

Министерство образования Республики Беларусь  
Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий  
Кафедра технологий программирования

**Методические указания**  
к выполнению лабораторной работы №9

по дисциплине «**Надёжность программного обеспечения**»  
для специальности 1-40 01 01 Программное обеспечение информационных  
технологий

на тему «**План тестирования**»

Новополоцк, 2017 г.

**Название:** «План тестирования».

**Цель работы:** ознакомиться с основами планирования тестирования. Научиться составлять план тестирования к программному продукту.

### **Теоретическая часть**

Основной принцип быстрого тестирования предусматривает максимально тесную связь с разработкой на протяжении всего жизненного цикла программного продукта. Во время выявления и анализа требований подобное интегрирование достигается за счет использования технологии статического тестирования. Данная технология предусматривает пересмотр требований с целью обнаружения дефектов, а также участие специалистов по тестированию в процессе формулирования требований с целью более раннего получения спецификаций, что, в свою очередь, позволяет задействовать требования во время планирования испытаний. Принцип интегрирования разработки и тестовых действий должен соблюдаться и в процессе планирования и определения трудозатрат. Тестирование программного обеспечения представляет собой дорогостоящую, ресурсоемкую работу, на выполнение которой расходуется до половины сметной стоимости проекта. Высокая эффективность планирования испытаний и правильная оценка трудозатрат в значительной мере содействует успеху всего процесса тестирования, в то время как неудачи на этой стадии могут привести к превышению сметной стоимости проекта и нарушению графика работ.

Диаграмма видов деятельности, связанных с планированием испытаний, показана на рисунке 1. Входными для процесса планирования будут документы, которые содержат требования. Если документы с формулировками требований отсутствуют, придется самостоятельно составить, по меньшей мере, сокращенный вариант таких документов. Невозможно провести необходимое тестирование программного продукта, не имея представлений о функциональных возможностях продукта и связанных с ними ожиданий заказчика.

Выходным результатом действий исполнителей, осуществляющих планирование тестирования, является документ или набор документов, который должен быть проверен тестовой группой, группой разработчиков и персоналом, осуществляющим управление разработкой и сопровождением программ. В плане проведения испытаний указаны ресурсы, необходимые для тестирования программного продукта, определено, что подлежит тестированию, как должно проводиться тестирование и какие выходы или выходные результаты будут получены по итогам тестирования.



**Рисунок 1 – Виды деятельности, осуществляемые при составлении плана испытаний**

Процесс планирования испытаний образуется из следующих основных видов деятельности:

- определение стратегии тестирования;
- определение состава и структуры испытательной системы (аппаратных и программных средств);
- оценка трудозатрат (ресурсы и график работ);
- оценка рисков невыполнения графика работ и подготовка плана мероприятий по смягчению последствий от овеществления этих рисков;
- подготовка и пересмотр (утверждение) документов с планом проведения испытаний.

Планирование тестирования можно сравнить с луковицей, т.е. это многоуровневый процесс. Планирование начинается с определения стратегии тестирования на концептуальном уровне, после чего добавляются уровни дальнейшей детализации, которые описывают архитектуру тестирования, условия испытаний, а также тестовые случаи до тех пор, пока не будет составлен план проведения испытаний в его окончательном виде. После того, как план проведения испытаний будет оформлен в виде документа и утвержден, процесс планирования не прекращается, поскольку возможны изменения требований, изменения в графике работ и другие виды изменений, которые влекут за собой коррекцию плана проведения испытаний. План проведения испытаний должен поддерживаться на всем протяжении процесса разработки программного продукта как динамически

развивающийся документ. Он должен храниться в безопасном хранилище и поддаваться управлению версиями.

### ***Стратегия тестирования***

Первое действие в планировании испытаний предусматривает разработку стратегии тестирования на высоком уровне. В общем случае стратегия тестирования должна определять объемы тестовых работ, типы методик тестирования, которые должны применяться для обнаружения дефектов, процедуры, уведомляющие об обнаружении и устраняющие дефекты, критерии входа и выхода из испытаний, которые управляют различными видами тестирования. Реализуя принцип тесного интегрирования разработки и тестирования с целью оптимизации графика разработки, стратегия тестирования должна отображать различные виды тестовой деятельности на жизненный цикл разработки. При формулировании общей стратегии должно быть предусмотрено как статическое, так и динамическое тестирование.

Если для поддержки различных видов тестовой деятельности используется автоматизация, стратегия автоматизации должна рассматриваться как составная часть общей стратегии тестирования. Автоматизация требует выполнения независимых параллельных работ, которые должны тщательно планироваться и выполняться только в тех случаях, когда это не приводит к снижению эффективности.

Предлагается следующий подход к формулированию стратегии тестирования:

- 1 Определить объемы тестовых работ путем анализа документов, содержащих требования к программному продукту (технические условия), дабы выяснить, что нужно тестировать. Рассмотреть виды тестирования, которые не следуют непосредственно из документов с требованиями, такие как тестирование возможности установки и наращивания возможностей программного продукта, удобство и простота обслуживания продукта, а также способности к взаимодействию с другими видами аппаратных средств из среды заказчика.

