

**Тема: «Документация, сопровождающая
процесс верификации и тестирования»**

**Дисциплина: «Верификация, стандартизация и
сертификация ПО»**

Лектор: старший преподаватель кафедры ИВС Савченко Н.К.

План лекции:

1. Проектная документация.
2. Стратегия и планы верификации.
3. Тест-требования.
4. Тест-планы.



1 Проектная документация

В ходе работы над проектом по созданию любой сложной программной системы создается большое количество проектной документации. Основное ее назначение: координация совместных действий большого количества разработчиков в течение более или менее длительных промежутков времени - в процессе первоначальной разработки системы, в процессе выполнения работ по ее модификации, в процессе сопровождения.

Верификация программной системы (в оптимальном случае) выполняется в течение всего жизненного цикла разработки достаточно большим коллективом разработчиков. Основное назначение документации, сопровождающей процесс верификации, помимо синхронизации действий тестировщиков различных уровней, - обеспечение гарантий того, что тестирование выполняется в соответствии с выбранными критериями оценки качества, а также того, что все аспекты поведения системы протестированы (рисунок 3.1).



Рисунок 3.1 - Документация, сопровождающая процесс верификации

Перед началом верификации менеджером тестирования (test manager) создается документ, называемый планом верификации (или планом тестирования, но это не то же самое, что тест-план). План тестирования - организационный документ, содержащий требования к тому, как должно выполняться тестирование в данном конкретном проекте.



В нем определяются общие подходы к согласованию процессов разработки и верификации, определяются методики проведения верификации, состав тестовой документации и ее взаимосвязь с документацией разработчиков, сроки различных этапов верификации, различные роли и квалификация тестировщиков, необходимые для выполнения всех работ по тестированию, требования к инструментам тестирования и тестовым стендам, а также оцениваются риски и приводятся пути для их преодоления.

В данном документе также определяются требования собственно к тестовой документации - тест-требованиям, тест-планам, отчетам о выполнении тестирования. Согласно этим требованиям по системным и функциональным требованиям разработчиками тестов (test procedure developers) создаются тест-требования - документы, в которых подробно описано то, какие аспекты поведения системы должны быть протестированы. На основании описания архитектуры создаются низкоуровневые тест-требования, где описываются аспекты поведения конкретной программной реализации системы, которые необходимо протестировать.

На основании тест-требований разработчиками тестов (test developers) создаются тест-планы - документы, которые содержат подробное пошаговое описание того, как должны быть протестированы тест-требования.

На основании тест-требований и проектной документации разработчиков также создается тестовое окружение, необходимое для корректного выполнения тестов на тестовых стендах - драйверы, заглушки, настроечные файлы и т.п.

По результатам выполнения тестов тестировщиками (testers) создаются отчеты о выполнении тестирования (они могут создаваться либо автоматически, либо вручную), которые содержат информацию о том, какие несоответствия требованиям были выявлены в результате тестирования, а также отчеты о покрытии, содержащие информацию о том, какая доля программного кода системы была задействована в результате выполнения тестирования.



По несоответствиям создаются отчеты о проблемах - документы, которые направляются на анализ в группу разработчиков с целью определения причины возникновения несоответствия.

Изменения в систему вносятся только после всестороннего изучения этих отчетов и локализации проблем, вызвавших несоответствие требованиям.



Для того, чтобы процесс изменений не вышел из под контроля и любое изменение протоколировалось (и связывалось с тестами, обнаружившими проблему), создается запрос на изменение системы. После завершения всех работ по запросу на изменение процесс тестирования повторяется до тех пор, пока не будет достигнут приемлемый уровень качества программной системы.



Следует особо отметить, что все документы должны иметь уникальные идентификаторы и храниться в единой базе документов проекта.

Это позволит сохранить управляемость процессом тестирования и поддерживать необходимое качество разрабатываемой системы. Нет ничего хуже ситуации, когда найденная проблема не была исправлена из-за того, что отчет о ней был утерян и не попал к разработчику.

2 Стратегия и планы верификации

Первый документ, входящий в состав технологической документации процесса верификации - стратегия тестирования. Стратегия верификации определяет общие подходы и методики верификации, необходимые уровни верификации проектной документации и программного кода, технологии и инструментальные средства.

Другой, не менее важный документ, создаваемый перед началом процесса верификации - план верификации.

Этот план содержит последовательное описание всех этапов верификации, процедур на каждом этапе и связей с этапами разработки.

