

Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ В.Ф.УТКИНА»  
Рязанский станкостроительный колледж РГРТУ

Лабораторная работа

**Дополнительная профессиональная программа  
повышения квалификации  
«Тестирование программного обеспечения (с учетом стандарта  
Ворлдскиллс по компетенции «Программные решения для бизнеса»)»**

Мазуренко Валерия Витальевна

Рязань 2022

## Задание №1

### Виды тестирования. Планирование тестирования

Цель работы: изучить классификацию видов тестирования, практически закрепить эти знания путем генерации тестов различных видов, научиться планировать тестовые активности в зависимости от специфики поставляемой на тестирование функциональности.

Теоретические сведения

Тестирование – процесс, направленный на оценку корректности, полноты и качества разработанного программного обеспечения.

Тестирование можно классифицировать по очень большому количеству признаков.

Далее приведен обобщенный список видов тестирования по различным основаниям.

#### **Типы тестов по покрытию (по глубине)**

**Smoke test** – тестирование системы для определения корректной работы базовых функций программы в целом, без углубления в детали. При проведении теста определяется пригодность сборки для дальнейшего тестирования.

**Minimal Acceptance Test (MAT, Positive test):** тестирование системы или ее части только на валидных данных (валидные данные – это данные, которые необходимо использовать для корректной работы модуля/функции). При тестировании проверяется правильной работы всех функций и модулей с валидными данными.

Для крупных и сложных приложений используется ограниченный набор сценариев и функций.

**Acceptance Test (AT):** полное тестирование системы или ее части как на корректных, так и на некорректных данных/сценариях. Вид теста, направленный на подтверждение того, что приложение может использоваться по назначению при любых условиях.

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.

### **Тестовые активности (типы тестов по покрытию (по ширине)):**

**Defect Validation** – проверка результата исправления дефектов. Включает в себя проверку на воспроизводимость дефектов, которые были исправлены в новой сборке продукта, а также проверку того, что исправление не повлияло на ранее работавшую функциональность

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

**Regression testing (регрессионное тестирование)** – 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 может быть проведено на уровне Smoke, MAT или AT.

### **Типы тестов по знанию коду**

**Черный ящик** – тестирование системы, функциональное или нефункциональное, 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.

**Белый ящик** – тестирование основанное на анализе внутренней структуры компонентов или системы. У тестировщика есть доступ к внутренней структуре и коду приложения.

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

### **Типы тестов по степени автоматизации**

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

**Автоматизированное** – 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.

### **Типы тестов по изолированности компонентов**

**Unit/component (модульное)** – тестирование отдельных компонентов (модулей) программного обеспечения.

**Integration (интеграционное)** – тестируется взаимодействие между интегрированными компонентами или системами.

**System (системное)** – тестируется работоспособность системы в целом с целью проверки того, что она соответствует установленным требованиям.

### **Типы тестов по подготовленности.**

**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. Тестировщик совершает проверки, продумывает их, придумывает новые проверки, часто использует для этого полученную информацию.

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

### **Типы тестов по месту и времени проведения**

**User Acceptance Testing (UAT) (приемочное тестирование)** – 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.

**Alpha Testing (альфа-тестирование)** – 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.

**Beta Testing (бета-тестирование)** – 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.

### **Типы тестов по объекту тестирования**

**Functional testing (функциональное тестирование)** – это тестирование, основанное на анализе спецификации, функциональности компонента или системы. Функциональным можно назвать любой вид тестирования, который согласно требованиям проверяет правильную работу.

**Safety testing (тестирование безопасности)** – тестирование программного продукта с целью определить его безопасность (безопасность – способность программного продукта при использовании оговоренным образом оставаться в рамках приемлемого риска причинения вреда здоровью, бизнесу, программам, собственности или окружающей среде.

**Security testing (тестирование защищенности)** – это тестирование с целью оценить защищенность программного продукта. Тестирование защищенности проверяет фактическую реакцию защитных механизмов, встроенных в систему, на проникновение.

**Compatibility testing (тестирование совместимости)** – процесс тестирования для определения возможности взаимодействия программного продукта, проверка работоспособности приложения в различных средах (браузеры и их версии, операционные системы, их типа, версии и разрядность)

#### **Виды тестов:**

- ☐ кросс-браузерное тестирование (различные браузеры или версии браузеров)
- ☐ кросс-платформенное тестирование (различные операционные системы или версии операционных систем)

**Нефункциональное тестирование** – это проверка характеристик программы. Иначе говоря, когда проверяется не именно правильность работы, а какие-либо свойства (внешний вид и удобство пользования, скорость работы и т.п.).