- 2 Определить подход к тестированию за счет выбора статических и динамических тестов, связанных с каждой стадией разработки. Здесь потребуется включить описания всех рабочих продуктов, которые должна подготовить тестовая группа.

- 3 Определить критерии входа и выхода для каждой стадии тестирования, равно как и все точки контроля качества, для чего потребуется участие специалистов по тестированию.

- 4 Определить стратегию автоматизации в случае, если планируется использование автоматизации какого-либо вида тестовой деятельности. Автоматизация требует проведения независимых параллельных работ, которые должны тщательно планироваться и выполняться только в тех случаях, когда это не приводит к снижению эффективности.

Поскольку подвергнуть тестированию абсолютно все невозможно, важность выбора того, что нужно протестировать, сомнений не вызывает.

Если допустить «перебор» в тестировании, т.е., если тестовое покрытие будет избыточным, то для отладки программного продукта потребуется значительное время, что поставит под угрозу срок сдачи проекта. Если тестирование окажется недостаточным (точнее, недостаточным будет тестовое покрытие), то увеличится риск пропуска того или иного дефекта, устранение которого будет стоить очень дорого, особенно после сдачи программного продукта в эксплуатацию. Отыскать нужный баланс между этими двумя крайностями поможет опыт и способ измерения успешности тестирования.

Вот несколько предложений по разработке стратегии тестирования, которые помогут в поиске оптимального тестового покрытия.

**Тестируйте в первую очередь требования с наивысшим приоритетом.** Предположим, что в вашем распоряжении имеется документ определения требований, в котором требованиям присвоены приоритеты. Выберите те из них, которые представляют для заказчика наибольшую важность, либо которые причинят заказчику наибольшие неприятности в случае выхода программного продукта из строя. Если запланировано тестирование всех требований, и ресурсы это позволяют, естественно, придется тщательно проверить выполнение всех требований. В случае нехватки ресурсов, перед отправкой продукта заказчику необходимо тщательно протестировать требования с наивысшими приоритетами. Возможно, стоит получить согласие заказчика на то, что требования, которые проверены частично или не проверены вообще, не будут поддерживаться вплоть до следующей версии продукта.

**Тестируйте новые функциональные возможности и программный код, который изменялся с целью исправления или совершенствования старых функциональных средств.** Эвристическое правило гласит: если программный код подвергался исправлениям, его необходимо протестировать. В начальной версии программного продукта новым является все. Однако если версия является очередным обновлением или эксплуатационной версией, особое внимание следует уделить новому коду. Имейте в виду, что любые изменения, внесенные в программный код, могут исказить даже те части программы, которые непосредственно «не затрагиваются» изменениями. В этом случае лучше всего как можно чаще выполнять регрессионные тесты для всей функциональности программы, какими бы ни были изменения в коде.

**Используйте разбиение на эквивалентные классы и анализ граничных значений для снижения трудозатрат на тестирование.** Обе технологии могут использоваться для выявления максимального тестового покрытия за счет определения поднабора входных значений во время тестирования компонентов ввода/вывода.

**Тестируйте те участки, в которых наиболее вероятно присутствие проблем.** Известная пословица гласит: «ошибки имеют свойство накапливаться». Если обнаружен какой-то дефект, очень часто рядом притаились еще несколько дефектов, ждущих своей очереди быть

обнаруженными. Если во время анализа требований некоторые участки вызывают особое беспокойство, или есть сведения от разработчиков, что часть компонентов породили массу проблем во время модульного тестирования и проверки взаимодействия и функционирования компонентов, упомянутые участки кода необходимо пометить, дабы обратить на них пристальное внимание при системных испытаниях.

**Сосредоточьте свое внимание на функциях и конфигурациях, с которыми наиболее часто будет иметь дело конечный пользователь.** Если в спецификации требований применяется методика случаев использования или же если у вас имеется доступ к функциональному разрезу конечного пользователя (т.е. математическому ожиданию использования каждой функции), вы получаете дополнительный источник информации для установки приоритетов тестирования. Большая часть времени, отведенного на тестирование, должна затрачиваться на проверку наиболее часто используемых конфигураций и операций.

Один из дополнительных подходов придать системному тестированию большую эффективность состоит в том, чтобы убедить разработчиков тщательно выполнять модульное тестирование и проверку взаимодействия и функционирования компонентов, прежде чем передавать программный код тестовой группе. Слишком часто передача тестовой группе той или иной программной конструкции завершается констатацией о невозможности ее установки, или тем, что после установки конструкция не работает. Значительную часть времени на тестирование можно сэкономить, если ввести в действие некоторые критерии приемки, не допускающие передачу тестовой группе неработающих программных кодов.

Объем тестирования должен определяться документами, регламентирующими план проведения испытаний.

### ***Определение подхода к тестированию***

Второй раздел формулировки стратегии тестирования касается определения подхода к тестированию. Построение подхода к тестированию начинается с исследования каждой стадии жизненного цикла разработки с целью отбора тестов статического и динамического тестирования, которые могут быть использованы на соответствующей стадии. При этом не имеет значения, какая модель жизненного цикла разработки используется: каскадная, спиралевидная или модель с итеративными версиями — для отбора эффективных тестов можно исследовать этапы любой перечисленной модели. В качестве примера возьмем каскадную модель и выясним, какие виды тестирования могут для нее использоваться.

