

ТЕСТИРОВАНИЕ КРУПНЫХ КОМПЛЕКСОВ ПРОГРАММ НА СООТВЕТСТВИЕ ТРЕБОВАНИЯМ

В.В. Лунаев,

главный научный сотрудник Института системного программирования РАН, профессор Государственного университета — Высшей школы экономики
algordom@mtu-net.ru

Рассмотрены проблемы разработки корректных требований к крупным системам и комплексам программ; тестирование и реализация требований к крупным программным продуктам.

Концепция тестирования программных продуктов на соответствие требованиям

Крупные комплексы программ — компоненты вычислительных систем, реализующие обычно их основные функциональные свойства и задачи, и создающие предпосылки для последующего их развития и применения. Реализация жизненного цикла, методология управления, обработки информации и изменения программных комплексов зависят от квалификации специалистов, технических, организационных и договорных требований заказчика и от сложности проекта. Множество текущих состояний и модификаций компонентов в жизненном цикле крупных комплексов программ, обуславливают необходимость упорядочивать, контролировать их развитие и реализацию участниками проекта. Организованное, контролируемое и методичное отслеживание создания и динамики изменений в жизненном цикле комплексов программ, их разработка при строгом учёте и контроле каждого изменения — базовая проблема эффективного, поступательного развития крупных систем высокого качества *методами программной инженерии* [7, 8, 10]. Накопление в мире знаний, опыта разработки и применения огромного количества различных сложных комплексов и компонентов программ для ЭВМ способствует систематизации и обобщению методов и технологий их разработки, сокращению дефектов и неопределённостей в характеристиках и качестве поставляемых и применяемых программных продуктов.

Методология программной инженерии основана на применении систематизированного, научного и предсказуемого процесса планирования, проектирования, разработки, тестирования и сопровождения крупных программных продуктов. Разработчики таких продуктов должны учитывать, что они работают в организационных и финансовых рамках заключённых контрактов, и искать решение поставленной перед ними задачи с учётом требований и условий заказчика. Программная инженерия *не рассматривает технические аспекты индивидуального создания (программирования) модулей и небольших компонентов*, в её ведении задачи коллективов специалистов по управлению проектами и разработке методов, теорий и средств, необходимых для обеспечения жизненного цикла крупных комплексов программ.

Особенности тестирования крупных программных комплексов на соответствие требованиям заказчика:

- ✧ объектами тестирования должны быть крупные, целостные комплексы программ (сотни тысяч строк) для систем обработки информации и управления в реальном времени, разрабатываемые большими коллективами (командами) специалистов;
- ✧ программные продукты должны соответствовать *первому эталону* — целостному комплексу требований к функциям, характеристикам, архитектуре и качеству, утверждённому заказчиком и согласованному с разработчиком;
- ✧ комплексы применяемых тестов рассматриваются как *второй эталон* функций и характеристик программного средства, которые должны

адекватно отражать и покрывать набор требований заказчика к программному продукту;

- ✧ разработка требований и тестирование комплексов программ осуществляется методами программной инженерии в условиях *регулярного планирования, управления и ограниченных ресурсов*, выделяемых заказчиком;
- ✧ применяются *сценарии и процедуры* детерминированных тестов и *генераторы динамических, стохастических тестов* на основе аттестованных, сложных моделей внешней среды и систем реального времени;
- ✧ процессы и продукты формирования требований и реализации тестирования должны быть охвачены глубоким, детальным *документированием, архивированием и конфигурационным управлением* исходными, промежуточными и отчётными документами;
- ✧ в крупных программных комплексах применяются *апробированные программные компоненты и модули высокого качества*.

Программные продукты всё чаще *встраиваются* в различные сложные системы реального времени. Работа над такими проектами требует от программных инженеров и тестировщиков, широкого взгляда и освоения *общих проблем проектирования и применения систем* определённого назначения и сферы использования. Программному инженеру необходимо участвовать в разработке требований для всей системы, а также освоить прикладную область применения создаваемого комплекса программ до начала тестирования и обдумывания функций компонентов, их характеристик и тестов, требованиям которых должен будет отвечать программный продукт.

