# Лабораторная работа №2 Виды тестирования. Планирование тестирования

Цель: изучить классификацию видов тестирования, разработать проверки для различных видов тестирования, научиться планировать тестовые активности в зависимости от особенностей поставляемой на тестирование функциональности.

*Практическое задание:*

1. Выбрать объект реального мира (например, карандаш, стол, чашка, клавиатура, сумка и др.) с целью последующей разработки тестовых проверок для него.
2. Разработать различные проверки в соответствии с классификацией видов тестирования для выбранного объекта реального мира. Результаты внести в таблицу 1.1.

Таблица 1.1 – Тестировые проверки для различных видов тестирования

|  |  |  |
| --- | --- | --- |
| **Объект тестирования:** *Карандаш* | | |
| **Вид тестирования** | **Краткое определение вида тестирования** | **Тестовые проверки** |
| Functional Testing | Тестирование, основанное на сравнительном анализе спецификации и функциональности компонента или системы. | Написать текст |
| Safety Testing | Тестирование программного продукта с целью определить его способность при использовании оговоренным образом оставаться в рамках приемлемого риска причинения вреда здоровью, бизнесу, программам, собственности или окружающей среде. | Уронить карандаш |
| Security Testing | Тестирование с целью оценить защищенность программного продукта от внешних воздействий (от проникновений). На практике зачастую под термином тестирование безопасности понимают в том числе и тестирование защищенности. | Попробовать согнуть/сломать |
| Compatibility Testing | Проверка работоспособности приложения в различных средах. Виды тестирования совместимости: кроссбраузерное тестирование кроссплатформенное тестирование. | Бросить в воду, в огонь |
| GUI Testing | Тестирование, выполняемое путем взаимодействия с системой через графический интерфейс пользователя | Проверить грифель на соответствие маркировки твёрдости |
| Usability Testing | Тестирование с целью определения степени понятности, легкости в изучении и использовании, привлекательности программного продукта для пользователя при условии использования в заданных условиях эксплуатации | Попробовать написать диктант |
| Accessibility Testing | Тестирование, которое определяет степень легкости, с которой пользователи с ограниченными способностями могут использовать систему или ее компоненты. | Дать попользоваться карандашом инвалидам с различными недугами |
| Internationalization Testing | Тестирование адаптации продукта к языковым и культурным особенностям целого ряда регионов, в которых потенциально может использоваться продукт. | Проверить язык надписи на карандаше и упаковке |
| Performance Testing | Процесс тестирования с целью определения производительности программного продукта. | В зависимости от маркировки проверить на долгосрочность использования |
| Stress Testing | Вид тестирования производительности, оценивающий систему или компонент на граничных значениях рабочих нагрузок или за их пределами, или же в состоянии ограниченных ресурсов, таких как память или доступ к серверу. | Писать карандашом, прилагая на дюбку грифеля чрезмерные усилия.  Писать слишком слабо. |
| Negative Testing | Полное тестирование системы или ее части  на некорректных данных/сценариях | Попробовать написать обратной (незаточенной) стороной карандаша, поохотиться на кролика, нарезать хлеб |
| Black Box Testing | Тестирование системы без знания внутренней структуры и компонентов системы | Дать обезьяне карандаш |
| Automated Testing | Набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования. | Дать карандаш роботу, вставить карандаш в сейсмограф |
| Unit/Component Testing | Тестируются отдельные части (модули) системы. | Проверить, как оболочка карандаша поддаётся заточка |
| Integration Testing | Тестируется взаимодействие между отдельными модулями. | Проверить, не выпадает ли грифель из оболочки |

1. Разработать композицию тестов для первой поставки программного обеспечения (build 1), состоящей из трех модулей (модуль 1, модуль 2, модуль 3).
2. Разработать композицию тестов для второй поставки программного обеспечения (build 2): исправлены заведенные дефекты, доставлена новая функциональность – модуль 4.
3. Разработать композицию тестов для третьей поставки программного обеспечения (build 3): заказчик решил расширять рынки сбыта и просит осуществить поддержку программного обеспечения на английском языке.
4. Разработать композицию тестов для четвертой поставки программного обеспечения (build 4): заказчик хочет убедиться, что программное обеспечение выдержит нагрузку в 2000 пользователей.
5. Оформить отчет и защитить лабораторную работу.