**1. Тестирование пользовательского интерфейса (GUI)** – тестирование, выполняемое путем взаимодействия с системой через графический интерфейс пользователя.

- ☐ навигация
- ☐ цвета, графика, оформление
- ☐ содержание выводимой информации
- ☐ поведение курсора и горячие клавиши
- ☐ отображение различного количества данных (нет данных, минимальное и максимальное количество)
- ☐ изменение размеров окна или разрешения экрана

**2. 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.

- ☐ визуальное оформление
- ☐ навигация
- ☐ логичность

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

**4. Тестирование интернационализации** – тестирование способности продукта работать в локализованных средах (способность изменять элементы интерфейса в зависимости от длины и направления текста, менять сортировки/форматы под различные локали и т.д.). (Максим Черняк).

**Интернационализация** – это процесс, упрощающий дальнейшую адаптацию продукта к языковым и культурным особенностям региона, отличного от того, в котором разрабатывался продукт. Это адаптация продукта для потенциального использования практически в любом месте, Интернационализация производится на начальных этапах разработки, в то время как локализация — для каждого целевого языка.

**5. Тестирование локализации (Localization testing)** – тестирование, проводимое с целью проверить качество перевода продукта с одного языка на другой.

**6. Тестирование производительности или нагрузочное тестирование** – процесс тестирования с целью определения производительности программного продукта.

Виды тестов:

- ☐ **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.

**7. Тестирование требований (Requirements testing)** – проверка требований на соответствие основным характеристикам качества.

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

**9. Тестирование установки (Installability testing) и лицензирования** – процесс тестирования устанавливаемости программного продукта.

Виды тестов:

☐ формальный тест программы установки приложения (проверка пользовательского интерфейса, навигации, удобства пользования, соответствия общепринятым стандартам оформления);

☐ функциональный тест программы установки;

☐ тестирование механизма лицензирования и функций защиты от пиратства;

☐ проверка стабильности приложения после установки.

**10. Тестирование на отказ и восстановление (Failover and Recovery Testing)** – тестирование при помощи эмуляции отказов системы или реально вызываемых отказов в управляемом окружении.

**Тестирование программного продукта включает следующие этапы:**

1. Изучение и анализ предмета тестирования.

2. Планирование тестирования.

3. Исполнение тестирования.

**Изучение и анализ предмета тестирования** начинается еще до утверждения спецификации и продолжается на стадии разработки (кодирования) программного обеспечения. Конечной целью этапа изучение и анализ предмета тестирования является получение ответов на два вопроса: - какие функциональности предстоит протестировать, - как эти функциональности работают.

**Планирование тестирования** происходит на стадии разработки (кодирования) программного обеспечения. На стадии планирования тестирования перед тестировщиком стоит задача поиска компромисса между объемом тестирования, который возможен в теории, и объемом тестирования, который возможен на практике. На данной стадии необходимо ответить на вопрос: как будем тестировать? Результатом планирования тестирования является тестовая документация.

**Выполнение тестирования** происходит на стадии тестирования и представляет собой практический поиск дефектов с использованием тестовой документации, составленной ранее.

**Для всех программных продуктов выполняют следующие типы тестов и их композиции.**

Для первого билда рекомендуется проводить Smoke+AT 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 багтрекинг-систему.

**Для последующих билдов композиции тестов могут быть следующими:**

- Если не была добавлена новая функциональность, то: **DV+MAT. 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+NFT+Regression Test.** 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+NFT+Regression test, т.е. без выполнения Smoke Test.** В зависимости от типа и специфики приложения (web, desktop, mobile) выполняют специализированные тесты (например, кроссбраузерное или кроссплатформенное тестирование, тестирование локализации и интернационализации и др.).



## Лабораторная работа

План для объекта «Увлажнитель воздуха».

### Верификация :

- Габариты домашний или стационарный
- Подцветка есть ли нет?
- Таймер да/нет
- Цвет
- Объем воды
- Регулировка мощности

### Валидность :

- Предназначение летом или зимой?
- Где устанавливается на улице или в помещении

*Типы тестов по покрытию (по глубине)*

### Smoke test:

Проверка на заводе пригодность работы:

- Включается данный прибор?
- Нагревает воду?

### Minimal Acceptance Test (MAT, Positive test):

- Модуль нагрева нагревает воду?
- Модуль подцветки работает?

### Acceptance Test (AT):

- Проверка, работает на улице?
- Как работает в доме?
- Как быстро нагревает воду?

