение аппроксимации рэлеевской кривой распределения ошибок [5] с фактическими данными установлено для момента получения этих данных, ему соответствует значение, равное также 3,48. Значения числа ошибок 3,3 и 7,8 на тысячу строк получены при расчетах на более ранние этапы соответственно эскизного и рабочего проектирования программ. При прогнозировании в соответствии с рэлеевской кривой распределения вероятности проявления дефектов комплекса программ на следующем этапе квалификационного тестирования компонентов следовало ожидать обнаружения около 2,12 ошибок на тысячу строк исходного кода. В случае сохранения той же закономерности в момент поставки клиенту на испытания, программный продукт мог содержать менее 0,07 ошибок на тысячу строк кода. Отмечается также, что частота проявления 0,1 – 0,05 ошибок на тысячу строк кода можно считать *допустимыми для ответственных систем реального времени*.

В [40] приведены результаты исследования 20 крупных поставляемых программных продуктов, созданных в 13 различных организациях. В этих продуктах коллективы специалистов добились среднего уровня 0,06 дефекта на тысячу строк нового и измененного программного кода. В пяти проектах достигнуто 0,04 – 0,075 ошибок на тысячу строк. Таким образом, *уровень ошибок около 0,05 на тысячу строк кода* в разных публикациях считается близким к предельному для высококачественных программных продуктов.

Другим примером оценок уровня ошибок критического проекта особенно высокого качества может служить программный продукт бортовых систем Шатла [40], созданный NASA. По оценке авторов в нем содержится менее одной ошибки на 10000 строк кода. Однако стоимость программного продукта достигает 1000 \$ за строку кода, что в среднем в сто раз больше, чем для административных систем и в десять раз больше, чем для ряда критических систем реального времени.

Лекция 11

ТРЕБОВАНИЯ К ПРОФЕССИОНАЛЬНОЙ КВАЛИФИКАЦИИ РУКОВОДИТЕЛЕЙ И СПЕЦИАЛИСТОВ ПО СТАНДАРТИЗАЦИИ ПРОЕКТИРОВАНИЯ И ПРОИЗВОДСТВА КРУПНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Требования к профессиональной квалификации руководителей и специалистов по организации, освоению и применению стандартов программной инженерии

В лекции 3 изложены общие проблемы обучения квалифицированных специалистов в области программной инженерии, среди которых отмечены базовые стандарты: **SE 2004; ISO 19759 – SWEBOK** и **Р-СММ** основных методов, процессов и Программ обучения. Эти стандарты полезно использовать при формировании архитектуры и планов *образовательных стандартов вузов* в области информатики и программной инженерии. В последующих лекциях (4 –10) основное внимание сосредоточено на *целях, результатах обучения и конкретизации требований к профессиональной квалификации* руководителей и специалистов, пригодных для проектирования и производства крупных программных продуктов преимущественно реального времени. Область программной инженерии поддержана, формализована и детализирована в *целостном комплексе (около50-и) систематизированных международных стандартов*, охватывающих и регламентирующих практически все процессы жизненного цикла сложных программных продуктов. Эти стандарты обобщили обширный профессиональный опыт квалифицированных специалистов, а также технологии проектирования и производства крупных программных продуктов высокого качества в различных отраслях народного хозяйства и оборонной техники. Их полезно *знать и уметь*

практически использовать при обучении руководителей и специалистов – рис. 11.1.

Квалифицированные руководители и специалисты при выборе и применении стандартов программной инженерии должны освоить и уметь учитывать:

- выбор и применение стандартов как органическую часть производства, развития и контроля качества крупных систем;
- цели и задачи базовых стандартов методов и процессов программной инженерии, которые должны включаться в Программы обучения для профессиональной квалификации руководителей и специалистов;
- психологические проблемы освоения и применения стандартов программной инженерии, регламентирующих дисциплину деятельности квалифицированных специалистов в коллективе;
- на базе стандартов для определенного проекта выделять комплекс регламентирующих документов программной инженерии, каждый из которых имеет конкретные цели и пользователей;
- в профиле стандартов выделять и устанавливать допустимые факультативные возможности и значения параметров каждого базового стандарта и/или нормативного документа;
- экономика и качество программного продукта при применении стандартов становятся предсказуемыми и управляемыми, зависящими от ресурсов, выделяемых на их достижение;
- стандарты определяют современную культуру промышленного проектирования, производства и жизненного цикла крупных комплексов программ высокого качества.

Рис. 11.1

Однако многие отечественные ***специалисты недостаточной квалификации*** в области производства программных комплексов привыкли видеть в стандартах рутину, ***сковывающую их «творчество»***. В вузах студентов обычно обучают индивидуальному программированию небольших программ без унификации их построения, интерфейсов и документации, а также без возможности встраивания в крупные комплексы программ. Усложнение и рост размеров комплексов программ приводит к созданию крупных программистских коллективов с профессиональным разделением труда, в которых необходимо регламентирование координированной ***деятельности команды квалифицированных специалистов*** над единым проектом.

Особенно трудно осваивается и внедряется формализация стандартных промышленных процессов жизненного цикла, оценивания характеристик и документирования качества крупных программных продуктов.