Для каждого этапа определяется:

- типы входных и выходных документов;
- общая процедура верификации;
- роли и ответственности;
- форматы и соглашения по идентификации выходных документов;
- критерии оценки результативности этапа.

Основная задача плана тестирования как документа - определение границ тестирования, подхода к тестированию, требуемых для тестирования ресурсов, плана-графика тестирования.

План тестирования определяет тестируемые элементы и функции системы, задачи, решаемые в ходе тестирования, сотрудников, ответственных за тестирование, и риски, связанные с этим планом, минимально необходимые элементы, которые рекомендуется включать в каждый план тестирования.

К ним относят:

- идентификатор плана тестирования и номер его версии, который позволяет однозначно находить нужный план тестирования и его последнюю актуальную версию в базе данных проекта;
- общее описание тест-плана;
- трассировка на другие документы - стандарты, планы тестирования, тест-требования, результаты выполнения тестов;
- определение тестируемых областей системы - указание частей проектной документации, исходных текстов, исполняемого кода, подвергаемых верификации с указанием типа верификации (автоматизированные тесты, формальные инспекции, тестирование на моделях, полунатурные испытания и т.п.);

- определение подходов к тестированию - определение общих методик, которых следует придерживаться при тестировании системы. Несмотря на то, что большинство тестов могут довольно сильно различаться, общие методы и подходы к их построению могут быть едиными;
- критерий успешности/неуспешности прохождения тестов (pass/fail criteria) - в данном разделе описывается то, каким образом определяется успешность прохождения тестов для различных частей системы;
- тестовые документы - как правило, план тестирования содержит в качестве приложений все тестовые документы более низких уровней - тест-требования, тест-планы, результаты выполнения тестов, данные о сборе покрытия;

- требования к среде тестирования - данный раздел описывает требования к аппаратным и программным средствам, необходимым для проведения тестирования;
- людские ресурсы и уровень их подготовки - в данном разделе приводится состав группы тестирования, необходимый для успешного завершения тестирования в поставленные сроки, а также приводятся необходимые знания для различных ролей в группе;
- план-график тестирования - содержит сроки всех фаз тестирования;
- риски - содержит список рисков, которые могут помешать завершить тестирование в срок или с необходимым уровнем качества.

3 Тест-требования

Тест-требования - основной документ для тестировщика, который определяет функциональность системы с точки зрения того, что должно быть проверено, чтобы удостовериться в ее корректном функционировании, а также - на основании какого внешнего эффекта можно убедиться, что проверяемая функция реализована правильно.

Существует два подхода к написанию тест-требований - функциональный и структурный.

Тест-требования, написанные в рамках функционального подхода, основываются на системных требованиях и требованиях к программному обеспечению системы.

Тест-требования, написанные в рамках структурного подхода, пишутся на основании описания архитектуры системы и принимают в расчет строение исходных текстов системы.

Структурные тест-требования важны в том случае, когда к надежности системы предъявляются повышенные требования.



На практике почти всегда применяются оба подхода к разработке тест-требований. В результате в состав документации проекта включаются тест-требования верхнего уровня и тест-требования нижнего уровня, по которым составляются тест-планы (рисунок 3.2).

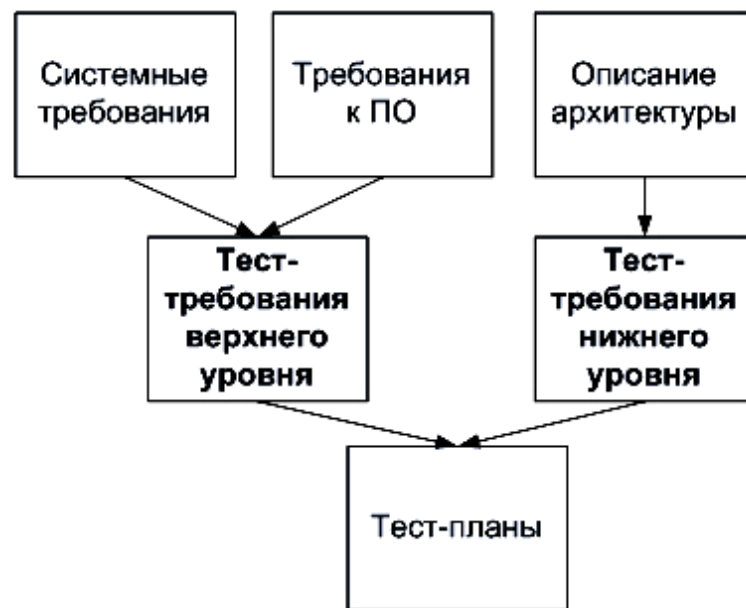


Рисунок 3.2 - Место тест-требований среди проектной документации

Тест-требования должны быть достаточными для построения тест-плана проверки реализации задачи без знакомства с ее программными текстами, т.е. тест-требования должны обладать свойством изоляции от внутренней структуры системы.