Проблема целевого управления крупными проектами систем и программных продуктов включает рациональное использование и предупреждение потери ресурсов путём сбалансированного распределения их по частным работам и компонентам на протяжении всего жизненного цикла объекта.

Управление крупными проектами — особый вид инженерной деятельности, включающий: постановку задач; подготовку требований; планирование, организацию и стимулирование специалистов; контроль хода работ и использования ограниченных ресурсов при создании сложных систем. Критический параметр управления такими проектами — *ограниченное время*. Основная проблема целевого управления работами — сводить воедино усилия исполнителей-специалистов разной квалификации, подрядчиков и субподрядчиков, добиваясь, чтобы они

выступали при создании компонентов и всей системы как *команда*, а не как разрозненная группа независимых, функциональных специалистов. В результате должны обеспечиваться *концептуальная целостность системы* и высокое качество решения её главных целей и функций *при сбалансированном использовании ресурсов* на реализацию всей совокупности задач.

Проблему рационального сочетания целей, стратегий действий, конкретных процедур и доступных ресурсов необходимо решать для достижения *основной цели получения программного продукта с заданными требованиями и качеством* [3, 5]. Планирование проектов крупных комплексов программ должно *обеспечивать компромисс* между требующимися функциями и характеристиками создаваемой системы, и ограниченными ресурсами, необходимыми на её разработку и применение.

В жизненном цикле любых крупных систем и комплексов программ *четыре ключевые объекта* определяют их назначение, создание, свойства и качество:

- ✧ требования заказчика и/или пользователей к системе, определяющие цели создания и назначение программного продукта, его функции и область применения, необходимые пользователям системы и иным заинтересованным лицам, в заданной эксплуатационной среде;
- ✧ *тесты-эталоны и вторая адекватная форма описания содержания и функционирования программного продукта* для сквозной проверки полноты реализации требований и верификации соответствия исходным требованиям к системе, комплекса и компонентов программ, способствующие исключению дефектов и ошибок при выполнении и изменении требований;
- ✧ *эксплуатационная документация*, обеспечивающая применение программного продукта пользователями в соответствии с требованиями к его функциям и характеристикам, и являющаяся *третьим адекватным эталоном* корректности их реализации;
- ✧ *программный продукт и система* — *результат реализации требований*: функциональные характеристики системы, её поведение и предусмотренные свойства, реализованные в системе и комплексе программ для обеспечения корректного применения; критические параметры, характеристики качества и эффективности; экономические характеристики производства программного продукта и эксплуатационные риски.

Эти четыре объекта ниже используются как базовые при анализе тестирования в жизненном цикле крупных комплексов программ. Основное внимание сосредоточено на исследовании и формализации полноты и качества:

- ✧ разработки требований к функциям и характеристикам крупных программных продуктов;
- ✧ тестирования и обеспечения корректности реализации этих требований.

Основная цель тестирования комплексов программ и их функциональных компонентов — обнаружить, зарегистрировать и устранить дефекты и ошибки, которые внесены во время последовательной разработки и реализации требований к функциям и характеристикам комплекса программ. Тестированию программных компонентов и модулей посвящена обширная литература, однако в ней мало внимания уделяется тестированию крупных, целостных комплексов программ на соответствие исходным требованиям к их функциям. В статье *акцент изменён* — всё содержание ориентировано на тестирование крупных программных комплексов, а тестирование компонентов и модулей не рассматривается; считается, что они применяются готовыми, апробированными и высокого качества [1, 2, 11]. Однако допускается, что для идентификации, локализации и устранения дефектов и ошибок необходимо применять тестирование и корректировку модулей и небольших компонентов с использованием соответствующих традиционных методов и средств.