Важная проблема освоения и применения стандартов программной инженерии – психологическая. При этом происходит качественный психологический **перелом методов и стиля работы** специалистов от индивидуальной деятельности и относительной свободы выбора стиля и времени завершения задания к стилю работы в коллективе с профессиональным разделением труда и взаимодействием с другими специалистами по интерфейсам компонентов и времени (графику) готовности компонентов к поставке. Стандарты **регламентируют определенную дисциплину трудового коллектива** квалифицированных специалистов при создании и применении крупных программных продуктов требуемого качества при ограниченных ресурсах. Для этого необходимо изучение и освоение соответствующих стандартов, а также корректное **применение их требований** и рекомендаций **всеми** участниками крупных проектов. Это должно обеспечивать повышение экономической эффективности, качества и конкурентоспособности жизненного цикла крупных программных продуктов. Жизненный цикл крупных программных комплексов отражается в стандартах набором процессов, этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующим ведение проектирования, производства, сопровождения и эксплуатации от анализа и подготовки требований до завершения испытаний ряда версий программного продукта и прекращения их использования. Накопленный капитал международных стандартов программной инженерии в значительной степени **обеспечивает современный, регламентированный жизненный цикл комплексов программ** и продолжает успешно развиваться и совершенствоваться (см. рис. 11.1).

Это требует от коллективов разработчиков программных продуктов **высокой квалификации в области стандартизации** и дополнительных усилий для освоения и мониторинга номенклатуры, текущего состояния и адаптации стандартов к свойствам и характеристикам конкретных продуктов. Практическое применение международных стандартов, сосредоточивших мировой опыт создания различных типов крупных комплексов программ, способствует значи-

тельному снижению стоимости, трудоемкости, длительности и улучшению других экономических показателей проектов крупных программных продуктов, **повышению квалификации и производительности труда** специалистов и **качества** создаваемых продуктов. Обещания руководителей производства в контрактах с заказчиками создать высококачественные продукты в согласованные сроки во многих случаях без применения стандартов не выполняются как вследствие различий в понимании требуемого качества, так и вследствие неумения оценивать ресурсы, необходимые для достижения высокого качества крупных комплексов программ. В результате качество программной продукции зачастую остается низким, неподдающимся достоверной оценке и не конкурентоспособным на международном рынке.

Стандарты программной инженерии включают описания исходной информации, способов и методов выполнения операций и работ, устанавливают требования к результатам и правилам их контроля, а также определяют содержание технологических и эксплуатационных документов. Они рекомендуют организационную **структуру коллективов специалистов**, регламентируют распределение и планирование работ, а также контроль за ходом разработки. Эти стандарты, в частности, определяют модификацию, мобильность и возможность повторного применения программных компонентов и комплексов, их расширяемость и переносимость на различные аппаратные и операционные платформы, что непосредственно отражается на росте экономической эффективности технологий и процессов создания различных программных продуктов.

Детализация требований и положений стандартов программного продукта определенного функционального назначения должна проводиться квалифицированными специалистами с ориентацией на унификацию конкретных процессов, работ и документов. Можно выделить следующие основные группы квалифицированных специалистов, **обязанных знать и использовать стандарты**:

- руководители крупных проектов системы и основных, функциональных компонентов программного продукта;
- системные аналитики, создатели спецификаций требований, пилотных проектов компонентов и алгоритмов решения функциональных задач;

- программисты-разработчики программных компонентов, структур и содержания данных;
- интеграторы функциональных программных компонентов, тестирующие и отлаживающие крупные функциональные компоненты или комплекс программ в целом;
- специалисты сопровождения и управления конфигурацией версий программных продуктов;
- испытатели и сертификаторы программных продуктов;
- разработчики технологий, инструментальных средств, методических, руководящих и инструктивных документов, обеспечивающих реализацию утвержденных стандартов конкретного проекта программного продукта.

Для эффективной деятельности квалифицированных специалистов на базе стандартов должен создаваться для определенного проекта **комплект (профиль) регламентирующих документов программной инженерии**, каждый из которых имеет конкретных пользователей в жизненном цикле. В них **должно быть отражено**:

- содержание и описание выбранных положений и разделов стандартов и нормативных документов с позиции их конкретного пользователя;
- параметры адаптации разделов стандартов и содержание дополнительных нормативных документов;
- методика и сценарии корректного применения всех обязательных и рекомендуемых положений стандартов;
- требования к содержанию отчетов о результатах контроля и тестирования компонентов системы на соответствие обязательным положениям стандартов в процессе их жизненного цикла.

В зависимости от характеристик и особенностей конкретного проекта специалистами должен проводиться целеустремленный отбор необходимых унифицированных требований и процессов, а также формирование на их базе проблемно-ориентированных **профилей стандартов** для определенных типов проектов и/или предприятий. Для регламентирования процессов жизненного цикла стандарты рекомендуется **адаптировать** и конкретизировать. При этом должна сохраняться концептуальная целостность применяемой совокупности стандартов и их эффективное, положительное влияние на процессы и результаты, на качество, надежность и безопасность программных продуктов при реальных ограничениях на использование доступных

ресурсов проектов [28, 53]. Комплексное, скоординированное применение стандартов квалифицированными специалистами в процессе создания, развития и применения программных продуктов позволяет исключать многие виды дефектов или значительно ослаблять их влияние. Качество продукта **становится предсказуемым и управляемым**, непосредственно зависящим от ресурсов, выделяемых на его достижение, а главное, от системы качества и эффективности технологий, используемых на всех этапах жизненного цикла.