**Стадия формулирования требований.** Определите все документы, которые содержат требования и генерируются на данной стадии, а также все документы, содержащие итоги инспекций, которые получены как результат статического тестирования. Если вы составляете план проведения испытаний на стадии формулирования требований, этот шаг может оказать помощь при составлении плана и графика выполнения необходимых инспекций.

Назначьте членов группы тестирования, которые должны принимать участие в проведении инспекций. Выберите хранилище для результатов инспекций.

**Стадия системного проектирования.** Определите все проектные документы, которые составляются на данной стадии и план участия группы тестирования в связанных с этим инспекциях. В документах, формулирующих требования (документы технического задания), указано, *что* должно тестироваться, а документы эскизного проекта помогут получить представление о том, *как* проводить тестирование.

**Стадии тестирования проектов программ, программных кодов, модульного тестирования и комплексных испытаний.** Если группа тестирования принимает участие в статическом или динамическом тестировании любых из указанных выше видов деятельности, потребуются соответствующие планы. Время от времени, группа тестирования прогоняет тесты с целью выявления утечек памяти или проверки цикломатической сложности тестирования на стадиях модульного тестирования и комплексных испытаний. Если это так, то планы таких испытаний должны учитываться в общем плане испытаний. Если ответственность за эти стадии возлагается только на коллектив разработчиков, их роль должна определяться в форме предположения.

**Системные испытания.** Опишите запланированные вами виды тестирования: функциональная проверка, нагрузочные испытания, испытания под нагрузкой, проверка безопасности, проверка наращиваемости, проверка удобства и простоты обслуживания и другие. Сформулируйте высокоуровневые цели для этих видов тестирования и укажите, как проследить их связь с породившими их требованиями и функциональными спецификациями. Если какие-то виды тестирования опускаются, дайте логическое обоснование их пропуска и оцените связанные с этим риски.

**Приемочные испытания.** Дайте описание плана приемочных испытаний, включая альфа-, бета- и другие виды тестирования. Часто бывает полезно составить отдельный план приемочных испытаний, а в нем указать объемы работ и ограничения, накладываемые на тестирование, критерии включений и исключения из испытаний и условия испытаний. Поскольку нередко цель приемочных испытаний состоит не только в том, чтобы продемонстрировать заказчику функциональные возможности программного продукта, но и в обнаружении дефектов, план приемочных испытаний целесообразно рассматривать как документ, направленный извне.

**Регрессионное тестирование.** Укажите в плане регрессионного тестирования, составлен ли он для эксплуатационных версий или же является частью стратегии итеративного выпуска версий. Опишите, как выбираются и разрабатываются регрессионные тесты, какое должно использоваться оборудование и какова стратегия автоматизации регрессионного тестирования. Исходя из предположения, что тестами покрываются не все функциональные возможности, укажите объемы регрессионного тестирования и риски, связанные с тем, что некоторые участки будут пропущены и останутся непокрытыми.

Подход к тестированию должен отражаться в документах, содержащих планы проведения испытаний.

### ***Определение критериев тестирования и точек контроля качества***

Цель определения критериев тестирования заключается в установке на месте заказчика набора правил, регламентирующих поток тестирования. Очень важно определиться с критериями тестирования до начала испытаний. После того, как вы приступили к выполнению плана испытаний, поздно объявлять критерии, инициирующие или останавливающие тестирование. Например, если вы рассчитываете на то, что программный продукт успешно пройдет модульное тестирование и комплексные испытания, вы должны заявить об этом заранее в подготовленном плане испытаний. После этого план должен быть проверен и утвержден людьми, которые будут заниматься модульным тестированием и комплексными испытаниями. Ввод в действие и выполнение критериев тестирования помогает проекту выдерживать намеченные сроки за счет исключения временных потерь, которые будут неизбежны в случае, когда на испытательный стенд попадает программный продукт, не готовый к испытаниям, либо в ситуации, когда продолжаются испытания продукта, который нужно отправлять на переделку.

Поскольку критерии тестирования затрагивают интересы других групп, они не могут односторонне устанавливаться группой тестирования. Применение критериев должно быть согласовано со всеми заинтересованными сторонами. Критерии могут быть установлены для модульного тестирования и комплексных испытаний, однако такие критерии обычно определяются организацией-разработчиком. Рассматриваемые здесь критерии тестирования применяются главным образом в отношении системного тестирования.

Существует пять типов критериев, которые могут определяться перед началом системного тестирования.

**Критерий входа** описывает, что нужно сделать перед началом тестирования. Например, возможно, потребуется располагать документами технического задания в окончательной форме. Можно поставить задачу, чтобы программный продукт был упакован в том же виде, в каком он будет поставляться заказчику. Во время тестирования может возникнуть необходимость в служебных программах, конфигурационных файлы или данных, которыми будет пользоваться заказчик. Если на вас возложена ответственность за проверку документации, сопровождающей программный продукт, она также должна быть доступна во время тестирования. Один из способов выполнения критерия входа предусматривает проверку готовности к испытаниям. Эта проверка использует контрольную таблицу элементов действий, которые другие группы согласились передать как входные данные для тестирования.