Содержание отчета:

1. Цель работы.
2. Разработанные проверки выбранного объекта реального мира для различных видов тестирования.
3. Тестовые активности для сформулированных задач.
4. Выводы по работе.

Контрольные вопросы:

1. Что такое тестирование?

Тестирование (Testing) – процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта.

1. Что такое качество программного обеспечения?

Качество (Quality) – степень, с которой компонент, система или процесс соответствует зафиксированным требованиям и/или ожиданиям и нуждам пользователя или заказчика [3].

1. Что такое дефект?

Дефект (defect, bug, ошибка) – ключевой термин тестирования, означающий отклонение фактического результата от ожидаемого.

1. Назовите три условия обнаружения дефекта.

Для обнаружения дефекта необходимо выполнить три условия: знать фактический результат, знать ожидаемый результат, зафиксировать факт разницы между фактическим и ожидаемым результатом.

1. Какие существуют виды тестирования в зависимости от объекта тестирования? Дайте характеристику каждому.

Виды тестирования в зависимости от объекта тестирования:

* Функциональное – тестирование в целях проверки реализуемости функциональных требований.
  + Функциональное тестирование
  + Тестирование безопасности
  + Тестирование защищенности
* Пограничное – тестирование продукта на крайних значениях / ситуациях.
  + Тестирование совместимости
  + Кроссбраузерное тестирование
  + Кроссплатформенное
* Нефункциональные – тестирование свойств, которые не относятся к функциональности системы.
  + Тестирование требований
  + Тестирование прототипа
  + Тестирование пользовательского интерфейса
  + Тестирование удобства использования
  + Тестирование доступности
  + Тестирование интернационализации
  + Тестирование локализации
  + Тестирование производительности
  + Нагрузочное тестирование
  + Объемное тестирование
  + Тестирование стабильности и надежности
  + Стрессовое тестирование
  + Тестирование на отказ и восстановление
  + Тестирование установки и лицензирования

1. Какие существуют виды функционального тестирования? Дайте характеристику каждому.

Виды функционального тестирования:

* + 1. Функциональное тестирование - тестирование, основанное на сравнительном анализе спецификации и функциональности компонента или системы.
    2. Тестирование безопасности - тестирование программного продукта с целью определить его способность при использовании оговоренным образом оставаться в рамках приемлемого риска причинения вреда здоровью, бизнесу, программам, собственности или окружающей среде.
    3. Тестирование защищенности - вид тестирования производительности, оценивающий систему или компонент на граничных значениях рабочих нагрузок или за их пределами, или же в состоянии ограниченных ресурсов, таких как память или доступ к серверу.

1. Какие существуют виды нефункционального тестирования? Дайте характеристику каждому.

Тестирование требований - проверка требований на соответствие основным атрибутам качества.

Тестирование прототипа - метод выявления структурных, логических ошибок и ошибок проектирования на ранней стадии развития продукта до начала фактической разработки.

Тестирование пользовательского интерфейса - тестирование, выполняемое путем взаимодействия с системой через графический интерфейс пользователя.

Тестирование удобства использования - тестирование с целью определения степени понятности, легкости в изучении и использовании, привлекательности программного продукта для пользователя при условии использования в заданных условиях эксплуатации.

Тестирование доступности - тестирование, которое определяет степень легкости, с которой пользователи с ограниченными способностями могут использовать систему или ее компоненты.

Тестирование интернационализации - тестирование адаптации продукта к языковым и культурным особенностям целого ряда регионов, в которых потенциально может использоваться продукт.

Тестирование локализации - тестирование адаптации продукта к языковым и культурным особенностям конкретного региона, отличного от того, в котором разрабатывался продукт.

Тестирование производительности - процесс тестирования с целью определения производительности программного продукта. В рамках тестирования производительности выделяют нагрузочное теситрование, объемное тестирование, тестирование стабильности и надежности, стрессовое тестирование.

Нагрузочное тестирование - вид тестирования производительности, проводимый с целью оценки поведения компонента или системы при возрастающей нагрузке.