Применение стандартов позволяет ориентироваться квалифицированным специалистам на построение систем и комплексов программ из крупных функциональных узлов, отвечающих требованиям стандартов, применять отработанные и проверенные проектные решения. Они определяют структуру и унифицированные **интерфейсы взаимодействия** компонентов таким образом, что разработчику системы, как правило, не требуется вдаваться в детали внутреннего устройства этих компонентов. Стандарты, относящиеся к программным комплексам (функциональным частям) систем, облегчают повторное использование в новых системах готовых и апробированных программных продуктов. Для унификации и регламентирования производственных процессов такие совокупности – **профили стандартов** должны адаптироваться и конкретизироваться специалистами применительно к определенным классам проектов, процессов и компонентов программных продуктов. Таким образом, производство программного продукта в значительной степени может сводиться к их интеграции и комплексированию из стандартизованных компонентов.

Методы и процессы, регламентированные стандартами комплексов программ, играют **стабилизирующую и организующую роль** во всем их жизненном цикле и обеспечивают:

- расширение и совершенствование функций систем и компонентов с сохранением их целостности и первичных затрат;
- систематическое повышение качества функционирования комплексов программ и баз данных для решения задач пользователей в различной изменяющейся внешней среде;
- улучшение экономических характеристик применения систем и программных продуктов;
- совершенствование технологий обеспечения жизненного цикла сложных систем и комплексов программ.

Для этого при создании и сопровождении сложных, распределенных систем, формировании их архитектуры, выборе программных компонентов и их связей стандартами рекомендуется учитывать ряд современных концептуальных *требований формирования их жизненного цикла* [20, 31, 51].

- архитектура комплекса программ должна соответствовать текущим и перспективным целям и стратегическим, функциональным задачам создаваемой системы, быть достаточно гибкой и допускать относительно простое, без коренных структурных изменений развитие и наращивание функций и ресурсов системы в соответствии с расширением сфер и задач ее применения;
- в структуре и компонентах комплекса программ и системы следует предусматривать обеспечение максимально возможной сохранности инвестиций в аппаратные и программные средства, а также в базы данных при длительном развитии, сопровождении и модернизации системы;
- необходимо обеспечивать эффективное использование ресурсов системы и минимизировать интегральные затраты на обработку данных в типовых режимах ее функционирования с учетом эксплуатационных затрат и капитальных вложений в создание системы и программного продукта;
- должны быть обеспечены безопасность функционирования системы и надежная защита данных от ошибок, от разрушения или потери информации, а также авторизация пользователей, управление рабочей загрузкой, резервированием и оперативным восстановлением функционирования системы и программного продукта;
- для обеспечения перспективы развития жизненного цикла системы и комплекса программ целесообразно предусматривать возможность интеграции гетерогенных вычислительных компонентов и возможность их переноса на различные аппаратные и операционные платформы на основе концепции и стандартов открытых систем;
- следует обеспечивать комфортное обучение и максимально упрощенный доступ конечных пользователей к управлению и результатам функционирования системы и программного продукта на основе современных графических средств и наглядных пользовательских интерфейсов.

Для регламентирования жизненного цикла сложных комплексов программ целесообразно выбирать, комплексировать, адаптировать и

применять конкретные *группы стандартов – профили*, которые определяют процессы и продукты при их проектировании, производстве и сопровождении. *Профиль стандартов* – это совокупность нескольких базовых стандартов (и других нормативных документов) с четко определенными и гармонизированными подмножествами обязательных и факультативных *возможностей и требований*, предназначенная для реализации заданной функции или группы функций. В профиле выделяются и устанавливаются допустимые факультативные возможности и значения параметров каждого базового стандарта и/или нормативного документа, входящего в профиль. Он должен использовать факультативные возможности, требования и значения параметров в пределах допустимых, выбранных из альтернативных вариантов в стандартах. На базе одной и той же совокупности положений базовых стандартов могут формироваться и утверждаться различные профили для разных проектов систем и сфер применения [6, 16, 31, 52].

Профиль стандартов конкретной системы должен учитывать ее функциональную ориентацию. Он должен содержать ссылки на стандартизованные интерфейсы между комплексом программ и внешней средой, которые описываются в профилях среды системы. Особенности организационных структур, различия в размерах и сложности проектов, в требованиях к системам и применяемым методам их разработки, необходимость преемственности с системами, находящимися в эксплуатации, влияют на организацию разработки, приобретения, применения и сопровождения программных продуктов. Каждый из выделенных профилей должен для последующего длительного использования пройти стадию формирования, адаптации и параметризации применительно к характеристикам стандартизируемых объектов или процессов. Для корректного применения *описания профилей должны содержать*:

- определение целей, которые предполагается достичь применением данного профиля стандартов;
- перечисление функций продукта или процесса стандартизации, определяемого данным профилем;
- формализованные сценарии применения базовых стандартов и спецификаций, включенных в данный профиль;

- сводку требований к системе или к ее компонентам, определяющих их соответствие профилю и требований к методам тестирования соответствия;
- ссылки на конкретный набор стандартов и других нормативных документов, составляющих профиль, с точным указанием используемых положений, редакций и ограничений, способных оказать влияние на достижение корректного взаимодействия объектов стандартизации при использовании данного профиля;
- информационные ссылки на спецификации тестов проверки соответствия профилю.

Профиль стандартов конкретной системы **не является статичным, он развивается и конкретизируется** в процессе жизненного цикла и оформляется в составе документации системы. Разработка и применение профилей стандартов являются **органической частью процессов производства и развития крупных систем**. Проектированию системы предшествует обследование объекта автоматизации, результатом которой являются его функциональная и информационная модели, определение целей создания системы и состава ее функций. Стандарты, важные с точки зрения заказчика, должны задаваться в спецификации требований на проектирование системы и составлять ее первичный профиль.

