Лекция 8. Системное тестирование

Лекция является последней из трех рассматривающих уровни процесса верификации. Тема данной лекции - процесс системного тестирования, его задачи и цели. Рассматриваются виды системного тестирования, особенности системного тестирования и испытаний при разработке сертифицируемого программного обеспечения. Цель данной лекции: дать представление о процессе системного тестирования, его технической и организационной составляющих

8.1. Задачи и цели системного тестирования

По завершению интеграционного тестирования все модули системы являются согласованными по интерфейсам и функциональности. Начиная с этого момента можно переходить к тестированию системы в целом как единого объекта тестирования - к *системному тестированию*. На уровне интеграционного тестирования тестировщика интересовали в основном структурные аспекты системы, на уровне системного тестирования интересуют поведенческие аспекты системы. Как правило, для системного тестирования применяется подход черного ящика, при этом в качестве входных и выходных данных используются реальные данные, с которыми работает система, или данные, подобные им.

Системное тестирование - один из самых сложных видов тестирования. На этом этапе проводится не только функциональное тестирование, но и оценка характеристик качества системы - ее устойчивости, надежности, безопасности и производительности. На этом этапе выявляются многие проблемы внешних интерфейсов системы, связанные с неверным взаимодействием с другими системами, аппаратным обеспечением, неверным распределением памяти, отсутствием корректного освобождения ресурсов и т.п.

После завершения системного тестирования разработка переходит в фазу приемо-сдаточных испытаний (для программных систем, разрабатываемых на заказ) или в фазу альфа- и бета-тестирования (для программных систем общего применения).

Поскольку системное тестирование - процесс, требующих значительных ресурсов, для его проведения часто выделяют отдельный коллектив тестировщиков, а зачастую системное тестирование выполняется организацией, которая не связана с коллективом разработчиков и тестировщиков, выполнявших работы на предыдущих этапах тестирования. При этом необходимо отметить, что при разработке некоторых типов программного обеспечения (например, авиационного бортового) требование независимого тестирования на всех этапах разработки является обязательным.

Системное тестирование проводится в несколько фаз, на каждой из которых проверяется один из аспектов поведения системы, т.е. проводится один из типов системного тестирования. Все эти фазы могут протекать одновременно или последовательно. Следующий раздел посвящен рассмотрению особенностей каждого из типов системного тестирования на каждой фазе.

8.2. Виды системного тестирования

Принято выделять следующие виды системного тестирования:

- функциональное тестирование;
- тестирование производительности;

- нагрузочное или *стрессовое тестирование*;
- тестирование конфигурации;
- *тестирование безопасности*;
- тестирование надежности и восстановления после сбоев;
- *тестирование удобства использования*.

В ходе системного тестирования проводятся далеко не все из перечисленных видов тестирования - конкретный их набор зависит от тестируемой системы.

Исходной информацией для проведения перечисленных видов тестирования являются два класса требований: функциональные и нефункциональные. Функциональные требования явно описывают, что система должна делать и какие выполнять преобразования входных значений в выходные. Нефункциональные требования определяют свойства системы, напрямую не связанные с ее функциональностью. Примером таких свойств может служить время отклика на запрос пользователя (например, не более 2 секунд), время бесперебойной работы (например, не менее 10000 часов между двумя сбоями), количество ошибок, которые допускает начинающий пользователь за первую неделю работы (не более 100), и т.п.

Рассмотрим каждый вид тестирования подробнее.

Функциональное тестирование. Данный вид тестирования предназначен для доказательства того, что вся система в целом ведет себя в соответствии с ожиданиями пользователя, формализованными в виде системных требований. В ходе данного вида тестирования проверяются все функции системы с точки зрения ее пользователей (как пользователей-людей, так и "пользователей" - других программных систем). Система при функциональном тестировании рассматривается как черный ящик, поэтому в данном случае полезно использовать классы эквивалентности. Критерием полноты тестирования в данном случае будет полнота покрытия тестами системных функциональных требований (или системных тест-требований) и полнота тестирования классов эквивалентности, а именно:

- все функциональные требования должны быть протестированы;
- все классы допустимых входных данных должны корректно обрабатываться системой;
- все классы недопустимых входных данных должны быть отброшены системой, при этом не должна нарушаться стабильность ее работы;
- в тестовых примерах должны генерироваться все возможные классы выходных данных системы;
- во время тестирования система должна побывать во всех своих внутренних состояниях, пройдя при этом по всем возможным переходам между состояниями.

