

**«Практическая работа: «Планирование тестирования»»**

**Выполнил(и): студенты группы 2ИСП11-18**

Кашутина М.А., Самсонова Н.М.

**Проверил(и)**: Возвахов Д.А., Мартиросян В.А.

Москва

2022

Оглавление

[Лабораторная работа №1 3](#_Toc113529475)

[Отчет 16](#_Toc113529476)

## Лабораторная работа №1

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.

Таблица 1.1 – Тестовые проверки для различных видов тестирования

|  |  |  |
| --- | --- | --- |
| **Объект тестирования**: мобильное приложение | | |
| **Вид тестирования** | **Краткое определение вида тестирования** | **Тестовые проверки** |
| **Functional Testing** | Тестирование, основанное на сравнительном анализе спецификации и функциональности компонента или системы. | На главном экране выбор группы и факультета осуществляется через системный графический компонент прокручиваемого списка, а выбор временного промежутка с помощью выпадающего списка.  Импорт расписания осуществляется получение данных расписания с сайта УГАТУ с последующим добавлением записей в календарь.  Очистка раннее внесенных записей осуществляется удалением календаря. |
| **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. | Нет риска утечки данных, причинения вреда здоровью, бизнесу, программа, собственности или окружающей среде при использовании оговоренным образом. |
| **Security Testing** | Тестирование с целью оценить защищенность программного продукта от внешних воздействий (от проникновений) | При попытке использования не оговоренным образом или использовании не предусмотренных команд происходит возраст к началу сценария. Защита от сетевых атак не предусмотрена. |
| **Compatibility Testing** | Проверка работоспособности приложения в различных средах (браузеры и их версии, операционные системы, их типы, версии и разрядность) | Данное приложение разработано для операционной системы Android. Гарантирована работа для версии ОС выше 7.0.  Контент отображается одинаково и работает на всех устройствах без перебоев в функционировании и ошибок в верстке, а также с одинаково корректной читабельностью контента. |
| **GUI Testing** | Тестирование, выполняемое путем взаимодействия с системой через графический интерфейс пользователя (правописание выводимой информации; расположение и выравнивание элементов GUI; соответствие названий форм/элементов GUI их назначению; унификация стиля, цвета, шрифта; окна сообщений; изменение размеров окна, по- ведение курсора и горячие клавиши). | Графический интерфейс пользователя выполнен в едином стиле с помощью графических компонентов ОС Android. |
| **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 (на этом уровне обращают внимание на визуальное оформление, навигацию, логичность, наличие обратной связи и др.). | На стартовом экране в списке групп выбрать необходимую группу.  В выпадающем списке выбрать нужный временный промежуток.  После нажатия на кнопку «Импорт» та блокируется до завершения операции импорта.  По завершению импорта выводится уведомление о результате. |
| **Accessibility Testing** | Тестирование, которое определяет степень легкости, с которой пользователи с ограниченными способностями могут использовать систему или ее компоненты. | Средства операционной системы позволяют увеличивать размер шрифта. |
| **Internationalization Testing** | Тестирование адаптации продукта к языковым и культурным особенностям целого ряда регионов, в которых потенциально может использоваться продукт. | Не реализовано данное тестирование в процессе разработки. |
| **Performance Testing** | Процесс тестирования с целью определения производительности программного продукта. В рамках тестирования производительности выделяют нагрузочное тестирование, объемное тестирование, тестирование стабильности и надежности, стрессовое тестирование. | Тестирование при устойчивом Интернет-соединении прошла успешно. |
| **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. | Система сохраняет работоспособность и обеспечивается восстановление своих функций при возникновении следующих ситуаций:  При возникновении ошибки импорта удаляются ранее внесенные записи в календарь.  При недостатке оперативной памяти на устройстве отдельные модули приложения выгружаются из ОЗУ в ПЗУ средствами ОС. |
| **Negative Testing** | Это метод тестирования приложения или системы, который гарантирует, что график приложения соответствует требованиям и может обрабатывать нежелательный ввод и поведение пользователя. | При использовании не предусмотренных команд или некорректных сообщений происходит возврат к началу сценария. |
| **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). | Не предусмотрено для тестирования. |
| **Automated Testing** | Набор техник, под- ходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования. | Не предусмотрено для тестирования. |
| **Unit/Component Testing** | Тестируются отдельные части (модули) системы. | Не предусмотрено для тестирования. |
| **Integration Testing** | Тестируются отдельные части (модули) системы. | Программное обеспечение не предусматривает наличие дополнительных модулей и является целостным. |

*Контрольные вопросы/ответы*:

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.