Наиболее актуальна **стандартизация процессов жизненного цикла** комплексов программ при коллективном производстве и сопровождении крупных **критических систем управления в реальном времени**, к которым предъявляются высокие требования к качеству функционирования. В этих случаях квалифицированным руководителям особенно необходимо четкое планирование и управление технологическими процессами. Комплекс международных стандартов ориентирован на программные продукты, выполняющие важные функции в системах управления объектами, технологическими процессами или при обработке ответственной информации. Применение таких стандартов полностью при создании и использовании простых программ узкого или экспериментального назначения не всегда может быть оправдано. Однако они определяют **современную культуру промышленного производства и стандартизации жизненного цикла комплексов программ высокого качества**.

Требования к составу и содержанию системы стандартов для производства крупных программных продуктов

Модель жизненного цикла сложных программных комплексов обычно формируется из **10 – 12 базовых процессов – этапов**. Их количество зависит от целей, сложности и особенностей проекта, от назначения и области применения модели, а также от возможностей формализации ее компонентов. При формировании модели **требований к квалификации руководителей и специалистов по стандартизации** учитывалось наличие международных стандартов, которые могли использоваться при определении жизненного цикла и **объединении в профиль**, пригодный для последующего использования в технологии создания и развития программного продукта. Поэтому ряд не стандартизованных – творческих процессов явно не отражен в такой модели, однако они существенны для регламентирования реального проектирования и производства программного продукта. Сформированные и используемые для применения международные стандарты производства крупных программных продуктов состоят из **трех групп**:

- **первая группа** – управления жизненным циклом сложных проектов систем и программных комплексов, возглавляемая стандартами менеджмента – **CMMI** и **ISO 9000** – рис. 11.2;
- **вторая группа** – проектирования, производства, сопровождения и управления конфигурацией, регламентируется базовыми стандартами жизненного цикла систем и программных средств – **ISO 15288** и **ISO 12207** – рис.11.3;
- **третья группа** – оценивания и обеспечения качества, безопасности и документирования в жизненном цикле программных средств, начинается головными стандартами – **ISO 9126** и **ISO 25000** – рис. 11.4.

На рис. 11.2 – 11.4 в каждой группе стандартов в скобках указано ориентированное число действующих международных стандартов, содержание которых комментируется в [31].

Эта схема далее рекомендуется как базовая, подлежащая конкретизации и адаптации в соответствии с особенностями развития профиля жизненного цикла конкретного программного комплекса. Большинство представленных стандартов имеют адекватные переводы на русский язык.

Квалифицированные руководители и специалисты должны уметь выбирать, адаптировать и применять основные стандарты административного управления жизненным циклом и качеством комплексов программ:

- интегрированную модель оценивания зрелости продуктов и процессов разработки программных комплексов СММІ (1 станд.);
- систему стандартов административного управления качеством продукции (8 станд.);
- руководство по административному управлению качеством комплексов программ (1 станд.);
- стандарты интерфейсов переносимых открытых систем (6 станд.).

Рис. 11.2

Квалифицированные руководители и специалисты должны уметь выбирать и применять стандарты обеспечения качества в жизненном цикле комплексов программ:

- стандарты классификации и выбора характеристик качества программных продуктов (6 станд.);
- стандарты оценивания и измерения характеристик качества программных продуктов (5 станд.);
- стандарты обеспечения функциональной и информационной безопасности комплексов программ (5 станд.);
- стандарты документирования комплексов программ (5 станд.);
- стандарты сертификации программных продуктов (1 станд.).

Рис. 11.3

Квалифицированные руководители и специалисты должны уметь выбирать и применять стандарты обеспечения качества в жизненном цикле комплексов программ:

- стандарты классификации и выбора характеристик качества программных продуктов (6 станд.);
- стандарты оценивания и измерения характеристик качества программных продуктов (5 станд.);
- стандарты обеспечения функциональной и информационной безопасности комплексов программ (5 станд.);
- стандарты документирования комплексов программ (5 станд.);
- стандарты сертификации программных продуктов (1 станд.).

Рис. 11.4

Значительные достижения в организации, планировании, развитии и применении современных методов и технологий обеспечения **крупных проектов комплексов программ** сосредоточены в методологии и **базовом стандарте де-факто СММ/СММІ** (Capability Maturity Model – *система и модель для оценки зрелости*) технологических процессов жизненного цикла комплексов программ (рис. 11.5).

Она основана на формализации и использовании **уровней зрелости** технологической поддержки ЖЦ, которые определяют потенциально возможное качество создаваемых на предприятии комплексов программ. Чем выше уровень зрелости, тем выше статус предприятия среди поставщиков, доверие к его продукции, его конкурентоспособность, а также возможное качество программного продукта.

Тем самым при выборе **требований к характеристикам качества** программного продукта можно в соответствующей степени доверять поставщику и предприятию разработчика, что они смогут полностью **реализовать требования заказчика**. Эти уровни зрелости характеризуются степенью формализации, адекватностью измерения и документирования процессов и продуктов, полнотой применения стандартов и инструментальных средств автоматизации работ, наличием системы качества технологических процессов [65].

Назначение методологии СММ/СММІ – оценки зрелости – состоит в предоставлении необходимых общих рекомендаций и инструкций предприятиям, производящим программный продукт, по выбору стратегии совершенствования качества, процессов и продуктов путем анализа степени их производственной зрелости и оценивания факторов, в наибольшей степени влияющих на качество, а также посредством выделения процессов, требующих модернизации. В методологии СММІ **выделены пять уровней зрелости**, раскрываемые в стандарте (см. рис.11.5).