**Критерий выхода** описывает то, что вы считаете необходимым для завершения испытаний. Несмотря на то что группы тестирования часто пытаются сделать критерий выхода условием поставки программного продукта, это нереально. Решение на поставку принимается и должно



приниматься высшим руководством разработок. Критерий выхода из испытаний должен звучать примерно так: «все запланированные тесты выполнены, все исправленные дефекты проверены, уведомления обо всех обнаруженных новых дефектах были выданы. Невыполненные пункты плана, такие как неудачный прогон некоторого набора тестов из-за неисправности оборудования, задокументированы». Как и в случае критерия входа в испытания, допускается проведение проверки готовности, которая обеспечит полное выполнение всех тестов, и оценки готовности программного продукта к поставкам на базе результатов тестирования.

**Критерий приостановки/возобновления** описывает, что произойдет, если по причине из-за дефектов продолжение тестирования окажется невозможным. Другими словами, если дела складываются настолько неудачно, что запланированные испытания провести не удастся, их необходимо прекратить до тех пор, пока обнаруженные дефекты не будут устранены.

**Критерий успешного/неудачного прохождения теста.** Прогон каждого теста должен давать заранее известные результаты. Если получен ожидаемый результат, считается, что продукт успешно прошел тест, в противном случае прохождение теста завершается неудачно. В то же время может случиться так, что прогон некоторой группы тестов не выполняется, поскольку они либо искажены или заблокированы дефектами, либо необходимые для их прогона ресурсы отсутствуют. Целесообразно определить заранее, что делать с тестами, которые не удалось выполнить. Возможно, будет запланировано пометить каждый невыполненный тест буквой «N» в итоговом отчете и объяснить в поле комментария, что случилось и что было предпринято в контексте решения проблемы. Может быть, в ваши планы входит замена искаженных тестов специальным видом тестирования и регистрация результатов в специальном акте об испытаниях.

**Другие критерии, определяемые процессом или стандартами.** Если программный продукт должен соответствовать некоторому стандарту или ваша компания предъявляет определенные требования к выполняемому процессу, скорее всего, придется учесть ряд дополнительных критериев. Например, в вашем локальном процессе могут быть определены конкретные средства, выдающие сообщения об обнаруженных дефектах или отчеты по результатам испытаний.

В дополнение к приведенным выше критериям, полезно определить критерий тестирования и подготовить контрольную таблицу готовности в рамках плана проведения испытаний.

### ***Архитектура тестов***

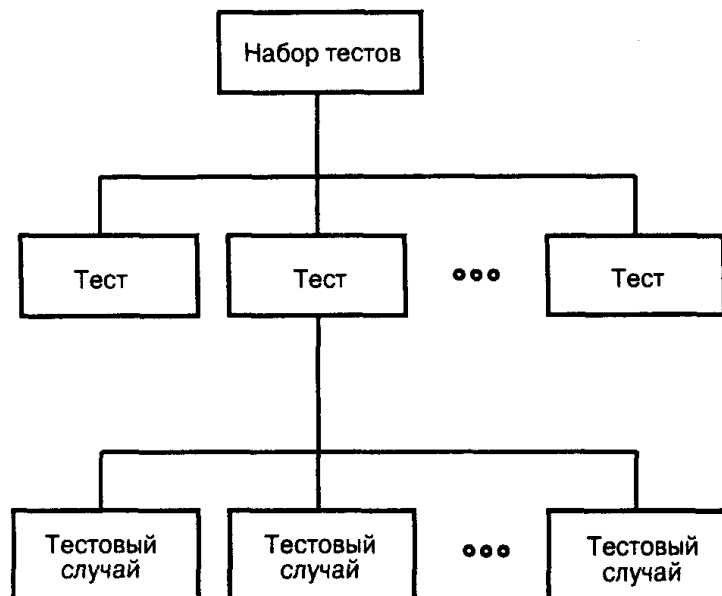
Архитектура тестов имеет отношение к организационной структуре тестовых случаев. На практике целесообразна такая организация тестов, когда каждый тест рассчитан на проверку некоторой четко заданной части функциональных возможностей программного продукта. Если тест закончится неудачей, разработчик легко может повторно прогнать этот тест и воспроизвести обнаруженный дефект, затем проанализировать его и

попытаться выявить причину. В идеальном случае границы обследования теста должны быть сужены, а сам тест должен выполнять некоторый минимальный набор действий, необходимых для того, чтобы вызвать отказ системы.

Если основой проводимых испытаний служат требования к программному продукту, то практическим правилом для определения границ обследования конкретного теста может служить критерий, по условиям которого на каждое такое требование полагается, по меньшей мере, один тест. Помимо того, что это правило определяет эффективные размеры тестов, принцип «по меньшей мере, один тест на требование» упрощает поддержку планирования проведения испытаний. На основе анализа технического задания можно получить представление о том, сколько нужно новых тестов для испытания нового программного продукта. Если составлять тесты, руководствуясь этим принципом, несложно оценить, какое время отымет разработка и прогон тестов, основанных на ранее полученных результатах.