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.

* **Функциональные** - тестирование, основанное на сравнительном анализе спецификации и функциональности компонента или системы;
* **Пограничные** - проверка работоспособности приложения в различных средах (браузеры и их версии, операционные системы, их типы, версии и разрядность).
* **Нефункциональные** – тестирование, необходимое для определения характеристик систем или программ, которые могут быть количественно измерены, такие как время отклика при тестировании производительности.

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

**Функциональное тестирование** (Functional Testing) – тестирование, основанное на сравнительном анализе спецификации и функциональности компонента или системы.

**Тестирование безопасности** (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.

**Тестирование защищенности** (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.

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

**Тестирование требований** (Requirements Testing) – проверка требований на соответствие основным атрибутам качества.

**Тестирование прототипа** (Prototype Testing) – метод выявления структурных, логических ошибок и ошибок проектирования на ранней стадии развития продукта до начала фактической разработки.

**Тестирование пользовательского интерфейса** (GUI Testing) – тестирование, выполняемое путем взаимодействия с системой через графический интерфейс пользователя (правописание выводимой информации; расположение и выравнивание элементов GUI; соответствие названий форм/элементов GUI их назначению; унификация стиля, цвета, шрифта; окна сообщений; изменение размеров окна, поведение курсора и горячие клавиши).

**Тестирование удобства использования** (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 (на этом уровне обращают внимание на визуальное оформление, навигацию, логичность, наличие обратной связи и др.).

**Тестирование доступности** (Accessibility Testing) – тестирование, которое определяет степень легкости, с которой пользователи с ограниченными способностями могут использовать систему или ее компоненты.

**Тестирование интернационализации** (Internationalization Testing) – тестирование адаптации продукта к языковым и культурным особенностям целого ряда регионов, в которых потенциально может использоваться продукт.

**Тестирование локализации** (Localization Testing) – тестирование адаптации продукта к языковым и культурным особенностям конкретного региона, отличного от того, в котором разрабатывался продукт.

**Тестирование производительности** (Performance Testing) – процесс тестирования с целью определения производительности программного продукта. В рамках тестирования производительности выделяют нагрузочное тестирование, объемное тестирование, тестирование стабильности и надежности, стрессовое тестирование.

**Нагрузочное тестирование** (Performance and Load Testing) – вид тестирования производительности, проводимый с целью оценки поведения компонента или системы при возрастающей нагрузке, например, количестве параллельных пользователей и/или операций, а также определения, какую нагрузку может выдержать компонент или система.

**Объемное тестирование** (Volume Testing) – позволяет получить оценку производительности при увеличении объемов данных в базе данных приложения.

**Тестирование стабильности и надежности** (Stability/Reliability Testing) – позволяет проверять работоспособность приложения при длительном (многочасовом) тестировании со средним уровнем нагрузки.

**Стрессовое тестирование** (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.

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.

# Отчет

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

Протестировала собственный объект тестирования (*мобильное приложение «Импорт расписания УГАТУ»*) на различных видах тестирования.

Подробно ответила на контрольные вопросы, которые были в теоретической части.

Подводя итоги по проделанной работе, можно сказать, что за эту лабораторную работу я изучила классификацию видов тестирования, разработала проверки для различных видов тестирования, научилась планировать тестовые активности в зависимости от особенностей поставляемой на тестирование функциональности. Протестировала собственный объект тестирования (*мобильное приложение «Импорт расписания УГАТУ»*) с помощью большого количества видов тестирования. Ответила на контрольные вопросы, которые были в теоретической части.