Виды деятельности для высоких уровней зрелости в соответствии с СММІ в стандарте делятся на базовые и общие. Базовые виды деятельности являются обязательными и сгруппированы в пять категорий: контрактная; инженерная; управленаческая; вспомогательная; организационная.

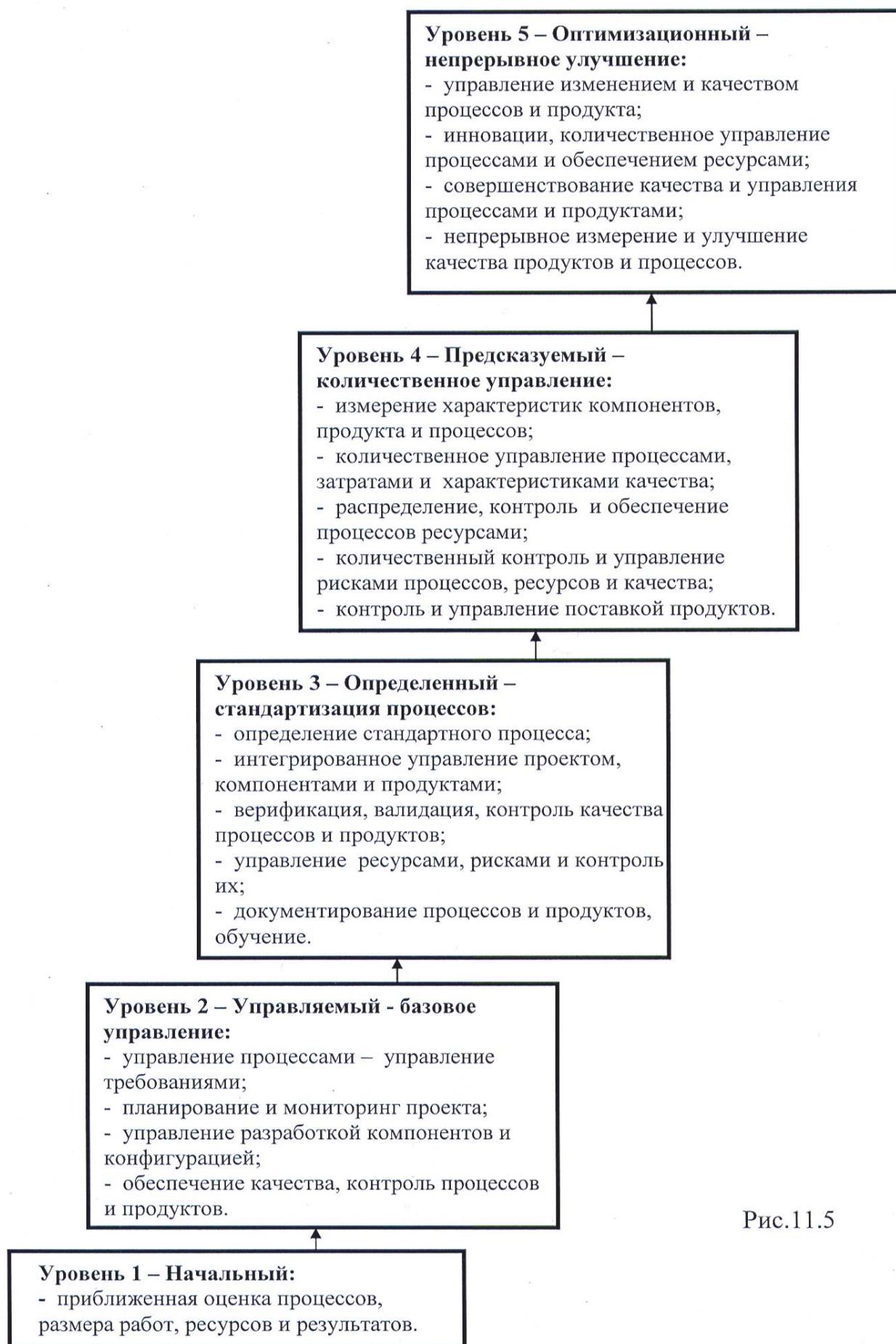


Рис.11.5

Эти *уровни зрелости характеризуются* степенью формализации, адекватностью измерения и документирования процессов и продуктов, широтой применения стандартов и инструментальных средств автоматизации работ, наличием и полнотой реализации функций системой обеспечения качества технологических процессов и их результатов.

Описание процессов производства в СММІ сфокусировано на поэтапном определении реально достижимых результатов и на оценивании качества их выполнения. Качество процессов зависит от технологической среды, в которой они выполняются. *Зрелость процессов* – это степень их управляемости, возможность поэтапной количественной оценки качества, контролируемость и эффективность результатов. Модель зрелости предприятия представляет собой методический нормативный материал, определяющий правила создания и функционирования системы управления жизненным циклом комплексов программ, *методы и стандарты систематического повышения культуры и качества производства*. На предприятиях, достигших высокого уровня зрелости, формализованные процессы должны принимать статус стандарта, фиксироваться квалифицированными руководителями в организационных структурах и определять производственную тактику и стратегию корпоративной культуры производства и системы обеспечения качества. Она содержит процедуры – четыре категории СММІ процессов: менеджмент процессов; менеджмент проекта; инжиниринг (технология); поддержка и использование СММІ модели. Последний раздел самый большой в стандарте и занимает около 500 страниц из полного объема документа до 700 страниц. В этом разделе представлены подробные рекомендации и требования для реализации каждого из перечисленных выше четырех уровней процессов, которые учитывают особенности конкретных моделей, *полезных для квалифицированных руководителей и специалистов*.