Испытания по описанному выше принципу позволяют систематизировать организацию тестов. Требования обычно объединяются в группы по функциональным признакам. Вы можете организовать свои тесты по такому же принципу. Тесты, осуществляющие проверку готовности к работе технических средств заказчика, могут быть объединены в одну группу; точно так же тесты, проверяющие механизмы установки и наращивания возможностей системы, образуют другую группу. Группа родственных тестов называется *тестовым набором (test suite)*.

Каждое требование может состоять из нескольких компонентов. Например, требование, регламентирующее процедуру редактирование элементов базы данных конкретного заказчика, может определять несколько различных сценариев. *Тестовый случай (test case)* есть набор входных данных тестов, условий выполнения тестов и ожидаемых результатов, который ориентирован на достижение конкретной цели. Тестовый случай представляет собой минимальный тестовый модуль, допускающий независимое выполнение от начала до конца. Отношения, связывающие тестовые наборы, тесты и тестовые случаи, показаны на рисунке 2 и в таблице 1.



**Рисунок 2 – Архитектура тестов**

После того, как удалось определить организационную структуру тестов, возникает необходимость дать описание хранилища для хранения тестов и средств управления версиями для последующего воспроизводства тестов. Это хранилище должно обладать определенными характеристиками:

- специалисты по тестированию, разрабатывающие и осуществляющие прогон тестов, должны иметь простой доступ к хранилищу. Часто в качестве хранилища используется сетевой дисковый накопитель коллективного доступа;

- с этого хранилища регулярно снимаются копии, а полученные резервные копии файлов периодически восстанавливаются с целью проверки правильности функционирования средств копирования и восстановления. Необходимость дублирования всех планов проведения испытаний, тестовых случаев, результатов тестирования очевидна, тем не менее, зачастую такое дублирование игнорируется;

- должно быть налажено управление версиями, что обеспечит простое выявление более старых тестов. Управление версиями при необходимости обеспечивает воспроизведение любого теста, который когда-либо выполнялся на программном продукте.

**Таблица 1 – Иерархия тестов**

Уровень	Определение
Набор тестов	Совокупность тестов, используемых для проверки связанных групп требований и функций.
Тест	Один или больше тестов, которые предназначены для проверки выполнения одного требования или функции.
Тестовый случай	Наименьший тестовый модуль, который может выполняться самостоятельно от начала до конца.

## **Формат плана проведения испытаний**

После выбора стратегии тестирования, определения испытательной системы и оценки трудозатрат можно приступать к составлению плана проведения испытаний. План проведения испытаний представляет собой документ или некоторый набор документов, представляющих собой результат выполнения определенных видов деятельности по планированию испытаний. Этот документ не является раз и навсегда установленным планом, он меняется по мере изменения требований. Полезные рекомендации по разработке плана проведения испытаний предлагает стандарт IEEE Standard 829: *IEEE Standard for Software Test Documentations* (стандарт IEEE по составлению документированию испытаний программного обеспечения). Конечно, можно воспользоваться другим форматом, однако потребуются оформить все материалы, рекомендуемые указанным выше стандартом IEEE в любом подходящем формате. При построении шаблона плана проведения испытаний будет использоваться стандарт IEEE Standard 829.

Согласно рассматриваемому стандарту, план проведения испытаний должен содержать 16 разделов:

- 1 Идентификатор плана проведения испытаний.
- 2 Введение.
- 3 Компоненты, которые должны тестироваться.
- 4 Характеристики и свойства, которые должны тестироваться.
- 5 Характеристики и свойства, которые не должны тестироваться.
- 6 Подход.
- 7 Критерий успешных и неудачных испытаний.
- 8 Критерий приостановки испытаний и требования возобновления испытаний.
- 9 Выходные результаты тестов.
- 10 Задачи тестирования.
- 11 Требования окружающей среды.
- 12 Распределение ответственности.
- 13 Подбор кадров и подготовка персонала.
- 14 График работ.
- 15 Риски и непредвиденные обстоятельства.
- 16 Утверждение плана проведения испытаний.

Тот факт, что указанный стандарт направлен на разделение планов проведения испытаний, процедур и результатов, становится очевидным из содержащихся в нем определений следующих документов.

**План проведения испытаний.** Это документ описывает возможности, подход, ресурсы и график исполнения различных видов тестовой деятельности. Он определяет объекты тестирования, функции, которые должны тестироваться, тестовые задания, исполнителей каждого тестового задания, и любой из рисков, требующий планирования на случай непредвиденных обстоятельств.

**Спецификация методики тестирования.** Этот документ описывает последовательность действий при выполнении конкретного теста.

**Отчетный доклад о тестировании.** Этот документ обобщает виды тестовой деятельности и их результаты. Он также содержит оценку соответствующих тестовых объектов.

Проведем анализ содержимого плана проведения испытаний и оптимизацию его структуры для целей быстрого тестирования.