Объемное тестирование - позволяет получить оценку производительности при увеличении объемов данных в базе данных приложения

Тестирование стабильности и надежности - позволяет проверять работоспособность приложения при длительном (многочасовом) тестировании со средним уровнем нагрузки.

Стрессовое тестирование - вид тестирования производительности, оценивающий систему или компонент на граничных значениях рабочих нагрузок или за их пределами, или же в состоянии ограниченных ресурсов, таких как память или доступ к серверу.

Тестирование на отказ и восстановление - тестирование при помощи эмуляции отказов системы или реально вызываемых отказов в управляемом окружении.

Тестирование установки и лицензирования - процесс тестирования установки программного продукта.Включает формальный тест программы установки приложения.

1. Какие существуют виды тестирования в зависимости от глубины покрытия? Дайте характеристику каждому.

Smoke Test – поверхностное тестирование для определения пригодности сборки для дальнейшего тестирования, должно покрывать базовые функции программного обеспечения; уровень качества: Acceptable / Unacceptable.

Minimal Acceptance Test (MAT, Positive Test) – тестирование системы или ее части только на корректных данных/сценариях; уровень качества: High / Medium / Low.

Acceptance Test (AT) – полное тестирование системы или ее части как на корректных (Positive Test), так и на некорректных данных/сценариях (Negative Test); уровень качества: High / Medium / Low. Тест на этом уровне покрывает все возможные сценарии тестирования: проверку работоспособности модулей при вводе корректных значений; проверку при вводе некорректных значений; использование форматов данных отличных от тех, которые указаны в требованиях; проверку исключительных ситуаций, сообщений об ошибках; тестирование на различных комбинациях входных параметров; проверку всех классов эквивалентности; тестирование граничных значений интервалов; сценарии не предусмотренные спецификацией и т.д.

1. Какие существуют тестовые активности? Дайте характеристику каждому.

Виды тестирования в зависимости от знания кода: белый ящик, серый ящик, черный ящик.

Белый ящик (White Box Testing) – тестирование, основанное на анализе внутренней структуры компонентов или системы (у тестировщика есть доступ к внутренней структуре и коду приложения).

Серый ящик (Grey Box Testing) – комбинация методов белого и черного ящика, состоящая в том, что к части кода архитектуры у тестировщика есть, а к части кода – нет.

Черный ящик (Black Box Testing) – тестирование системы без знания внутренней структуры и компонентов системы (у тестировщика нет доступа к внутренней структуре и коду приложения либо в процессе тестирования он не обращается к ним).

Виды тестирования в зависимости от степени автоматизации: ручное, автоматизированное тестирование.

Ручное тестирование – такое тестирование, в котором тест-кейсы выполняются тестировщиком вручную без использования средств автоматизации.

Автоматизированное тестирование (Automated Testing) – набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования. Тест-кейсы частично или полностью выполняет специальное инструментальное средство.

Виды тестирования в зависимости от степени изолированности тестируемых компонентов: модульное, интергационное, системное тестирование.

Модульное тестирование (Unit/Component Testing) – тестируются отдельные части (модули) системы.

Интеграционное тестирование (Integration Testing) – тестируется взаимодействие между отдельными модулями.

Системное тестирование (System Testing) – тестируется работоспособность системы в целом.

Виды тестирования в зависимости от подготовленности: интуитивное тестирование, исследовательское тестирование, тестирование по документации.

Интуитивное тестирование выполняется без подготовки к тестам, без определения ожидаемых результатов, проектирования тестовых сценариев.

Исследовательское тестирование – метод проектирования тестовых сценариев во время выполнения этих сценариев.

Тестирование по документации – тестирование по подготовленным тестовым сценариям, руководству по осуществлению тестов.

Виды тестирования в зависимости от места и времени проведения тестирования: приемочное тестирование, альфа-тестирование, бета-тестирование.

Приемочное тестирование (User Acceptance Testing, UAT) – формальное тестирование по отношению к потребностям, требованиям и бизнес процессам пользователя, проводимое с целью определения соответствия системы критериям приёмки и дать возможность пользователям, заказчикам или иным авторизованным лицам определить, принимать систему.