Рекомендуется на каждом более высоком уровне зрелости применять *все процессы* предыдущих нижних уровней (см. рис.11.5). Упорядочение и оценка используемых процессов в соответствии с уровнями *позволяет устанавливать производственный потенциал предприятий* – разработчиков программных продуктов по прогнозируемому качеству результатов их деятельности и возможности сертификации. Это уменьшает зависимость заказчиков и пользователей

от возможных недостатков исполнителей проектов и позволяет их выбирать с учетом прогнозируемого качества продуктов.

Стандарты в первой группе (см. рис.11.2) **CMMI** и **ISO 9001:2000** во многом подобны по структуре и содержанию детальных требований к организации и планированию проектирования и производства программного продукта. При практической реализации и обеспечении *всего жизненного цикла сложных комплексов программ* разработчикам целесообразно использовать полный профиль стандартов, а для *оценивания заказчиками уровня* руководства, организационного и технологического обеспечения проектов применять конкретные рекомендации **CMMI** или **ISO 9001:2000**. Эти рекомендации могут эффективно использоваться при *сертификации качества процессов производства* на предприятиях по комплексу стандартов менеджмента **ISO 9000** в зависимости от особенностей проекта и требований заявителя на сертификацию программного продукта и/или технологии обеспечения его жизненного цикла.

Стандарты во второй группе (см. рис. 11.3) **ISO 15288** и **ISO 12207** являются базовыми, на основе которых сформировался и развивается *профиль стандартов обеспечения жизненного цикла сложных программных средств*. Компоненты стандарта **ISO 12207** и архитектура некоторых разделов непосредственно использованы при детализации структуры нескольких важных стандартов: управления проектами комплексов программ – **ISO 16326**; сопровождения – **ISO 14764**; управления конфигурацией – **ISO 15846**. В ряде стандартов приводятся перекрестные ссылки, и рекомендуется использовать отдельные фундаментальные положения стандарта **ISO 12207**.

Документация является органической, составной *частью стандартизации производства крупного программного продукта* и требуются значительные ресурсы и специалисты для ее создания и применения (см. рис. 11.4). Тексты и объектный код комплексов программ могут стать программным продуктом только в совокупности с документами, полностью отражающими их содержание и достаточными для его освоения, применения и изменения. Для этого документы должны быть корректными, строго адекватными текстам программ и содержанию баз данных – систематически, структурировано и понятно изложены для возможности их успешного освоения и использования достаточно квалифицированными специалистами различных рангов и назначения. Качество и полнота отображения в до-

кументах процессов и продуктов в жизненном цикле программных комплексов, должны полностью определять достоверность информации для взаимодействия заказчиков, пользователей и разработчиков, а тем самым, корректность функций и достигаемое качество программных продуктов и соответствующих систем. Посредством документов (электронных или бумажных) специалисты взаимодействуют с программными средствами и данными в реализующих их вычислительных машинах, а также между собой.

Необходимо обеспечивать возможность достоверного, формально точного общения всех участников производства между собой, с создаваемым продуктом и с документами для гарантии поступательного развития и совершенствования комплекса программ. Адекватность документации требованиям, состоянию текстов и объектных кодов программ должна инспектироваться и удостоверяться (подписываться) ответственными руководителями и заказчиками проекта. ***Ошибки и дефекты документов не менее опасны для применения и изменения программного продукта***, чем ошибки в структуре, интерфейсах, файлах текстов программ и в содержании данных. Поэтому к разработке, полноте, корректности и качеству документации необходимо столь же тщательное отношение, как к производству и изменениям текстов программ и данных [32, 37, 67].

Реализация корректных документов в значительной степени определяет достигаемое качество сложных программных продуктов, трудоемкость и длительность их создания. Для этого должна формироваться и использоваться регламентированная стратегия, стандарты, распределение ресурсов и планы создания, изменения и применения документов на программы и данные сложных систем. В общем случае должны быть ***выделены квалифицированные руководители и коллектив специалистов***, которые способны планировать, описывать, утверждать, выпускать, распространять и сопровождать комплекты адекватных документов. Они должны стимулировать разработчиков программ осуществлять непрерывное, полноценное документирование процессов и результатов своей деятельности, а также контролировать полноту и качество исходных, результирующих и отчетных документов. Официальная, описанная и утвержденная стратегия документирования ***должна устанавливать дисциплину***, необходимую для эффективного создания высококачественных документов на продукты и процессы в жизненном цикле комплексов программ [18, 21].

Для создания и реализации положений этих документов должны быть выбраны и применяться инструментальные средства, совместно образующие взаимосвязанный комплекс технологической поддержки и автоматизации ЖЦ и не противоречащие предварительно скомпонованному набору нормативных документов. Организация и реализация *работ по созданию документов должны распределяться между квалифицированными специалистами*, ведущими непосредственное и преимущественное создание комплексов программ, и специалистами, осуществляющими, в основном, разработку, контроль и издание интегральных документов. При создании особо сложных систем целесообразно выделение специального коллектива, обеспечивающего организацию и реализацию основных системных работ по документообороту на комплекс программ.