Формализация требований к программному продукту (первый эталон)

В требованиях к программному продукту должно быть зафиксировано соглашение между заказчиком и выполняющими проект специалистами, отражающее потребности заказчика и пользователей в таком виде, чтобы разработчик мог построить удовлетворяющий их комплекс программ и его компоненты [3, 5]. Требования являются *исходным эталоном* при разработке любой системы или программы, должны быть достаточно конкретными, чтобы при тестировании и испытаниях можно было установить, когда они удовлетворены. Вся разработка проекта должна вытекать из требований, а все спецификации программного комплекса и компонентов найти отражение в процессах и продуктах разработки. Эти данные должны устанавливать состав, содержание и значения результатов исполнения программ, получаемыми системой или пользователями при определённых условиях и исходных данных.

Любое отклонение результатов функционирования программы от предъявляемых к ней требований и сформированных по ним эталонов следует квалифицировать как дефект или ошибку в программном продукте и по возможности устранять.

При разработке требований к крупным системам и программным продуктам необходимо учитывать функции, основные свойства и общие требования к проектам крупных систем [3, 5, 6]. Формирование назначения, функций и технического задания на проект системы должно включать в себя основные требования к комплексу программ и особенности требований к нему заинтересованных лиц. Общие требования к качеству функционирования крупных программных продуктов реального времени обязательно включают: требования к надёжности и функциональной безопасности продуктов; требования к производительности и эффективности использования ресурсов ЭВМ программными продуктами в реальном времени, а также к допустимым рискам при их применении. Архитектурные требования к комплексам программ должны предусматривать управление изменениями требований, организацию изменений и сопровождение требований, обеспечение повторного использования программных компонентов и комплексов. Для обеспечения качества программного продукта необходима верификация и трассирование требований к функциям и компонентам комплекса, а также обеспечение баланса требований к ним. Должны быть обеспечены процессы достаточно полного и корректного документирования требований к функциям и характеристикам комплексов программ.

При формировании требований к системе и программному продукту заказчик и разработчики должны осуществлять согласованные действия в соответствии с принятой политикой и процедурами [7, 9, 10]:

- ✧ определить функциональные границы системы в терминах её поведения и предусмотренных свойств;
- ✧ определить каждую функцию, которая должна быть реализована в системе и программном продукте для обеспечения корректного применения;
- ✧ определить необходимые ограничения по реализации, обусловленные требованиями заказчика или неизбежными ограничениями системы, связанными с принятыми решениями;
- ✧ определить технические и потребительские характеристики, позволяющие оценивать технические результаты;

- ✧ задать системные требования и функции, в соответствии с которыми определяются риски и параметры системы и программного продукта, связанные с критическими показателями, надёжностью, безопасностью, производительностью;
- ✧ проанализировать целостность системных требований для обеспечения уверенности в том, что каждое требование и/или набор требований обладают системной целостностью;
- ✧ продемонстрировать связь между требованиями и практическими потребностями заказчика;
- ✧ на протяжении всего жизненного цикла вести учёт состояния совокупности требований вместе с их обоснованиями, связанными решениями и допущениями.

В требованиях должно устанавливаться *ожидаемое поведение системы и программного продукта*, выраженное, желательно, количественными характеристиками, а также границы их допустимых значений. Следует определить, насколько хорошо система, включая операторов-пользователей, приспособлена для реализации заданных функций. Требования могут зависеть от абстрактных представлений о подходящих характеристиках системы и включать методы и виды моделирования функционирования и среды, для достаточно полного описания заданных требований. Критические показатели функционирования анализируются и проверяются для подтверждения удовлетворения требований заказчика и для определения стоимости проекта, проектных графиков или эксплуатационных рисков, описывающих процессы установления, определения и использования соответствующих характеристик. Каждое положение *требований к системе и программному продукту должно проверяться* для установления его уникальности, полноты, непротиворечивости, совместимости с другими требованиями, реализуемости и проверяемости при испытаниях. Требуется удостоверять, что требования, с одной стороны, *необходимы и достаточны* для удовлетворения заказчика, а с другой – необходимые и достаточные входные данные для других процессов, например, *для проектирования функций и архитектуры комплекса программ*.

