**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
 РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**«Севастопольский государственный университет»**

**Институт информационных технологий и управления   
в технических системах**

**ОСОБЕННОСТИ ТЕСТИРОВАНИЯ ПО**

**Методические указания**

к лабораторной работе №5

по дисциплине

**“Управление IT-проектами”**

для студентов специальности 09.03.02 –

"Информационные системы и технологии"

всех форм обучения

**Севастополь**

**2015**

1. **Цель работы:** изучить особенности тестирования программного обеспечения; рассмотреть виды тестирования.
2. **ОСНОВНЫЕ ПОЛОЖЕНИЯ**

Тестирование – это проверка соответствия между реальным поведением программы и ее ожидаемым поведением на конечном наборе тестов, выбранных определенных образом.

Тестирование программного обеспечения  – процесс исследования, испытания программного продукта, имеющий две различные цели:

* продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
* выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.



Все вид­­­ы тестирования программного обеспечения, в зависимости от преследуемых целей, можно условно разделить на следующие группы:

1. функциональные;
2. нефункциональные;
3. связанные с изменениями.

**Функциональные виды тестирования**

Функциональные тесты базируются на функциях и особенностях, а также взаимодействии с другими системами, и могут быть представлены на всех уровнях тестирования: компонентном или модульном (Component/Unit testing), интеграционном (Integration testing), системном (System testing) и приемочном (Acceptance testing). Функциональные виды тестирования рассматривают внешнее поведение системы. Далее перечислены одни из самых распространенных видов функциональных тестов:

* функциональное тестирование (Functional testing);
* тестирование безопасности (Security and Access Control Testing);
* тестирование взаимодействия (Interoperability Testing).

**Нефункциональные виды тестирования**

Нефункциональное тестирование описывает тесты, необходимые для определения характеристик программного обеспечения, которые могут быть измерены различными величинами. В целом, это тестирование того, "Как" система работает. Далее перечислены основные виды нефункциональных тестов:

Все виды тестирования производительности:

* нагрузочное тестирование (Performance and Load Testing);
* стрессовое тестирование (Stress Testing);
* тестирование стабильности или надежности (Stability / Reliability Testing);
* объемное тестирование (Volume Testing);

Тестирование установки (Installation testing);

Тестирование удобства пользования (Usability Testing);

Тестирование на отказ и восстановление (Failover and Recovery Testing);

Конфигурационное тестирование (Configuration Testing);

**Связанные с изменениями виды тестирования**

После проведения необходимых изменений, таких как исправление бага/дефекта, программное обеспечение должно быть перетестировано для подтверждения того факта, что проблема была действительно решена. Ниже перечислены виды тестирования, которые необходимо проводить после установки программного обеспечения, для подтверждения работоспособности приложения или правильности осуществленного исправления дефекта:

* дымовое тестирование (Smoke Testing);
* регрессионное тестирование (Regression Testing);
* тестирование сборки (Build Verification Test);
* санитарное тестирование или проверка согласованности/исправности (Sanity Testing).

1. **ПРАКТИЧЕСКАЯ ЧАСТЬ**

Протестировать предмет согласно варианту (таблица 1).

Таблица 1 – Варианты

|  |  |
| --- | --- |
| **Вариант** | **Описание** |
| 1. | Протестировать ластик |
| 2. | Протестировать лак для волос |
| 3. | Протестировать бритву |
| 4. | Протестировать кирпич |
| 5. | Протестировать ножницы |
| 6. | Протестировать воздушный шарик |
| 7. | Протестировать шариковую ручку |
| 8. | Свой вариант |

Выполнение предусматривает командную работу (**2 человека**).

**Пример выполнения:**

*Задание*: протестируйте футбольный мяч.

**Unit Testing**

Необходимо наличие 32 синтетических панелей, 12 пятиугольных и 20 шестиугольных для изготовления покрышки.

Подкладка должна состоять из 4-х слоев спрессованного хлопка.

Камера должна быть изготовлена из латекса.

*Все необходимые детали в наличии, из требуемого материала и в нужном количестве.*

**Integration Testing**

Покрышка, подкладка и камера производятся на разных стадиях изготовления мяча.