Особое внимание в последнее время уделяется *совершенствованию и детализации стандартов документирования, обеспечивающих высокое качество создаваемых программных продуктов*, а также возможности их эффективного итерационного развития длительное время в многочисленных версиях. Соответственно должны изменяться документы, отражающие состояние производственных процессов и компонентов программных проектов. Для этого организация процессов документирования должна обеспечивать возможность гибкого и точного изменения документов – сопровождения и конфигурационного управления версиями и редакциями каждого документа (см. лекцию 7). Эти процессы и поддерживающие их средства автоматизации должны быть адекватными тем, которые применяются при сопровождении непосредственных продуктов производства – программ и данных. Они должны быть поддержаны организацией контроля, регистрации и утверждения версии каждого документа в той степени и на том уровне, которые необходимы в данном проекте для обеспечения корректности определенного документа.

Технологическая документация, непосредственно и в наибольшей степени должна отражать производственные процессы жизненного цикла комплексов программ и данных и требования к ним. Стандарты и нормативные документы, входящие в жизненный цикл проекта, должны регламентировать структуру, состав производственных этапов, работ и документов. Они должны формализовать выполнение и документирование конкретных работ при проектировании, разработке и сопровождении; обеспечивать адаптацию документов к

характеристикам среды разработки, внешней и операционной системы; регламентировать процессы обеспечения качества продукта и его компонентов, методы и средства их достижения, реальные значения достигнутых показателей качества. Для контроля возможных изменений специалистам целесообразно предусматривать и согласовывать с заказчиком специальный документ, регламентирующий правила применения и корректировки их номенклатуры, а также состава и содержания документации, поддерживающей производство комплексов программ.

В *технологической документации*, обеспечивающей конфигурационное управление и длительное сопровождение версий программного продукта, необходимы требования, тесты, тексты программ и описания алгоритмов. Это приводит к увеличению объема документации на порядок, т.е. может составлять около 100 страниц совокупности документов на тысячу строк программы. Статистический анализ объема документации для программных продуктов различных классов дал широкий разброс числа страниц на единицу объема программ. Однако выявились некоторые средние характеристики, которые близки к указанным выше оценкам. Подтверждена наиболее высокая документированность тиражируемых программ реального времени, особенно в тех случаях, когда необходима высокая безопасность и надежность функционирования продукта (до 200 страниц на тысячу строк).

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

ЛИТЕРАТУРА

1. Армстронг М. Стратегическое управление человеческими ресурсами. Пер. с англ. – М.: ИНФРА-М. 2002.
2. Архипенков С. Марш победителей. Адаптивное управление проектом: принципы и примеры. – М.: Материалы конференции. SEC(R) 2008.
3. Бейзер Б. Тестирование черного ящика. Технология функционального тестирования программного обеспечения и систем. Пер. с англ. – СПб: ПИТЕР. 2004.
4. Блэк Р. Ключевые процессы тестирования. Пер. с англ. – М: ЛОРИ. 2006.
5. Боэм Б.У. Инженерное проектирование программного обеспечения. Пер. с англ./Под ред. А.А. Красилова. – М.: Радио и связь. 1985. (Barry W. Boehm. Software Engineering Economics. Prentice-Hall. 1981).
6. Брауде Э. Д. Технология разработки программного обеспечения. Пер. с англ. – М.: ПИТЕР. 2004.
7. Буланже А. Надежность и безопасность: открытый код против закрытого. – М.: Открытые системы. № 12. 2005.
8. Воас Д. Чертова дюжина проблем программной инженерии. – М.: Открытые системы. № 7. 2007.
9. Вигерс К.И. Разработка требований к программному обеспечению. Пер. с англ. – М.: Русская редакция. 2004.
10. Волков О.И., Скляренко В.К. Экономика предприятия. – М.: ИНФРА-М. 2007.
11. Гецци К., Джазайери М., Мандриоли Д. Основы инженерии программного обеспечения. Пер. с англ. – СПб.: БХВ-Петербург. 2005.
12. Гринфилд Д., Шорт К. Фабрика разработки программ. Пер. с англ. – М: Диалектика. 2007.
13. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. Пер. с англ. – М.: ЛОРИ. 2003.
14. Дмитриева М.А., Крылов А.А. Психология труда и инженерная психология. – М.: Дельфа. 2005.

15. Иванников В.П. Свободное программное обеспечение. Проблемы и перспективы. – М.: Материалы конференции. SEC(R). 2008.
16. Ильин А.И. Планирование на предприятии. Учебное пособие: В 2 ч. Ч. 1. Стратегическое планирование. – Минск: Новое знание, 2000.
17. Калбертсон Р., Браун К., Кобб Г. Быстрое тестирование. Пер. с англ. – М.: Вильямс. 2002.
18. Калянов Г.Н. Теория и практика реорганизации бизнес-процессов. – М.: СИНТЕГ. 2000.
19. Канер С., Фолк Д., Нгуен Е. Тестирование программного обеспечения. Пер. с англ. – М: ДиаСофт. 2001.
20. Кантор М. Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения. Пер. с англ. – М.: Вильямс. 2002.
21. Костогрызов А.И., Липаев В.В. Сертификация качества функционирования автоматизированных информационных систем. М.: Изд. Вооружение. Политика. Конверсия. 1996.
22. Костогрызов А.И., Степанов П.В. Инновационное управление качеством и рисками в жизненном цикле систем. – М.: Изд-во ВПК. 2008.
23. Коберн А. Современные методы описания требований к системам. Пер. с англ. – М.: Лори. 2002.
24. Кузнецов С.Д. Базы данных. Модели и языки. Учебник. – М.: БИНОМ. 2008.
25. Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. Пер. с англ. – М.: Вильямс. 2002.
26. Липаев В.В., Потапов А.И. Оценка затрат на разработку программных средств. – М.: Финансы и статистика. 1988.
27. Липаев В.В. Отладка сложных программ. М.: Энергоатомиздат. 1993.
28. Липаев В.В. Методы обеспечение качества крупномасштабных программных средств. – М.: РФФИ. СИНТЕГ. 2003.
29. Липаев В.В. Технико-экономическое обоснование проектов сложных программных средств. – М.: СИНТЕГ. 2004.
30. Липаев В.В. Функциональная безопасность программных средств. – М.: СИНТЕГ. 2004.
31. Липаев В.В. Процессы и стандарты жизненного цикла сложных программных средств. Справочник. – М.: СИНТЕГ. 2006.