Результаты системного тестирования протоколируются и анализируются совершенно аналогично тому, как это делается для модульного и интеграционного тестирования. Основная сложность здесь заключается в локализации дефектов в программном коде системы и определении зависимостей одних дефектов от других (эффект "четного числа ошибок").

Тестирование производительности. Данный вид тестирования направлен на определение того, что система обеспечивает должный уровень производительности при обработке пользовательских запросов. Тестирование производительности выполняется при различных уровнях нагрузки на систему, на различных конфигурациях оборудования.

Выделяют три основных фактора, влияющие на производительность системы: количество поддерживаемых системой потоков (например, пользовательских сессий), количество свободных системных ресурсов, количество свободных аппаратных ресурсов.

Тестирование производительности позволяет выявлять узкие места в системе, которые проявляются в условиях повышенной нагрузки или нехватки системных ресурсов. В этом случае по результатам тестирования проводится доработка системы, изменяются алгоритмы выделения и распределения ресурсов системы.

Все требования, относящиеся к производительности системы, должны быть четко определены и обязательно должны включать в себя числовые оценки параметров производительности. Т.е., например, требование "Система должна иметь приемлемое время отклика на запрос пользователя" является непригодным для тестирования. Напротив, требование "Время отклика на запрос пользователя не должно превышать 2 секунды" может быть протестировано.

То же самое относится и к результатам тестирования производительности. В отчетах по данному виду тестирования сохраняют такие показатели, как загрузка аппаратного и системного программного обеспечения (количество циклов процессора, выделенной памяти, количество свободных системных ресурсов и т.п.). Также важны скоростные характеристики тестируемой системы (количество обработанных в единицу времени запросов, временные интервалы между началом обработки каждого последующего запроса, равномерность времени отклика в разные моменты времени и т.п.).

Для проведения тестирования производительности требуется наличие генератора запросов, который подает на вход системы поток данных, типичных для сеанса работы с ней. Тестовое окружение должно включать в себя кроме программной компоненты еще и аппаратную, причем на таком тестовом стенде должна существовать возможность моделирования различного уровня доступных ресурсов.

Стрессовое тестирование. *Стрессовое тестирование* имеет много общего с тестированием производительности, однако его основная задача - не определить производительность системы, а оценить производительность и устойчивость системы в случае, когда для своей работы она выделяет максимально доступное количество ресурсов либо когда она работает в условиях их критической нехватки. Основная цель стрессового тестирования - вывести систему из строя, определить те условия, при которых она не сможет далее нормально функционировать. Для проведения стрессового тестирования используются те же самые инструменты, что и для тестирования производительности. Однако, например, генератор нагрузки при стрессовом тестировании должен генерировать запросы пользователей с максимально возможной скоростью либо генерировать данные запросов таким образом, чтобы они были максимально возможными по объему обработки.

Стрессовое тестирование очень важно при тестировании web-систем и систем с открытым доступом, уровень нагрузки на которые зачастую очень сложно прогнозировать.

Тестирование конфигурации. Большинство программных систем массового назначения предназначено для использования на самом разном оборудовании. Несмотря на то, что в настоящее время особенности реализации периферийных устройств скрываются драйверами операционных систем, которые имеют унифицированный с точки зрения прикладных систем интерфейс, проблемы совместимости (как программной, так и аппаратной) все равно существуют.

В ходе тестирования конфигурации проверяется, что программная система корректно работает на всем поддерживаемом аппаратном обеспечении и совместно с другими программными системами. Необходимо также проверять, что система продолжает стабильно работать при горячей замене любого поддерживаемого устройства на аналогичное. При этом система не должна давать сбоев ни в момент замены устройства, ни после начала работы с новым устройством.

Также необходимо проверять, что система корректно обрабатывает проблемы, возникающие в оборудовании, как штатные (например, сигнал конца бумаги в принтере), так и нештатные (сбой питания).