**1 Идентификатор плана проведения испытаний.** В долгосрочной перспективе можно сэкономить время, если присвоить всем тестовым документам идентификаторы, которые нетрудно отслеживать. Наличие уникального идентификатора каждого тестового документа позволяет воспрепятствовать тому, что кто-либо из исполнителей напрасно затратит время на изучение не того документа, что надо. Таким идентификатором может быть алфавитно-цифровая строка, которая может содержать данные, отображающие название продукта, номер версии, дату и любую другую информацию, позволяющую отслеживать этот документ. Рекомендуется отдавать предпочтение идентификаторам, которые вписываются в общую систему управления документами.

**2 Введение.** Цель введения заключается в том, чтобы дать краткую информацию для любого исполнителя, желающего воспользоваться планом проведения испытаний. Суть состоит в необходимости указания во вводном разделе плана проведения испытаний исходных документов, служащих входными данными выполняемого тестирования.

**3 Компоненты, которые должны тестироваться.** Здесь приводятся высокоуровневые списки компонент, которые вы намерены тестировать. В этом разделе будут упомянуты некоторые из них:

- идентификатор выпуска/версии тестируемого программного проекта;
- исправления дефектов, обнаруженных в предыдущей версии. Если список дефектов приводится в спецификации функций, необходимо указать ссылку на спецификации функций и не дублировать их здесь;
- описание среды, используемой для распределения программного продукта (например, компакт-диски, Web- или ftp-сайты);
- документы для конечного пользователя, такие как руководство пользователя, инструкции по установке, примечания по версии продукта.

Обратите внимание на тот факт, что это перечень представляет собой не что иное, как входные данные для выполняемого тестирования.

**4 Характеристики и свойства, которые должны тестироваться.** Стандарт IEEE Standard 829 предлагает следующее определение *свойства программного обеспечения*: это отличительная характеристика программного компонента, например, производительность, переносимость или функциональные возможности.

Это очень общее определение, поскольку оно должно охватить большое количество понятий. Возможно, имеет смысл давать определения свойств в бизнес-терминах. Рассмотрим свойство быть тем, «что может продаваться заказчику». Например, если приложение работает медленнее, чем это нужно заказчику, то увеличение быстродействия его наиболее часто

используемых функций может подаваться как свойство. Более совершенный новый способ ввода ускорения ввода данных или запроса в базу данных также может подаваться как свойство.

Ключевым моментом для специалистов по тестированию является то, что если что-то было обещано заказчику, оно должно быть отражено в плане проведения испытаний с тем, чтобы протестировать его до поставки объекта заказчику. Основным источником, позволяющим узнать, что было обещано заказчику, служат документы определения требований, на которые существуют ссылки во введении.

Перечень свойств в этом разделе предоставляет в ваше распоряжение контрольную таблицу для вычисления трудозатрат, необходимых для проведения испытаний программного продукта. Каждая позиция этого списка соответствует конкретным пунктам документов описания требований, в то же время каждой такой позиции должен соответствовать один или большее число тестов, определенных в спецификации методики тестирования.

### **5 Характеристики и свойства, которые не должны тестироваться.**

Этот раздел предназначен для того, чтобы заблаговременно и однозначно определить, какие объекты не должны охватываться тестированием. В число объектов, объявляемых как «не подлежащие тестированию», могут входить следующие:

- функции, реализация которых отложена до последующих инкрементальных версий продукта;

- конфигурации компьютерных средств, которые невозможно проверить по причине отсутствия необходимого оборудования. Иногда используются настолько дорогостоящие конфигурации компьютерного оборудования, что их невозможно продублировать в условиях испытательных лабораторий. Если вы не можете воспроизвести операционную среду, но в то же время можете ее смоделировать, об этом потребуется сообщить в разделе «Подход». Если вы вообще не можете проводить испытания в операционной среде, этот вопрос следует рассмотреть в разделе, в котором обсуждаются «риски»;

- комбинации настроек или конфигурации аппаратных средств, которые не могут быть испытаны за время, выделенное для выхода программного продукта на рынок. Если принято решение ограничить объем тестирования некоторого конкретного свойства, необходимо обсудить данные ограничения и обосновать их в этом разделе.

На завершающих стадиях проекта, когда, как правило, на группу тестирования оказывают давление в плане поторопиться с окончанием работ, очень полезно иметь в своем распоряжении документ, в котором оговаривается, что планируется тестировать, а что нет. В пылу предфинишной лихорадки люди склонны забывать, какие соглашения были достигнуты – совсем неплохо опереться на этот раздел плана проведения испытаний при ответах на вопросы, почему то или иное свойство не охвачено тестированием. Разумеется, если проблема возникнет в области, которая не

подвергалась тестированию в вашей лаборатории, она должна быть устранена и проверена после выпуска очередной версии программного продукта. Поскольку стоимость исправления дефектов после сдачи программного продукта в эксплуатации очень велика, важно получить по возможности точную оценку рисков, прежде чем принимать решение не проводить испытаний того или иного свойства или конфигурации.