При формировании требований следует учитывать, что *источниками ошибок* в комплексах программ являются специалисты – конкретные люди с их индивидуальными особенностями, квалификацией, талантом и опытом. Можно выделить предсказуемые дефекты требований, модификаций, расширения и совершенствования комплекса, и необходимые изменения, обусловленные выявлением

случайных, непредсказуемых дефектов и ошибок [7, 9]. Поэтому плотность потоков и размеры необходимых *корректировок в требованиях* к комплексу и компонентам программ могут различаться в десятки раз. Однако в крупных комплексах программ статистика и распределение типов ошибок и выполняемых изменений для коллективов разных специалистов нивелируются и проявляются достаточно общие закономерности, могущие использоваться как ориентиры при их выявлении. Каждому типу необходимых корректировок соответствует более или менее определённая категория специалистов – источники дефектов данного типа. Эту корреляцию целесообразно учитывать как *общую качественную тенденцию* при анализе и поиске их причин. Этому могут помочь оценки типовых дефектов и корректировок путём их накопления и обобщения по опыту разработки определённых классов комплексов программ в конкретных предприятиях.

Одна из основных причин ошибок в комплексах программ – *организационные дефекты при определении и использовании требований к программному продукту*, отличающиеся от остальных типов и условно выделяемые как самостоятельные. Ошибки и дефекты данного типа появляются из-за недостаточного понимания коллективом специалистов целей и функций комплекса программ, а также вследствие отсутствия чёткой его организации и поэтапного контроля требований качества продуктов. Это порождается *пренебрежением руководителей к организации всего технологического процесса* формализации требований сложных программных продуктов и приводит к серьёзной недооценке их дефектов, трудоёмкости и сложности их выявления и устранения. Для сокращения этого типа массовых ошибок, активную роль должны играть лидеры – менеджеры и аналитики-системотехники, способные вести контроль и конфигурационное управление требованиями, изменениями и развитием версий и компонентов комплексов программ.

Формализация тестирования программного продукта (второй эталон)

Совокупности требований к тестам могут применяться как *эталон* и *вторая адекватная форма описания функций и характеристик комплексов программ*. Для обеспечения высокого качества программных продуктов стратегию создания *требований к тестам* целесообразно строить, *основываясь на требованиях к функциям и характеристикам* крупных комплексов программ. Требования к тестам можно дополнять структурным методом на основе анализа логики архитектуры системы. Он может

применяться в зависимости от условий договора или особых требований к безопасности или ограниченному риску системы. Например, покрытие полной структуры решений для некоторых функций может быть необходимо в аэрокосмических и других критических системах с повышенными требованиями безопасности.

Тестирование комплексов программ может иметь две цели, одинаково важные для качества тестирования. Первая цель – тестирование для идентификации дефектов – подразумевает успешность процедуры тестирования, если *дефект найден и устранён*. Это отличается от подхода в тестировании (вторая цель), когда тесты исполняются для демонстрации того, что программный комплекс удовлетворяет предъявляемым требованиям и, соответственно, тест считается успешным, если *не найдено дефектов*. Тестирование программ может использоваться для демонстрации наличия дефектов, но никогда не гарантирует их отсутствие. Основная причина этого в том, что *полное, всеобъемлющее тестирование недостижимо* для реального крупного программного продукта [1, 4, 9]. Когда пытаются оценивать методы тестирования, надо чётко определять, что подразумевается под их эффективностью, желательно, в количественных величинах. Только обладая такого рода данными можно говорить о корректности сравнения и оценки эффективности разных методов тестирования.

При разработке сложных комплексов программ надо иметь пригодные для применения тестирования *эталоны – требования и/или сценарии использования функций* программного продукта. Их необходимо анализировать и определять в терминах адекватных спецификаций требований к функциям и характеристикам программного продукта, а также к тестам, устанавливающим их содержание и корректность. Вторая форма описаний содержания программного продукта должна служить для сквозной верификации спецификаций требований к тестам сверху вниз, и подвергаться верификации на корректность соответствия исходным требованиям к комплексу и компонентам программ разного уровня.