Как правило, структура тест-требований следует структуре раздела функциональных требований на систему. Задача каждого требования - определение того, что надо проверить. Обычный формат описания отдельного требования следующий:

Проверить, что при *<описание внешнего воздействия>* [происходит] *<описание реакции программы>*.

Тест-требования, написанные в рамках функционального подхода, обычно разделяют на следующие группы:

- функции контроля входных данных;
- функции обработки ошибок (ввода, вычислений);
- функции получения основного результата;
- функции обработки особых ситуаций;
- функции оформления и вывода результатов.

Совокупность тест-требований должна обладать некоторыми важными свойствами: полнота, верифицируемость и непротиворечивость.

Как правило, одному системному или функциональному требованию соответствует минимум одно тест-требование.

Если совокупность проверок, задаваемых тест-требованиями, покрывает всю функциональность системы, определенную в системных требованиях и требованиях к программному обеспечению, то говорят о полноте тест-требований.

При изменениях требований к системе для поддержания полноты должны меняться и тест-требования.

Как системные требования и требования к ПО, так и тест-требования должны обладать свойством верифицируемости. Т.е. для каждого требования должна существовать возможность определить четкий критерий проверки - выполняется это требование в реализованной системе или нет.

Примером не верифицируемого требования может служить следующее «требование»:

Система должна иметь интуитивно понятный пользовательский интерфейс.

Очевидное «тест-требование» будет выглядеть как:

Проверить, что система имеет интуитивно понятный пользовательский интерфейс.

Без четкого определения критериев интуитивной понятности проверить такое требование при помощи написания тестовых примеров не представляется возможным. Однако, если сопровождать такое требование количественными или качественными характеристиками интуитивно понятного интерфейса - написание тестовых примеров по требованиям становится возможным. Так, среди критериев интуитивной понятности могут быть следующие: глубина вложенности меню не более трех, наличие всплывающих подсказок на каждом элементе управления каждой экранной формы и т.п.

Тест-требования по которым составляются тест-планы для тестирования системы, обычно обладают свойством непротиворечивости, поскольку противоречия обычно устраняются на уровне верификации проектной документации. Однако противоречия могут быть выявлены и позже, в результате попытки создать адекватные тестовые примеры.



4 Тест-планы

На основании тест-требований составляются тест-планы - программы испытаний (проверки, тестирования) программной реализации системы. В отличие от тест-требований в тест-плане описываются конкретные способы проверки функциональности системы, т.е. то, как должна проверяться функциональность. Тест-план состоит из отдельных тестовых примеров, каждый из которых проверяет некоторую функцию или набор функций системы.

Для каждого тестового примера однозначно определяется критерий успешного прохождения (pass/fail criteria), при помощи которого можно судить - соответствует ли поведение системы заданному в требованиях или нет (рисунок 3.3).

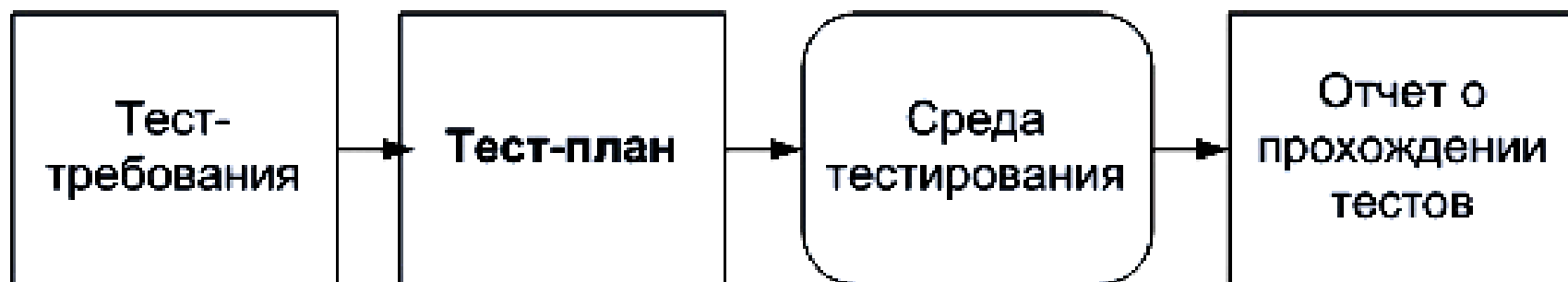


Рисунок 3.3 - Место тест-планов среди проектной документации

Критерием качества тест-плана является покрытие (выполнение) всех требований к проверке правильности функционирования программной реализации. Желательной характеристикой тест-плана является проверка исполнения всех веток схемы программной реализации.