**6 Подход.** Этот раздел плана проведения испытаний предназначен для описания на высоком уровне, как вы намерены проводить испытания программного продукта. Это описание не является подробной спецификацией всех методик испытаний, которые планируется использовать. В этот раздел можно включить, например, такие темы:

- статическое тестирование требований и проектная документация;
- статическое и динамическое тестирование, которое должно проводиться на стадиях тестирования программных кодов, модулей и проверки взаимодействия и функционирования компонентов системы;
- тестирование свойств;
- испытания при перегрузках/испытания под нагрузкой/тестирование производительности;
- проверка средств защиты;
- тестирование установки/обновления программного продукта;
- тестирование средств дублирования/восстановления;
- тестирование GUI-интерфейса;
- регрессивное тестирование;
- приемочные испытания: альфа-, бета- и другие виды испытаний на месте;
- проверка результатов устранения дефектов;
- прерывание испытаний, проводимых в автоматизированном и неавтоматизированном режимах;
- виды тестирования, проведение которых поручается сторонним организациям;
- использование системы отслеживания дефектов для ввода сообщений о неисправностях.

При выполнении тестирования инкрементальных версий программного продукта подход к тестированию каждой версии, по всей видимости, будет одним и тем же. Вы должны обладать возможностью экономить время за счет заимствования больших фрагментов того же раздела плана проведения испытаний, составленного для предыдущих версий, но в то же время тщательно анализировать последствия любых изменений, внесенных в текущую версию программного продукта.

**7 Критерии успешного/неудачного прохождения испытаний.** Два критерия успешного/неудачного прохождения испытаний заслуживают особого внимания при тестировании. Первый относится к индивидуальным тестам. Все тестовые случаи должны снабжаться наборами ожидаемых результатов; по существу, если получены ожидаемые результаты, итог

прогона теста считается успешным, если нет – результат прогона теста расценивается как неудачный. Критерии успешного/неудачного прохождения отдельных тестов должны быть включены в сами тестовые случаи, в связи с чем их не следует помещать в план проведения испытаний. Однако не лишним будет указать в этом плане, что определение критериев успешного/неудачного прохождения испытаний является неотъемлемой частью тестового случая.

Второй тип критериев успешного/неудачного прохождения испытаний имеет отношение к тестированию всего программного продукта. С самого начала вы должны определить, что следует считать успешным прохождением стадии тестирования, т.е. когда можно прекратить испытания продукта и наладить его поставку. В некоторых случаях критерий выхода из испытаний определяется в программе испытаний или даже в документе, который содержит формулировки требований, предъявляемых к программному продукту. Вне зависимости от того, что служит источником этих критериев, полезно дать их четкую формулировку в плане проведения испытаний.

**8 Критерий приостановки испытаний и требования возобновления испытаний.** Было установлено, что критерий вхождения в испытание показывает, что необходимо предпринять, прежде чем начнете тестирование, а критерий выхода из испытаний описывает, что, по вашему мнению, требуется для завершения испытаний. Критерии приостановки и возобновления испытаний описывают, что происходит, когда продолжению испытаний препятствуют дефекты. Например, можно объявить, что если модуль содержит устойчивую системную ошибку, исключающую успешную установку программного продукта, тестирование приостанавливается на период времени, необходимый для устранения этой ошибки. Разумеется, нельзя останавливать тестирование каждый раз, когда какая-то часть тестов заблокирована дефектами, однако следует поставить в известность руководство в ситуациях, когда программный продукт содержит такое количество дефектов, что продолжать испытания нецелесообразно.

**9 Выходные результаты тестов.** Этот раздел плана проведения испытаний является тем местом, в котором определяются выходные данные работ по тестированию, например, можно предложить следующий список:

- план проведения испытаний;
- документы, регламентирующие проектирование тестов;
- документ, содержащий спецификацию тестов;
- сообщения о результатах прогона тестов;
- сообщения об обнаруженных дефектах;
- примечания по выпуску программного продукта.

Определение и согласование того, что вы намереваетесь опубликовать по завершении тестирования на раннем этапе жизненного цикла, может стать хорошей практикой, поскольку в этом случае упрощается построение реалистичного плана работ. Дабы не закладывать в план работ неоправданных накладных расходов, нежелательно отображать в плане



результаты работ, которые не существенны для достижения конечной цели, каковой в рассматриваемом случае является максимально быстрая поставка программного продукта.

**10 Задачи тестирования.** Если план проведения испытаний применяется для обоснования произведенной оценки трудозатрат на тестирование, этот раздел является подходящим местом для определения индивидуальных задач, которые необходимы для подготовки и выполнения тестирования. Еще лучше, если вы занесете вашу оценку в динамическую электронную таблицу. Другой способ предполагает использование программы управления проектами, например, Microsoft Project, для того, чтобы в этом разделе плана организовать ссылку на другой документ. Это еще один подходящий момент, чтобы указать ссылку на справочный документ или, по крайней мере, путь к соответствующему файлу или сетевому дисковому накопителю.