Для обеспечения высокого качества программного продукта параллельно с *верификацией требований* и разработкой корректировок, следует разрабатывать и верифицировать *спецификации и сценарии тестов*, отражающие методы и конкретные процедуры проверки реализации этих требований. Тестовые спецификации могут использоваться для проверки согласованности, внутренней непротиворечивости и полноты реализации требований без исполнения программ. Для каждого требования

к функциям комплекса программ, его архитектуре, функциональным компонентам должна быть разработана спецификация требований к тестам, обеспечивающая проверку корректности, адекватности и возможности в последующем реализовать тестирование компонента на соответствие этому требованию. Взаимная проверка корректировок функций компонентов, отражённых в требованиях и в спецификациях тестов, обеспечивает повышение их качества, сокращение дефектов, ошибок, неоднозначностей и противоречий.

Спецификации тестов должны обеспечивать дополнительный контроль корректности спецификаций требований и верификацию взаимодействия компонентов на соответствующем уровне описания комплекса программ. Независимая разработка спецификаций тестов на основе спецификаций требований, создаёт базу для обнаружения не тестированных требований или требований, которые принципиально не могут быть проверены тестированием. Таким образом, *верификация спецификаций требований тестов* к функциям и характеристикам программных компонентов и комплекса могут использоваться с *двумя целями*:

- ✧ для разработки и проверки программ и интерфейсов взаимодействия программных компонентов разных уровней в комплексе программ;
- ✧ для создания требований к скоординированному комплексу тестов для проверки совокупности компонентов, обеспечивающих взаимную проверку реализации спецификаций требований комплексом программ.

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. Параллельная и независимая разработка спецификаций программ и спецификаций тестов, а также их реализация позволяет распараллеливать работы, что ведёт к сокращению сроков создания компонентов и комплексов программ требуемого качества.

Реализация этих целей верификации и тестирования может производиться разными методами и *независимыми специалистами* – программистами, интеграторами и тестировщиками. Это 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, и *мышление их разработчиков (программистов)* – на основе требований, функций, характеристик структуры и исполнения программ – существенно *отличаются от представлений и методов описаний тех же функций комплекса программ тестировщиками* – создателями сценариев и эталонов требуемых результатов тестирования. Они акцентируют свою деятельность на конкретных процедурах проверки функционирования, возможных результатах и взаимодействиях компонентов комплекса программ. Это позволяет выявлять вторичные дефекты, и повышать качество *сопоставлением двух методов* и результатов описания *одних и тех же функций и характеристик программ* за счёт того, что мала вероятность одинаковых ошибок в сценариях и реализациях тестов и в описаниях требований к функциям и характеристикам программ.

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.

Тестовое покрытие – это отображение пространства варьирования тестов на некоторую совокупность требований к функциям и характеристикам компонентов или комплекса программ, предназначенных для тестирования. Покрытие можно отражать *с помощью матрицы*, где по одной оси представлены тестовые сценарии, а по другой – требуемые функции, характеристики, безопасность или риски компонентов или комплекса программ [4, 6, 12]. При этом можно отслеживать степень покрытия требований тестами, поддерживаемыми конфигурацией и сценариями использования программного продукта. Такие документы – статические *графовые модели* функционирования программного продукта. Подобные модели могут фиксировать потоки и виды данных системы, последовательность состояний, в которых может находиться программный продукт и последовательность их обработки. Можно проводить оценку степени покрытия относительно этих моделей, проверяя, что каждому узлу и каждой связи в графе соответствует тестовый сценарий. Оценка

покрытия требует учёта, в какой степени тесты нацелены на пространство требуемых функций и характеристик комплекса программ. Точность определения степени покрытия ограничена, но выделение *сильных и слабых связей между тестами и пространством требований* к функциям и характеристикам продукта позволяет акцентировать анализ на безопасности, допустимых рисках или некоторых характеристиках, особенно важных для конкретного проекта.