Альфа-тестирование (Alpha Testing) – моделируемое или действительное функциональное тестирование, выполняется в организации, разрабатывающей продукт, но не проектной командой (это может быть независимая команда тестировщиков, потенциальные пользователи, заказчики). Альфа тестирование часто применяется к коробочному программному обеспечению в качестве внутреннего приемочного тестирования.

Бета-тестирование (Beta Testing) – эксплуатационное тестирование потенциальными или существующими клиентами/заказчиками на внешней стороне (в среде, где продукт будет использоваться) никак связанными с разработчиками, с целью определения действительно ли компонент или система удовлетворяет требованиям клиента/заказчика и вписывается в бизнес-процессы. Бета-тестирование часто проводится как форма внешнего приемочного тестирования готового программного обеспечения для того, чтобы получить отзывы рынка.

Виды тестирования в зависимости от глубины тестового покрытия: Smoke, MAT, AT.

Тестовое покрытие – одна из метрик оценки качества тестирования, представляющая из себя плотность покрытия тестами требований либо исполняемого кода.

Smoke Test – поверхностное тестирование для определения пригодности сборки для дальнейшего тестирования, должно покрывать базовые функции программного обеспечения; уровень качества: Acceptable / Unacceptable.

Minimal Acceptance Test (MAT, Positive Test) – тестирование системы или ее части только на корректных данных/сценариях; уровень качества: High / Medium / Low.

Acceptance Test (AT) – полное тестирование системы или ее части как на корректных (Positive Test), так и на некорректных данных/сценариях (Negative Test); уровень качества: High / Medium / Low. Тест на этом уровне покрывает все возможные сценарии тестирования: проверку работоспособности модулей при вводе корректных значений; проверку при вводе некорректных значений; использование форматов данных отличных от тех, которые указаны в требованиях; проверку исключительных ситуаций, сообщений об ошибках; тестирование на различных комбинациях входных параметров; проверку всех классов эквивалентности; тестирование граничных значений интервалов; сценарии не предусмотренные спецификацией и т.д.

Виды тестирования в зависимости от тестовых активностей: NFT, RT, DV.

Данная классификация тестирования иначе назвыется видами тестирования в зависимости от ширины тестового покрытия.

Тестирование новых функциональностей (New Feature Test, NFT) – определение качества поставленной на тестирование новой функциональности, которая ранее не тестировалась. Данный тип тестирования включает в себя: проведение полного теста (АТ) непосредственно новой функциональности; тестирование новой функциональности на соответствие документации; проверку всевозможных взаимодействий ранее реализованной функциональности с новыми модулями и функциями.

Регрессионное тестирование (Regression Testing, RT) проводится с целью оценки качества ранее реализованной функциональности. Включает в себя проверку стабильности ранее реализованной функциональности после внесения изменений, например добавления новой функциональности, исправление дефектов, оптимизация кода, разворачивание приложения на новом окружении. Регрессионное тестирование как правило выполняется на уровне MAT.

Валидация дефектов (Defect Validation, DV) – проверка результатов исправления дефектов; может включать элементы регрессионного тестирования; уровень проверки не определяется.

1. Какие существуют виды тестирования в зависимости от знания кода? Дайте характеристику каждому.

Белый ящик (White Box Testing) – тестирование, основанное на анализе внутренней структуры компонентов или системы (у тестировщика есть доступ к внутренней структуре и коду приложения).

Серый ящик (Grey Box Testing) – комбинация методов белого и черного ящика, состоящая в том, что к части кода архитектуры у тестировщика есть, а к части кода – нет.

Черный ящик (Black Box Testing) – тестирование системы без знания внутренней структуры и компонентов системы (у тестировщика нет доступа к внутренней структуре и коду приложения либо в процессе тестирования он не обращается к ним).

1. Какие существуют виды тестирования в зависимости от степени автоматизации? Дайте характеристику каждому.

Ручное тестирование – такое тестирование, в котором тест-кейсы выполняются тестировщиком вручную без использования средств автоматизации.

Автоматизированное тестирование (Automated Testing) – набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования. Тест-кейсы частично или полностью выполняет специальное инструментальное средство.