32. Липаев В.В. Документирование сложных программных средств. – М.: СИНТЕГ. 2005.
33. Липаев В.В. Программная инженерия. Методологические основы. Учебник. – М.: ТЕИС. 2006.
34. Липаев В.В. Тестирование крупных комплексов программ на соответствие требованиям. – М.: ГЛОБУС. 2008.
35. Липаев В.В. Экономика производства сложных программных продуктов. – М.: СИНТЕГ. 2008.
36. Непейвода Н. Какая математика нужна информатикам? Открытые системы. №9. 2005.
37. Организация производства и управления предприятием. Учебник. Под ред. О.Г. Туровца. – М.: ИНФРА-М. 2006.
38. Петренко А.К. и др. Разработка и внедрение технологии автоматизированного тестирования программного обеспечения на основе формальных спецификаций. – М: ИСП РАН. 2007.
39. Рекомендации по преподаванию программной инженерии и информатики в университетах. Пер. с англ. – М.: Интuit. 2007.
40. Риган П., Хемилтон С. NASA: Миссия надежна. – М.: Открытые системы. № 3. 2004.
41. Смольянинов А., Ложечкин А. Некоторые секреты командной разработки. – М.: Открытые системы. № 7-8. 2005.
42. Соммервилл И. Инженерия программного обеспечения. 6-е издание. Пер. с англ. – М.: Вильямс. 2002.
43. Стрелков Ю.К. Инженерная и профессиональная психология. – М.: Академия. 2001.
44. Тельнов Ю.Ф. Интеллектуальные информационные системы в экономике. Учебное пособие. – М.: СИНТЕГ. 2002.
45. Терехов А. Экономика подготовки кадров на предприятиях. – М.: Материалы конференции. SEC(R). 2008.
46. Торингтон Д., Холл Л., Темлер С. Управление человеческими ресурсами. Учебник. Пер. с англ. – М.: Дело и сервис. 2004.
47. Трахтенгерц Э.А. Субъективность в компьютерной поддержке управлеченческих решений. – М.: СИНТЕГ. 2001.
48. Трубачев А.П., Долинин М.Ю., Кобзарь М.Т. и др. Оценка безопасности информационных технологий. Общие критерии. Пер. с англ. Под ред. В.А. Галатенко. – М.: СИП РИА, 2001.
49. Тэллес М., Хсих Ю. Наука отладки. Пер. с англ. – М.: Кудиц-образ. 2003.

50. Уайт Б.А. Управление конфигурацией программных средств. Практическое руководство по Rational ClearCase. Пер. с англ. – М.: ДМК Пресс. 2002.
51. Фатрелл Р.Т., Шафер Д.Ф., Шафер Л.И. Управление программными проектами: достижение оптимального качества при минимальных затратах. Пер. с англ. – М.: Вильямс. 2003.
52. Щербо В.К., Козлов В.А. Функциональные стандарты в открытых системах. Ч. 1. Концепция открытых систем. Ч.2. Международные функциональные стандарты. – М.: Изд. МЦНТИ. 1997.
53. Экономика промышленного предприятия. Учебник. Под ред. Е.Л. Кантора, Г.А.Маховиковой. – М.: МарТ. 2007.
54. Boehm B.W. et al. Software cost estimation with COCOMO II. Prentice Hall PTR. New Jersey. 2000.
55. Boehm B.W. Software risk management. IEEE Computer Society Press. Washington. 1989.
56. Charett R. Software engineering risk analysis and management. N.Y.: McGraw – Hill. 1989.
57. Davis A. Software requirements: Objects, functions and states. – Englewood Cliffs. NY. Prentice-Hall. 1993.
58. Grady R. Practical software metrics for project management and process improvement. – Englewood Cliffs. NY. Prentice-Hall. 1992.
59. Jones C. Applied software measurement, assuring productivity and quality. McGraw-Hill. NY. 1996.
60. Howden W.E. Functional program testing and analysis. N.Y.: McGraw - Hill, 1987.
61. Karolak D.W. Software engineering risk management. IEEE Computer Society Press. Washington. 1996.
62. Littlewood B. ed. Software Reliability – Achievement and Assessment. London. Blackwell Scientific Publications. 1987.
63. Martin J., McClure C. Software maintenance, the problems and its solutions. - N.Y.: Prentice-Hall.1983.
64. Musa J.D., Iannino A., Okumoto K. Software Reliability: Measurement, Prediction, Application. N.Y. McGraw Hill. 1987.
65. Mutafelija B., Stromberg H. Systematic process improvement using ISO 9001:2000 and CMMI. SEI. 2003.
66. Perry W.E. Effective Methods for Software Testing. NY. Wiley. 2000.
67. Schindler M.J. Computer – aided software design. Build quality software with CASE. - N.Y. John Wiley & Sons, 1990.

68. Shooman M.L. Software Engineering: Reliability, Development and Management. N.Y. McGraw-Hill. 1983.
69. Yourdon E. Modern Structured Analysis. N.J. Yourdon Press/Prentice-Hall. 1995.