Для обеспечения эффективности затрат ресурсов, крупномасштабное тестирование должно быть интегрировано как можно раньше с основными процессами проектирования, разработки и сопровождения в жизненном цикле программных комплексов, должно проводиться, начиная от анализа общих требований, заданных в техническом задании и/или в спецификации на всю систему, до детальных испытаний программных продуктов и их взаимодействия с внешней средой. По существу, требования к комплексу программ и тесты для проверки их адекватности и полноты, *отражают один и тот же объект*, но в разной форме. Поэтому сложность представительного описания комплекса тестов *соизмерима* со сложностью описания полной совокупности требований к функциям и характеристикам соответствующего комплекса программ. Вследствие этого *трудоёмкость и другие ресурсы* для разработки адекватных тестов *должны соответствовать ресурсам и трудоёмкости* при создании и реализации корректных требований, определяющих полностью программный продукт. Однако на практике это обычно не учитывается, и выделяемые на тестирование ресурсы оказываются значительно меньше и недостаточными для полноценного тестирования требований к крупным комплексам программ, что определяет их неполное соответствие требованиям и недостаточное качество.

Жизненный цикл разработки тестов и процессов тестирования программных комплексов должен проходить во времени, *параллельно динамике изменения требований* и жизненному циклу реализации комплекса программ. Тесты и сценарии их применения должны быть адекватными и полностью отражать содержание и изменения функций и характеристик компонентов комплексов программ, но в иной форме. Их следует рассматривать и анализировать как ещё одну форму описания изменений функций комплексов программ которые могут содержать дефекты и ошибки. Таким образом, программной (процедурной) форме представления программ должно полностью соответствовать его равноправное содержание в форме сценариев тестов

для проверки их взаимного соответствия. При этом дефекты и ошибки возможны в обеих формах интегрального описания комплекса программ (не только в текстах программного средства, но и в требованиях к тестам) определение их места и устранение – основная задача тестирования требований и тестов.

Квалификационное тестирование и испытания программного продукта на соответствие требованиям должно базироваться на организации, программе и методиках испытаний крупных программных продуктов. Тестирование на соответствие требованиям к динамическим характеристикам и рискам крупных программных продуктов должно включать тестирование надёжности и функциональной безопасности, характеристик производительности и динамического использования ресурсов ЭВМ, сокращение и ликвидацию опасных рисков при применении программных продуктов в реальном времени. Для обеспечения качества продукта важно тестирование эксплуатационной документации на соответствие требованиям к продукту. Тестирование должно поддерживать методы, процессы и средства управления конфигурацией требований и тестов крупных комплексов программ. Особое значение могут иметь процессы завершения испытаний и внедрения версий продуктов, анализ результатов, усовершенствование процессов тестирования требований и сертификация крупных программных продуктов.

Тестирование всегда предполагает компромисс между ограниченными ресурсами и заданными сроками, с одной стороны, и практически неограниченными требованиями к качеству результатов тестирования и программного продукта, с другой. Оценка стоимости и затрат и другие измерения процессов, связанных с оценкой ресурсов, необходимых для тестирования, как и оценка экономической эффективности тестирования на разных этапах и уровнях, основывается на точке зрения и практике менеджмента проектов и используется для оценки и улучшения процесса тестирования. Разная техника, концепции и модели тестирования требуют разных затрат по времени и необходимым ресурсам. Учёт соответствия между стоимостью или затратами, необходимыми для того или иного метода, и эффективностью тестирования – обязательная часть современного управления проектами крупных программных комплексов.

Характеристики качества встроенных программных продуктов реального времени зависят не только от их внутренних свойств, но и от свойств внешней среды – сценариев и генераторов динамических тестов, в которых они испытываются и применяются.

Для сокращения неопределённостей и прямых ошибок при оценивании качества комплексов программ необходимо предварительно определить основные параметры внешней среды, при которых должен функционировать комплекс программ с требуемыми характеристиками. Для этого заказчик и разработчик совместно должны структурировать, описать и согласовать модель внешней среды и её параметры в среднем, типовом режиме применения программного продукта, а также в наиболее вероятных и критических режимах генерации тестов с обеспечением требуемых характеристик качества функционирования. Необходимость проверять широкий набор требований и характеристик крупных программных комплексов приводит к необходимости динамического генерирования тестов посредством создания моделей внешней среды, охватывающих весь спектр данных, воздействующих на тестируемый объект. Подготовку тестов для испытаний крупных комплексов программ целесообразно осуществлять с применением автоматизированной генерации динамических тестов внешней среды в реальном времени, а также средств обработки результатов динамического тестирования.