1. Какие существуют виды тестирования в зависимости от изолированности компонентов? Дайте характеристику каждому.

Виды тестирования в зависимости от степени изолированности тестируемых компонентов: модульное, интергационное, системное тестирование.

Модульное тестирование (Unit/Component Testing) – тестируются отдельные части (модули) системы.

Интеграционное тестирование (Integration Testing) – тестируется взаимодействие между отдельными модулями.

Системное тестирование (System Testing) – тестируется работоспособность системы в целом.

1. Какие существуют виды тестирования в зависимости от подготовленности? Дайте характеристику каждому.

Виды тестирования в зависимости от подготовленности: интуитивное тестирование, исследовательское тестирование, тестирование по документации.

Интуитивное тестирование выполняется без подготовки к тестам, без определения ожидаемых результатов, проектирования тестовых сценариев.

Исследовательское тестирование – метод проектирования тестовых сценариев во время выполнения этих сценариев.

Тестирование по документации – тестирование по подготовленным тестовым сценариям, руководству по осуществлению тестов.

1. Какие существуют виды тестирования в зависимости от места и времени проведения? Дайте характеристику каждому.

Виды тестирования в зависимости от места и времени проведения тестирования: приемочное тестирование, альфа-тестирование, бета-тестирование.

Приемочное тестирование (User Acceptance Testing, UAT) – формальное тестирование по отношению к потребностям, требованиям и бизнес процессам пользователя, проводимое с целью определения соответствия системы критериям приёмки и дать возможность пользователям, заказчикам или иным авторизованным лицам определить, принимать систему.

Альфа-тестирование (Alpha Testing) – моделируемое или действительное функциональное тестирование, выполняется в организации, разрабатывающей продукт, но не проектной командой (это может быть независимая команда тестировщиков, потенциальные пользователи, заказчики). Альфа тестирование часто применяется к коробочному программному обеспечению в качестве внутреннего приемочного тестирования.

Бета-тестирование (Beta Testing) – эксплуатационное тестирование потенциальными или существующими клиентами/заказчиками на внешней стороне (в среде, где продукт будет использоваться) никак связанными с разработчиками, с целью определения действительно ли компонент или система удовлетворяет требованиям клиента/заказчика и вписывается в бизнес-процессы. Бета-тестирование часто проводится как форма внешнего приемочного тестирования готового программного обеспечения для того, чтобы получить отзывы рынка.

1. Какие этапы составляют процесс тестирования?

Процесс тестирования программного продукта включает следующие этапы:

1. Изучение и анализ предмета тестирования.

2) Планирование тестирования.

3) Исполнение тестирования.

1. Какая композиция тестов выполняется для первой поставки программного продукта?

Для первой поставки программного обеспечения рекомендуется проводить Smoke + NFTAT готовой функциональности: поверхностное тестирование (Smoke Test) выполняется для определения пригодности сборки для дальнейшего тестирования; полное тестирование системы или ее части как на корректных, так и на некорректных данных/сценариях (Acceptance Test, AT) позволяет обнаружить дефекты и внести запись о них в багтрэкинговую систему.

1. Какие композиция тестов выполняется для последующих поставок программного продукта?

Для последующих поставок программного обеспечения композиции тестов могут быть следующими.

Если не была добавлена новая функциональность, то: DV + RTMAT. Т.е., выполняется проверка исправления дефектов программистом (Defect Validation,

DV), а также проверка работоспособности остальной функциональности после исправления дефектов на позитивных сценариях (Minimal Acceptance Test, MAT).

Если была добавлена новая функциональность, то:

Smoke + DV + NFTAT + RTMAT. В частности, выполняется поверхностное тестирование (Smoke Test), проверка исправления дефектов программистом (Defect Validation, DV), тестирование новых функциональностей (New Feature

Testing, NFT), проверка старых функциональностей, т.е. регрессионное тестирование (Regression Test).

Если была добавлена новая функциональность, то возможен также вариант: DV + NFTAT + RTMAT, т.е. без выполнения Smoke Test.

Таким образом, для второй и последующих поставок обобщенная схема композиции тестов выглядит следующим образом: (Smoke) + DV + (NFTAT) + RTMAT.