### Тестовые активности (типы тестов по покрытию (по ширине)):

#### Defect Validation:

- Быстрый нагревается воды за 1 сек

#### New Feature Test (NFT, AT of NF) :

- Добавление таймера
- Отключение, если закончится вода, проверка на действительность
- Установление мощности, проверка, как она меняется
- Увеличение объема на 2 л, тем самым соответствует кубатуре 20 кв м

### **Regression testing (регрессионное тестирование):**

- Как прибор работает в течении 12 часов
- Отключается прибор, без воды ?
- Соответствует мощность: минимальная 1, максимальная 6, средняя 3?
- Работа прибора на улице
- Работа прибора дома

### **Типы тестов по знанию коду**

#### **Черный ящик :**

Непонятно, как устроен

Нажимаем, куда видим

#### **Белый ящик :**

Тестируем прибор по назначению в доме

#### **Серый ящик**

Проверяем прибор на работоспособность:

- с водой/без воды

### **Типы тестов по степени автоматизации**

#### **Ручное**

Заливаем воду проверяем идет пар или нет

#### **Автоматизированное**

Заводская проверка автоматизированным станком

### **Типы тестов по изолированности компонентов**

#### **Unit/component (модульное)**

- Заливаем воду, проверяем на герметизацию бочка
- Проверяем шнур на подачу электричества
- Модуль нагрева на отключение без воды

#### **Integration (интеграционное)**

- Выдержит ли конструкция подставка с бочком воды вверху

#### **System (системное)**

- Сработает отключение прибора из сети при отсутствии воды
- Напряжение 220 в.

### **Типы тестов по подготовленности.**

#### **Интуитивное тестирование**

-Включаем прибор в разных условиях в поезде/дома/на улице

#### **Исследовательское тестирование**

-Пробуем не залить воду в бачок/ залить полностью

#### **Тестирование по документации**

-Смотрим инструкцию по эксплуатации

### **Титы тестов по месту и времени проведения**

#### **User Acceptance Testing (UAT) (приемочное тестирование)**

-Отдать на проверку друзьям

#### **Alpha Testing (альфа-тестирование)**

-Отдать на проверку в другой цех

#### **Beta Testing (бета-тестирование)**

-Как прибор работает в домашних условиях

### **Типы тестов по объекту тестирования**

#### **Functional testing (функциональное тестирование)**

-Как работает дома?:

-отключается, если в контейнере нет воды?

-регулируется ли мощность подачи пара?

-верно работает таймер?

#### **Safety testing (тестирование безопасности)**

-Не оголены провода ввилке?

-Может ударить током?

-Герметичен ли контейнер, может протечь на пол?

#### **Security testing (тестирование защищенности)**

-Самостоятельно починить нагревательный элемент?

#### **Compatibility testing (тестирование совместимости)**

-Смотрим работоспособность без воды

-Минимальное количество воды

## **Нефункциональное тестирование**

### **1. Тестирование пользовательского интерфейса (GUI)**

- Удобно ли наливать воду?
- На сколько лаконичный дизайн?
- На сколько понятный дисплей?

### **2. Тестирование удобства использования (Usability Testing)**

- Все ли функции присутствуют
- На сколько понятны функции в обозначении регулировок мощности/таймера

### **3. Тестирование доступности (Accessibility testing)**

- Простота в эксплуатации старому/малому

### **4. Тестирование интернационализации**

- Обозначение символами без языкового барьера

### **5. Тестирование локализации (Localization testing)**

- Не требуется

### **6. Тестирование производительности или нагрузочное тестирование**

Виды тестов:

#### ☐ **нагрузочное тестирование (Performance and Load testing)**

- Сколько по времени может работать с полным контейнером?

#### ☐ **объемное тестирование (Volume testing)**

- Как долго может работать от сети с постоянно добавлением воды?

#### ☐ **тестирование стабильности и надежности (Stability / Reliability testing)**

- На сколько защищена плата от перегрева при длительной работе?

#### ☐ **стрессовое тестирование (Stress testing)**

- Уронить
- Вместо воды налить молоко, проверить на сколько надежный элемент нагрева

**7. Тестирование требований (Requirements testing)** – проверка требований на соответствие основным характеристикам качества.

- Включит прибор на час по таймеру с минимальным количеством воды:
- проверить отключается прибор да/нет
- проверить работу таймера да/нет

### **8. Тестирование прототипа (Prototype testing)**

- Все ли провода подключены к плате
- Исправный ли элемент нагрева
- Герметичный контейнер

## **9. Тестирование установки (Installability testing) и лицензирования**

-Читабельность дисплея

## **10. Тестирование на отказ и восстановление (Failover and Recovery Testing)**

-Перегрев платы

-Плохая вода появление накипи