Тестирование безопасности. Если программная система предназначена для хранения или обработки данных, содержимое которых представляет собой тайну определенного рода (личную, коммерческую, государственную и т.п.), то к свойствам системы, обеспечивающим сохранение этой тайны, будут предъявляться повышенные требования. Эти требования должны быть проверены при тестировании безопасности системы. В ходе этого тестирования проверяется, что информация не теряется, не повреждается, ее невозможно подменить, а также к ней невозможно получить несанкционированный доступ, в том числе при помощи использования уязвимостей в самой программной системе.

В отечественной практике принято проводить сертификацию программных систем, предназначенных для хранения данных для служебного пользования, секретных, совершенно секретных и совершенно секретных особой важности. Существует ряд отечественных стандартов Федеральной службы по техническому и экспортному контролю (ФСТЭК), регламентирующих свойства программных систем по обеспечению необходимого уровня безопасности [5] и по отсутствию недокументированных возможностей ("закладок") [4], которые могут быть использованы злоумышленником для несанкционированного доступа к данным. Кроме того, существует международный стандарт *Common Criteria* [37], также регламентирующий вопросы защиты информации в программных системах.

Несмотря на то, что сертификация - процесс, следующий за верификацией, требования этих стандартов могут быть использованы и при тестировании системы. Так, стандарт [5] ФСТЭК, традиционно сокращенно называемый РД СВТ, выделяет следующие группы свойств программной системы, подлежащие проверке (некоторые группы свойств укрупнены для сокращения списка):

- разграничение и контроль доступа - предотвращение доступа к "чужой" информации;
- очистка и защита памяти - предотвращение доступа к остаточной информации после удаления объектов из памяти;
- маркировка и защита информации, передаваемой во внешний мир - сохранение уровня секретности даже вне системы;
- идентификация и аутентификация - предоставление доступа только санкционированным пользователям и отказ в доступе всем остальным;
- регистрация (аудит событий) - регистрация в специальном журнале всех событий системы, связанных с безопасностью для последующего анализа;
- гарантии проектирования и архитектуры - система должна быть спроектирована таким образом, чтобы гарантировать защищенность информации с определенным уровнем уверенности;

- тестирование - все функции по обеспечению безопасности должны быть протестированы во всех режимах;
- *целостность и восстановление* средств защиты - система должна иметь средства контроля корректности всех правил разграничения доступа и системы безопасности в целом, а также средства их восстановления при сбое;
- документация разработчика, администратора и пользователя - все средства системы по обеспечению безопасности должны быть описаны в соответствующих руководствах.

При разработке и верификации программной системы, которая будет подвергаться последующей сертификации, работы по сертификации должны включать в себя проверку всех перечисленных свойств.

Тестирование надежности и восстановления после сбоев. Для корректной работы системы в любой ситуации необходимо удостовериться в том, что она восстанавливает свою функциональность и продолжает корректно работать после любой проблемы, прервавшей ее работу. При тестировании восстановления после сбоев имитируются сбои оборудования или окружающего программного обеспечения либо сбои программной системы, вызванные внешними факторами. При анализе поведения системы в этом случае необходимо обращать внимание на два фактора - минимизацию потерь данных в результате сбоя и минимизацию времени между сбоем и продолжением нормального функционирования системы

Тестирование удобства использования. Отдельная группа нефункциональных требований - требования к удобству использования пользовательского интерфейса системы. Этот вид тестирования будет рассмотрен в следующей лекции.

В результате выполнения всех рассмотренных выше видов тестирования делается заключение о функциональности и свойствах системы, после чего узкие места системы дорабатываются до реализации необходимой функциональности или до достижения системой необходимых свойств.

8.3. Системное тестирование, приемо-сдаточные и сертификационные испытания при разработке сертифицируемого программного обеспечения

При разработке массового ("коробочного", *COTS*) программного обеспечения после проведения системного тестирования система проходит этапы альфа- и бета-тестирования, во время которого работу системы проверяют потенциальные пользователи (либо специально выделенные фокус-группы пользователей, либо все желающие). На этом этапе в программную систему вносятся последние незначительные изменения, не влияющие на суть системы. После завершения этой стадии система поступает в продажу конечным пользователям.