Структура тест-плана может соответствовать структуре тест-требований или следовать логике внешнего поведения системы.



Каждый пункт тест-плана описывает, как производится проверка правильности функционирования программной реализации, и содержит:

- ссылку на требование(я), которое проверяется этим пунктом;
- конкретное входное воздействие на программу (значения входных данных);
- ожидаемую реакцию программы (тексты сообщений, значения результатов)
- описание последовательности действий, необходимых для выполнения пунктов тест-плана.

В состав тест-плана рекомендуется дополнительно включать пункты, которые служат для проверки ветвей программы, не выполнявшихся при проверке удовлетворения функциональных требований. Такие пункты тест-плана могут иметь указание "Для полноты покрытия" в поле ссылки.

Тест-план может готовиться в формализованной форме и служить входным документом для тестовой оснастки, по которому тесты будут выполняться в автоматическом режиме с автоматической фиксацией результатов. В случае, если тест-план готовится в виде текстового документа, возможно только ручное тестирование системы по данному тест-плану.

Форма представления тест-плана в первую очередь зависит от того, каким образом тест-план будет использоваться в процессе тестирования. При ручном тестировании удобно представление тест-планов в виде текстовых документов, в которых отдельные разделы представляют собой описания тестовых примеров. Каждый тестовый пример в таком случае включает в себя перечисление последовательности действий, которые необходимо выполнить тестировщику для проведения тестирования - сценария теста, а также ожидаемые отклики системы на эти действия. Такая форма представления тест-плана неудобна для автоматизации тестирования, поскольку описания на естественном языке практически не поддаются формализации.

Для автоматизированного тестирования сценарий теста может записываться на каком-либо формальном языке, в этом случае возможно непосредственное использование тест-планов как входных данных для среды тестирования.

Другой формой представления тест-планов является таблица. Эта форма наиболее часто используется при четко и формально определенных входных потоках данных системы. Например, каждый столбец таблицы может представлять собой тестовый пример, каждая строка - описание входного потока данных, а в ячейке таблицы записывается передаваемое в данном тестовом примере в данный поток значение. Ожидаемые значения для данного теста записываются в аналогичной таблице, в которой в строках перечисляются выходные потоки данных.

И, наконец, третьей формой представления тестовых примеров является определение примеров в виде конечного автомата. Такая форма представления используется при тестировании протоколов связи или программных модулей, взаимодействие которых с внешним миром производится при помощи обмена сообщениями по заранее заданному интерфейсу. Модуль при этом может быть представлен как конечный автомат с набором состояний, а тест-план будет состоять из двух частей - описания переходов между состояниями и их параметров и тестовых примеров, в которых задается маршрут перехода между состояниями, параметры переходов и ожидаемые значения.

Контрольные вопросы для СРС:

- 1. Какую роль играет проектная документация в процессе верификации программного обеспечения?** Перечислите основные виды проектной документации и объясните, какие из них используются на различных этапах верификации.
- 2. Что представляет собой стратегия верификации программного обеспечения?** Опишите ее цели, основные разделы и объясните, чем стратегия верификации отличается от плана верификации.
- 3. Что такое тест-требования (Test Requirements)?** Опишите порядок их разработки, источники формирования и взаимосвязь с функциональными и нефункциональными требованиями программного обеспечения.
- 4. Какие разделы включает тест-план (Test Plan)?** Разработайте пример структуры тест-плана для программного модуля, указав цели тестирования, объекты проверки, виды тестирования, необходимые ресурсы, критерии начала и завершения тестирования.



Список рекомендуемой литературы

1. В.В. Кулямин. Методы верификации программного обеспечения. - М.: Институт системного программирования РАН, 2019. – 211 с.
2. В.В. Липаев. Программная инженерия. – М.: Гос. ун-т - Высшая школа экономики. – ТЕИС, 2016. – 608 с.
3. С.В. Сеницын. Верификация программного обеспечения: учебное пособие. – Саратов: Профобразование, 2019. — 368 с.



Лекция окончена.

Спасибо за внимание!