**11 Информация о конфигурации средств тестирования (требования окружающей среды).** Дайте описание аппаратных и программных средств, необходимых для проведения испытаний, и начертите диаграммы, показывающие, как аппаратные компоненты соединяются друг с другом. Этот раздел предназначен для того, чтобы позволить любому другому заинтересованному лицу построить такую же испытательную установку в случае необходимости воспроизведения неисправности или проверки, что неисправность устранена. Данный раздел плана проведения испытаний должен составляться на базе анализа, проведенного в разделах «Среда тестирования» и «Конфигурации средств тестирования». Возможно, потребуется расширить рамки этого раздела и включить в него архитектуру тестов (т.е. как организованы тесты) и инструментальные средства тестирования, которые вы намерены использовать.

**12 Распределение ответственности при проведении тестирования.** Если возможности вашей тестовой группы по завершению тестирования каким-то образом зависят от деятельности другой группы, в этом разделе плана проведения тестирования должно быть указано, кто что делает. Например, если вы рассчитываете на разработчиков программного обеспечения в том, что в проводимую ими проверку взаимодействия и функционирования компонентов программного продукта будет включен некоторый конкретный набор тестов, данный раздел является подходящим местом для формулировки соответствующих условий. Возможно, в этом испытании намерены принять участие инженеры, обслуживающие заказчика, занявшись тестированием пользовательского интерфейса или проверкой документации.

В этом разделе не указываются служебные обязанности конкретных инженеров и техников; для этой цели предназначен следующий раздел плана проведения испытаний.

**13 Подбор кадров и подготовка персонала.** В этом разделе можно привести список персонала, который вы намерены привлечь к тестированию программного продукта. Одна из причин включения такого списка связана с

тем, что своевременное выполнение ключевых этапов тестирования зависит от возможности участия этих людей на стадии тестирования. Если кто-либо из этих специалистов был переброшен на другой проект или вообще уволился, у вас появляются серьезные основания для привлечения других исполнителей или смещения сроков тестирования.

**14 График работ.** Этот раздел обычно не является местом, в котором оглашается подробный график проектных работ – для таких целей существует общий план проектных работ или самостоятельный файл, который может подвергаться изменениям независимо от плана проведения испытаний. В конце концов, подробный план работ может быть очень динамичным документом, однако содержимое того, что вы тестируете, кадровые проблемы, рабочие продукты тестирования и другие компоненты плана проведения испытаний остаются достаточно статичными.

Скорее всего, в план проведения испытаний потребуется включить краткий список ключевых этапов тестирования, например, срок начала тестовых работ, когда бета-версия продукта должна поступить к первым пользователям и когда завершатся испытания. С другой стороны, можно просто принять решение обращаться к планам проектных работ или самостоятельному плану работ, используя для этих целей ссылку на Web-странице или путь к сетевому дисковому накопителю.

**15 Риски и непредвиденные обстоятельства.** Риски могут описываться в таблице, в которой указываются риски, вероятность осуществления этого риска, влияние этого риска и план мероприятий по снижению от овеществления рисков.

**16 Утверждение плана проведения испытаний.** Для решения этой задачи на титульном листе плана проведения тестовых работ потребуется составить список лиц, утверждающих план, и получить их одобрение. Можно также отправить утверждающему лицу по электронной почте копию плана проведения испытаний и получить ответ с утвержденным планом. Хранить сообщения и ответы электронной почты рекомендуется в совместно используемом сетевом каталоге, который дублируется через регулярные промежутки времени, либо присоединить все ответы и подтверждения к специальному приложению к плану проведения испытаний.

## Практическая часть

### *Содержание задания*

Составить план тестирования для программного продукта из предыдущей лабораторной работы. Пример плана тестирования приведен в книге «Быстрое тестирование» страницы 313 – 327 (книга прикреплена к заданию вместе с методическими указаниями).

### *Порядок выполнения работы*

- 1 Ознакомиться с теоретической частью.
- 2 Составить план тестирования (**обязательными являются ВСЕ пункты указанные ниже в содержании документа**).
- 3 Показать работу преподавателю.
- 4 После защиты, выполненного задания, скинуть отчёт преподавателю (в названии указать **фамилию и номер лабораторной**, например, Ivanov10).

### *Содержание документа*

- 1 Титульный лист (**на титульном листе вместо варианта указать название тестируемого программного продукта**).
  - 2 Введение.
  - 3 Тестируемые элементы.
  - 4 Характеристики и свойства, которые должны тестироваться.
  - 5 Характеристики и свойства, которые не должны тестироваться.
  - 6 Применяемый подход.
  - 7 Критерий успешных и неудачных испытаний.
  - 8 Критерий приостановки испытаний и требования возобновления испытаний.
  - 9 Выходные результаты тестов.
  - 10 Задачи тестирования.
  - 11 График работ.
  - 12 Риски и непредвиденные обстоятельства (**указать не менее 2**).
  - 13 Вывод о проделанной работе.
- Защита работ проводится индивидуально.

### *Контрольные вопросы*

- 1 Перечислите виды деятельности, осуществляемые при составлении плана испытаний.
- 2 Что является выходным результатом действий исполнителей, осуществляющих тестирование?
- 3 Для чего нужен критерий входа?
- 4 Опишите критерий успешного/неудачного прохождения теста.
- 5 Что такое тестовый случай?
- 6 Что такое тестовый набор?
- 7 Перечислите разделы плана проведения испытаний.