Формализация эксплуатационной документации программных продуктов (третий эталон)

Эксплуатационная документация должна обеспечивать эффективное применение программного продукта и точно отражать его назначение, функции, характеристики и требования, для использования квалифицированными специалистами-пользователями [7, 8, 10]. Для этого эксплуатационные документы необходимо тестировать на полное соответствие выполнения всей совокупности требований на программный продукт, согласованной между разработчиками и заказчиком. В результате эксплуатационные документы можно использовать как отдельный, независимый при разработке третий эталон и вид тестов для квалификационного тестирования реализации требований к функциям и характеристикам программного продукта. Апробирование документов пользователями при применении комплекса программ – практический метод тестирования корректности реализации требований к программному продукту, завершающий полный цикл испытаний и контроля его качества. Разработчики этих документов должны обеспечивать комфортное и корректное применение программных продуктов пользователями на основе ясного и непротиворечивого изложения в документах технологических процедур и операций для штатного применения, функционирования и получения требуемых результатов.

Организация документирования должна определять стратегию, стандарты, процедуры, распределение ресурсов и планы создания, изменения и применения эксплуатационных документов на программный продукт. Для этого должны быть выделены *специалисты*, обязанные планировать, утверждать, выпускать, распространять и сопровождать комплекты апробированных и утверждённых эксплуатационных документов. Они должны стимулировать разработчиков программных средств, осуществлять непрерывное, регламентированное документирование процессов и результатов своей деятельности, контролировать полноту и качество утверждённых эксплуатационных и отчётных документов.

Состав и содержание комплекта документов конкретного программного продукта, целесообразно адаптировать разработчиками к его особенностям и свойствам *на основе использования стандартов и типовых структур-шаблонов* для двух классов продуктов, в наибольшей степени различающихся особенностями эксплуатации [8, 10]. *Первый класс* составляют программные продукты автоматизированного управления динамическими объектами и процессами в реальном масштабе времени. В процессе их применения допускается минимальное вмешательство пользователями в их процедуры управления, и необходим небольшой объём эксплуатационных документов, выделяемых из стандартизированного, технологического комплекта. Для продуктов *второго класса* (для административных систем) возможно применение пользователями широкого набора процедур управления, регламентированных достаточно полным набором и подробным содержанием документов.

Документация администрирования при эксплуатации систем и крупных программных продуктов должна обеспечивать поддержку первичной инсталляции, безопасного функционирования и восстановления программ и данных после сбоев. Администратор системы и программного продукта должен быть информирован обо всех изменениях функционирования устройств системы и внешней среды, могущих привести к сбою или возникновению аварийной ситуации, и предпринимать соответствующие профилактические действия. Для этого требуется полная информация о *требованиях к программному продукту*, компонентам системы и внешней среды. Каждый из документов административного управления программным продуктом *должен не противоречить международным стандартам* на коммуникацию, интерфейсы с пользователями и базами данных, на защиту и обеспечение безопасности информации.

Документация для оперативных пользователей программных продуктов в системе, их функционирования, обработки и анализа результатов должна обеспечивать взаимодействие пользователей с различными аппаратно-программными реализациями терминалов. Для этого необходима унификация концепции, архитектуры, функций и методов визуализации пользовательского интерфейса. Типовые *формы-шаблоны документов* и процедуры работы с ними, рассматриваемые как *объекты стандартизации*, относятся к функциональному, оперативному уровню взаимодействия пользователей с системами и программными продуктами в соответствии с исходными требованиями к ним.