При разработке заказного программного обеспечения фазу альфа- и бета-тестирования заменяют приемо-сдаточные испытания. Во время этих испытаний заказчик удостоверяется, что система работает в соответствии с его потребностями (как зафиксированными в техническом задании на систему, так и не зафиксированными). Заказчик может проводить такие испытания самостоятельно, выполняя заранее подготовленные тесты системы, либо проводить их совместно с представителями коллектива разработчиков. В этом случае тестовые примеры также готовятся разработчиками, например, на основе тестовых примеров, использовавшихся на этапе системного тестирования.

Завершаются приемо-сдаточные испытания либо подписанием акта приемки, либо выдачей заказчиком дополнительных требований к системе, которые должны быть исправлены до приемки системы. После устранения всех недостатков системы приемо-сдаточные испытания повторяются (возможно, по сокращенной программе). После успешного подписания акта система поступает в эксплуатацию заказчику.

Существует специальный вид программных систем, к свойствам которых предъявляются особые требованиями. Примером таких систем могут служить бортовые авиационные программные системы, для которых особое внимание уделяется вопросам безопасности, надежности и отказоустойчивости. Несмотря на то, что большая часть таких систем может быть отнесена к категории заказного программного обеспечения, для получения разрешения на установку системы на борт требуется получение сертификата на летную пригодность.

Таким образом, после проведения системного тестирования и приемо-сдаточных испытаний проводятся сертификационные испытания. Сертификация программного обеспечения - процесс установления и официального признания того, что разработка ПО проводилась в соответствии с определенными требованиями. В процессе сертификации происходит взаимодействие заявителя, сертифицирующего органа и наблюдательного органа.

Заявитель - это организация, подающая заявку в соответствующий сертифицирующий орган на получение сертификата (соответствия, качества, годности и т.п.) изделия.

Сертифицирующий орган - организация, которая рассматривает заявку заявителя о проведении сертификации ПО и либо самостоятельно, либо путем формирования специальной комиссии производит набор процедур, направленных на проведение процесса сертификации ПО заявителя.

Наблюдательный орган - комиссия специалистов, наблюдающих за процессами разработки заявителем сертифицируемой информационной системы и дающих заключение о соответствии данного процесса определенным требованиям, которое передается на рассмотрение в сертифицирующий орган. [34]

Основной объект проверки в ходе сертификационных испытаний - удовлетворяет ли процесс разработки программной системы регламенту и рекомендациям стандарта, на соответствие которому проводится сертификация. Такое соответствие определяется при помощи анализа жизненного цикла сертифицируемой системы и документов, создаваемых на ключевых его этапах. Весь процесс анализа и те свойства системы, которые подвергаются сертификации, описывается в плане сертификационных испытаний, который утверждается совместно заявителем и сертифицирующим органом.

В случае сертификации бортовой системы по стандарту DO-178B (или его аналогам КТ-178, JB-12 и т.п.) план дополнительно определяет уровень влияния отказа программной системы на безопасность полета (уровень отказобезопасности) по которому будет проводиться сертификация. Любые вопросы, которые возникают у сертифицирующего органа относительно содержания плана сертификационных испытаний, должны быть разрешены до начала самих испытаний.

Согласно требованиями DO-178B план сертификационных испытаний (план программных аспектов сертификации) должен включать:

- **обзор системы**. Этот раздел описывает систему, включая описание ее функций и их размещение в программном и аппаратном обеспечении, ее архитектуру, используемый процессор (процессоры), аппаратно-программный интерфейс, и особенности отказобезопасности;
- **обзор программного обеспечения**. Этот раздел коротко описывает функции программного обеспечения с акцентом на концепцию обеспечения отказобезопасности и разделения на обособленные части, например, распределение ресурсов, резервирование, несимметрично резервированное программное обеспечение, *устойчивость к отказам*, стратегии таймирования и диспетчеризирования;
- **сертификационные соображения**. Этот раздел содержит сводку сертификационного базиса, включая средства подтверждения соответствия, как это определяется программными аспектами сертификации. В этом разделе также заявляется предложенный уровень (уровни) программного обеспечения и приводятся подтверждения правильности этого уровня, полученные в процессе оценки отказобезопасности системы, включая потенциальный вклад программного обеспечения в отказные ситуации;
- **жизненный цикл программного обеспечения**. Этот раздел определяет *жизненный цикл программного обеспечения*, который будет использоваться, а также включает сводку его процессов, детальная информация о которых определяется в соответствующих планах программного обеспечения. В сводке разъясняется, как будут удовлетворяться цели каждого процесса жизненного цикла, указываются вовлекаемые организации, организационная ответственность, а также ответственность за процессы жизненного цикла системы и за процесс поддержания контактов в ходе сертификации;
- **данные жизненного цикла программного обеспечения**. Этот раздел определяет данные жизненного цикла, которые будут выпущены и будут контролироваться в процессах жизненного цикла программного обеспечения. Этот раздел также описывает взаимосвязь данных между собой или с другими данными, определяющими систему, данные жизненного цикла программного обеспечения, представляемые сертифицирующим властям, форму данных и средства, с помощью которых данные жизненного цикла программного обеспечения могут быть сделаны доступными для сертифицирующих властей;
- **план-график**. Этот раздел описывает средства, которые заявитель будет использовать для того, чтобы обеспечить для сертифицирующих властей обозримость деятельности в процессах жизненного цикла программного обеспечения и, следовательно, возможность планирования проверок;
- **дополнительные соображения**. Этот раздел описывает особенности, которые могут повлиять на процесс сертификации, например, альтернативные методы подтверждения соответствия, квалификацию инструментальных средств, ранее разработанное программное обеспечение, вариантное программное обеспечение, которое может быть выбрано по желанию, программное обеспечение, доступное для модификации пользователем, готовое программное обеспечение *COTS*, используемое без модификаций, программное обеспечение, загружаемое в полевых условиях, несимметрично резервированное программное обеспечение или использование истории эксплуатации продукта.

В процессе самих сертификационных испытаний заявитель предоставляет свидетельства того, что процессы жизненного цикла программного обеспечения удовлетворяют планам программного обеспечения. Заявитель организует доступ сертифицирующего органа к данным жизненного цикла программного обеспечения. При этом минимальный перечень этих данных включает в себя:

- план сертификационных испытаний (план программных аспектов сертификации);
- индекс конфигурации программного обеспечения - документ, который должен однозначно идентифицировать каждый компонент проекта (включая требования, исходные коды, объектный и исполняемый код), среду реализации системы, инструкции по компиляции системы, аппаратное и программное обеспечение для работы системы, аппаратное и программное обеспечение для проведения сертификации.
- итоговое заключение о программном обеспечении.

Итоговое заключение по программному обеспечению является основным документом для демонстрации соответствия программного обеспечения Плану программных аспектов сертификации. Итоговое заключение должно включать:

- **обзор системы**. Этот раздел содержит обзор системы, включая описание ее функций и их размещения в аппаратном и программном обеспечении, архитектуру, используемый процессор (процессоры), аппаратно-программный интерфейс, средства обеспечения отказобезопасности. В этом разделе также описываются все отличия от описания системы, ранее помещенного в план программных аспектов сертификации;
- **обзор программного обеспечения**. Этот раздел кратко описывает функции программного обеспечения (особое внимание уделяется используемой концепции отказобезопасности и разделения на обособленные части), а также разъясняет отличия от обзора программного обеспечения, ранее помещенного в план программных аспектов сертификации;
- **сертификационные соображения**. Этот раздел повторно формулирует сертификационные соображения, приведенные в плане программных аспектов сертификации, а также описывает любые отличия от ранее приведенных соображений;
- **характеристики программного обеспечения**. Этот раздел констатирует данные о размере исполняемого кода, запасах по времени и памяти, ограничениях ресурсов, а также описывает средства для измерения каждой характеристики;
- **жизненный цикл программного обеспечения**. Этот раздел суммирует реальный жизненный цикл (циклы) программного обеспечения и разъясняет отличия от жизненного цикла программного обеспечения и процессов жизненного цикла, ранее предложенных в плане программных аспектов сертификации;
- **данные жизненного цикла программного обеспечения**. Этот раздел дает ссылку на данные жизненного цикла программного обеспечения, продуцируемые в процессах разработки программного обеспечения и процессах обеспечения целостности. Он описывает взаимосвязь данных между собой и с другими данными, определяющими систему, и средства, с помощью которых к данным жизненного цикла программного обеспечения может быть обеспечен доступ со стороны сертифицирующих властей. Этот раздел также описывает любые отличия от описания данных жизненного цикла, ранее помещенного в плане программных аспектов сертификации;
- **дополнительные соображения**. Этот раздел суммирует вопросы, которые могут привлечь внимание сертифицирующих властей, и дает ссылки на данные, применимые к этим вопросам, такие, как выпущенные документы или специальные условия;
- **идентификация программного обеспечения**. Этот раздел идентифицирует конфигурацию программного обеспечения по номенклатурному номеру или версии;