Требуется собрать все части вместе. Вес готового изделия должен составлять от 410 до 450 грамм.

*Все части мяча соединили в одно изделие, взвесили. Вес готового изделия составил 440 грамм. Дефект не найден.*

**Smoke Testing**

Мяч должен иметь отскок при падении с высоты без ускорения.

Бросить мяч с высоты 2 метра, проверить наличие отскока.

*Дефекты не найдены.*

**Functional Testing**

Мяч должен иметь отскок 40-50 процентов от высоты падения без ускорения.

Бросить мяч без ускорения с высоты 2 метра, отскок должен составлять около 1 метра (±20 см).

*Дефекты не найдены.*

**GUI Testing**

Покрышка должна быть изготовлена из 32 синтетических панелей, 12 пятиугольных и 20 шестиугольных. Не должны быть видны швы.

*Внешний вид соответствует описанию.*

**Usability Testing**

Не должны быть видны швы и ниппель. Покрышка должна быть изготовлена из 12 черных пятиугольников и 20 белых шестиугольников.

*Швы и ниппель не видны, что свидетельствует о хорошем качестве изделия. Внешний вид соответствует описанию.*

**Performance Testing**

Мяч не должен терять давление в течение 72 часов.

*Мяч накачали (величина давления составила 1 атм). Оставили на 72 часа. После 72 часов давление в мяче не изменилось. Дефект**не обнаружен.*

**Load Testing**

При 2000 ударах со скоростью 50 км/ч за 24 часа мяч не должен иметь повреждений.

*Мяч выстреливают в металлическую пластину со скоростью 50 км/ч 2000 раз в течении 24 часов. Швы и клапан сохранились и не повреждены. Дефект не найден.*

**Stress Testing**

При 5000 ударах со скоростью 50 км/ч за 48 часов мяч не должен иметь значительных повреждений.

*Мяч выстреливают в металлическую пластину со скоростью 50 км/ч 5000 раз в течении 48 часов.*

*Дефект: швы стали видны. На 5 синтетических панелях появились трещины.*

**Acceptance Testing**

Мяч не должен терять сферическую форму при приложении на него силы весом в 50 кг. Мяч должен иметь отскок 40-50 процентов от высоты падения без ускорения. Не должны быть видны швы.

*Мяч соответствует требованиям.*

**Alpha Testing**

Передать изделие на тестирование коллегам, занимающимся футболом. Срок 2 месяца.

**Beta Testing**

Передать изделие в ПФК «Севастополь». Срок 2 месяца.

1. **СОДЕРЖАНИЕ ОТЧЁТА**

Отчёт должен содержать в себе:

1. Титульный лист.

2. Краткое описание видов тестирования.

3. Тестирование предмета согласно описанию варианта.

3. Вывод о проделанной работе (полученные навыки, замеченные трудности или, наоборот, улучшения в работе).

4. Список использованных источников

1. **КОНТРОЛЬНЫЕ ВОПРОСЫ**
2. Что такое тестирование ПО?
3. Каким образом происходит тестирование ПО?
4. Что подпадает под тестирование?
5. Какие бывают виды тестирования? Охарактеризуйте каждый вид более подробно.
6. Что такое автоматизация тестирования? Зачем она применяется?
7. **РЕКОМЕНДУЕМЫЕ ИНТЕРНЕТ-ИСТОЧНИКИ**
8. Савин Р. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах. – М.: Дело, 2007. – 312 с. Электронный ресурс. <http://adm-lib.ru/books/4/testirovanie_dot-com.pdf>
9. Виды тестирования программного обеспечения: <http://www.protesting.ru/testing/testtypes.html>

**Дополнительная литература**

1. «Быстрое тестирование», Роберт Калбертсон, Крис Браун, Гэри Кобб;
2. «Введение в тестирование программного обеспечения», Луиза Тамре;
3. «Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование», Рекс Блэк;
4. «A Practitioner’s Guide to Software Test Design», Lee Copeland;
5. «Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений», Сэм Канер, Джек Фолк, Енг Кек Нгуен;
6. «Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем», Борис Бейзер
7. «Автоматизированное тестирование программного обеспечения», Элфрид Дастин, Джефф Рэшка, Джон Пол.