Программы для вычислительных машин — исключительно *гибкие изделия*, что отражается на массовом, не формализованном, «художественном» создании относительно небольших программных компонентов с нерегламентированным качеством и документами. Однако программные продукты и системы, поставляемые заказчиком или на рынок должны иметь *конкретные, гарантированные и документированные цели, функции и характеристики*. 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. Крупные программные продукты обычно имеют длительный жизненный цикл с рядом версий; они совершенствуются, расширяются, повышается их качество. Поэтому процессы формирования изменений и реализации *требований, тестов и эксплуатационных документов должны организовываться на основе методологии регулируемых итераций множества версий*, в которых периоды поставляемых, стабильных, испытанных версий программных продуктов, чередуются с интервалами пошагового расширения функций, совершенствования качества и оперативного адаптирования к характеристикам внешней среды у пользователей или в системах. Для этого в крупных проектах целесообразно применять международные стандарты, регламентирующие жизненный цикл комплексов программ, и *жёсткую дисциплину* координируемой деятельности коллектива специалистов для сопровождения и документируемого *конфигурационного управления версиями* требований, тестов и документов для обеспечения качества реализации изменяющихся требований.

Заключение

В мире накоплено огромное количество готовых программных компонентов, сокращается необходимость программирования и тестирования новых

модулей и повторно используемых компонентов. Это приводит к *увеличению роли интеграции* готовых компонентов, применения соответствующих методов и инструментария программной инженерии. Однако вследствие их принципиальной новизны и сложности, они трудно воспринимаются традиционными программистами и тестировщиками модулей и небольших компонентов, и множеством преподавателей отечественной высшей школы. *Коренные отличия* между методами и инструментарием индивидуального, «художественного» программирования и тестирования небольших программ и технологией планомерной, регламентированной, инженерной разработки и тестирования крупных программных продуктов приводят к тому, что последние медленно осваиваются и входят в практику работы больших коллективов специалистов.

В стране быстро возрастает, как общий объём производства программных продуктов, так

и сложность отдельных проектов. Необходимость обеспечения взаимодействия распределённой вычислительной и коммуникационной техники предъявляет новые системотехнические требования к специалистам по программным комплексам. Это одна из причин возникновения проблем при разработке крупных программных продуктов, как и то, что многие предприятия, занимающиеся их производством, не уделяют должного внимания экономически эффективному применению современных методов управления проектами и международным стандартам, автоматизации и обеспечению всего жизненного цикла крупных программных комплексов. Эти проблемы *отражаются на экономике, качестве и конкурентоспособности отечественных программных продуктов*, на них целесообразно обращать внимание и решать заказчикам, руководителям и разработчикам крупных программных продуктов. ■

Литература

1. Бейзер Б. Тестирование черного ящика. Технология функционального тестирования программного обеспечения и систем. Пер. с англ. – СПб: ПИТЕР. 2004.
2. Блэк Р. Ключевые процессы тестирования. Пер. с англ. – М: ЛОРИ. 2006.
3. Вигерс К.И. Разработка требований к программному обеспечению. Пер. с англ. – М.: Русская редакция. 2004.
4. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. Пер. с англ. – М. ЛОРИ. 2003.
5. Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. Пер. с англ. – М.: Вильямс. 2002.
6. Липаев В.В. Отладка сложных программ. М.: Энергоатомиздат. 1993.
7. Липаев В.В. Программная инженерия. Методологические основы. Учебник. – М.: ТЕИС. 2006.
8. Соммервилл И. Инженерия программного обеспечения. 6-е издание. Пер. с англ. – М.: Вильямс. 2002.
9. Тэллес М., Хсих Ю. Наука отладки. Пер. с англ. – М.: Кудиц-образ. 2003.
10. Фатрелл Р. Т., Шафер Д. Ф., Шафер Л. И. Управление программными проектами: достижение оптимального качества при минимальных затратах. Пер. с англ. – М.: Вильямс. 2003.
11. Beizer B. Software testing techniques. N.Y.: Van Nostrand Reinhold. 1990.
12. Kit E. Software Testing in the Real World. NY. Addison-Wesley. 1996.