- **история изменений**. Этот раздел, если это применимо, включает сводку изменений программного обеспечения. Особое внимание уделяется изменениям, которые сделаны для исправления ошибок, влияющих на отказобезопасность, а также идентификацию изменений в процессах жизненного цикла программного обеспечения со времени предыдущей сертификации;
- **статус программного обеспечения**. Этот раздел содержит сводку сообщений о проблемах, не разрешенных на момент сертификации, включая заявления о функциональных ограничениях;
- **заявление о соответствии**. Этот раздел включает заявление о соответствии программного обеспечения настоящему документу, а также сводку методов, использованных для демонстрации соответствия с указанием критериев, которые специфицированы в планах программного обеспечения. В этом разделе также указываются дополнительные, по отношению к планам программного обеспечения, стандартам и настоящему документу, использованные правила и отклонения от планов, стандартов и настоящего документа.

Полный перечень данных жизненного цикла, которые могут понадобиться при сертификации, включает в себя:

- план программных аспектов сертификации;
- план разработки программного обеспечения;
- план верификации программного обеспечения;
- план управления конфигурацией программного обеспечения;
- план гарантии качества программного обеспечения;
- стандарты на требования к программному обеспечению;
- стандарты проектирования программного обеспечения;
- стандарты на код программного обеспечения;
- данные требований на программное обеспечение;
- описание проекта;
- исходный текст;
- исполняемый объектный код;
- тестовые примеры и тестовые процедуры верификации программного обеспечения;
- отчет по результатам верификации программного обеспечения;
- индекс конфигурации окружающей среды жизненного цикла программного обеспечения;
- индекс конфигурации программного обеспечения;
- сообщения о проблемах;
- документы по управлению конфигурацией программного обеспечения;
- документы по гарантии качества программного обеспечения;
- итоговое заключение по программному обеспечению.

Сертифицирующий орган устанавливает т.н. сертификационный базис для системы в ходе консультаций с заявителем. Сертификационный базис определяет конкретные правила вместе с любыми специальными условиями, которые могут дополнять опубликованные правила сертификации, регламентированные стандартом.

Для программного обеспечения установка базиса производится по рассмотрению итогового заключения о программном обеспечении и свидетельств соответствия.

В ходе сертификации сертифицирующий орган оценивает план программных аспектов сертификации на полноту и согласованность с критериями оценки отказобезопасности системы и другими данными жизненного цикла программного обеспечения. Если верны

все данные жизненного цикла, являющиеся доказательством того, что в ходе проекта были активны все необходимые процессы разработки и верификации, то сертифицирующий орган выдает положительное решение о выдаче сертификата.

Сертификаты на программное обеспечение можно отнести к двум типам: сертификаты соответствия и сертификаты качества.

- **Сертификат качества** - свидетельство, удостоверяющее качество фактически поставленного товара и его соответствие условиям договора. В сертификате качества дается характеристика товара либо подтверждается соответствие товара определенным стандартам или техническим условиям заказа. Сертификат качества выдается компетентными организациями, торговыми палатами, специальными лабораториями как в стране экспорта, так и импорта. Стороны договора купли-продажи могут договориться о предоставлении сертификатов различных контрольных и проверочных учреждений.
- **Сертификат соответствия** - результат действий третей стороны (документ), подтверждающий уверенность в том, что должным образом идентифицированная продукция, процесс или услуга соответствуют конкретному стандарту или другому нормативному документу.

Сертификат на летную пригодность в рассматриваемом примере сочетает в себе свойства обоих типов сертификатов. С одной стороны, он удостоверяет, что разработанная система имеет определенный уровень качества реализации, а с другой - что процессы по ее разработке соответствуют международному авиационному отраслевому стандарту.