Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

Высшего профессионального образования

Владимирский государственный университет

Кафедра информационных систем

информационного менеджмента

**Методические указания к лабораторныой работе №1. Изучение этапов тестирования ПО. Тестирование калькулятора**

**по дисциплине «Тестирование программного обеспечения»**

Составители:

Котова Е.М.,

Конушин А.В.

Владимир 2012

**Лабораторная работа №1**

**Изучение этапов тестирования ПО. Тестирование калькулятора**

1. **Цель работы**

Изучить этапы тестирования ПО, виды тестирования. Провести тестирование предложенного приложения.

1. **Общие сведения**

**Тестирование** – это процесс, содержащий в себе все активности жизненного цикла, касающиеся планирования, подготовки и оценки программного продукта, а также связанных с этим результатов работ с целью определить, что они соответствуют описанным требованиям, подходят для заявленных целей и для определения дефектов.

**Цели тестирования:**

* Обнаружение дефектов;
* Повышение уверенности в уровне качества;
* Предоставление информации для принятия решений;
* Предотвращение дефектов;
* Оценка тестируемого приложения.

**Этапы тестирования:**

1. Этап «*Планирование*».

На этом этапе определяются задачи, сроки тестирования, составляют календарный план работ, пишется тест-план. В тест-план заносится следующая информация:

* Цель тестирования (то, для чего проводится тестирование).
* Процедуры и методики тестирования (используемые техники и технологии, см. таблицу 1).
* Описание тестируемой функциональности (то, что будет протестировано).
* Критерии завершения тестирования (выполнение всех тестовых сценариев будет одним из критериев завершения тестирования).

1. Этап «*Анализ и проектирование тестов*».

*Тестовый сценарий* (тест-кейс) – это высокоуровневое описание действий тестировщика для проверки функций системы с целью обнаружения дефектов.

Каждый *тестовый сценарий* содержит:

* название;
* начальные условия;
* последовательность действий;
* ожидаемый результат.

# *Методики проектирования тестов.*

1. *Эквивалентное разбиение* – методика тестирования, в котором текстовые сценарии разбиваются в порядке эквивалентности.

*Цель:* разбиение входных и выходных данных на классы эквивалентности для уменьшения числа тестов.

*Предположение****:*** программа реагирует на все элементы одного класса эквивалентности одинаково.

*Проектирование тестовых сценариев:*

1. комбинирование классов эквивалентности
2. для каждого класса эквивалентности создаётся тестовый сценарий
3. для каждого тестового сценария определяются предусловия и ожидаемые результаты

*Пример*. Допустим, тестируем Интернет-магазин, продающий карандаши. В заказе необходимо указать количество карандашей (максимум для заказа – 1000 штук). В зависимости от заказанного количества карандашей различается стоимость:

1 – 500 – 10 руб. за карандаш;

501 – 1000 – 8 руб. за карандаш;

Очевидно, что входные данные мы можем разделить на следующие классы эквивалентности:

* Невалидное значение: >1000 штук;
* Невалидное значение: <=0;
* Валидное значение: от 1 до 500;
* Валидное значение: от 501 до 1000.

1. *Анализ граничных значений*

По статистике много дефектов находится вокруг граничных значений, разделяющих классы эквивалентности. Следовательно, границы необходимо тестировать более тщательно для выявления дефектов, связанных с операторами отношения **(<, <=, <>, =, >=, >).** Создан для дополнения эквивалентного разбиения.

Пример. Используя пример, приведенный в эквивалентном разбиении, можно определить граничные значения для каждого класса эквивалентности:

* 1000, 1001;
* 0, 1;
* 500, 501.

1. На этапе «*Реализация и выполнение тестов*» происходит выполнение тестов, запись результатов тестирования, оформление дефектов.

*Дефект*: 1. частично неверная программа, неверная команда или определение данных, являющееся причиной отказа; 2. состояние программного продукта или одного из его компонентов, которое в определённых условиях (например, при большой нагрузке) может нарушить требуемую функцию продукта или привести к отказу.

*Дефект* содержит в себе следующие данные:

* Название/краткое описание дефекта
* Описание дефекта. Содержит в себе последовательность действий для обнаружения дефекта, ожидаемый результат, фактический результат.
* Вложения. Например скриншоты.

1. Этап «*Оценка и отчетность*»

* Оценка результатов тестирования на соответствие критериям выхода;
* Описание протестированной функциональности;
* Количество и критичность найденных дефектов;
* Заключение об использовании приложения.

Таблица 1 Виды тестирования ПО

|  |  |
| --- | --- |
| **Вид тестирования** | **Описание** |
| **По объекту тестирования:** | |
| Функциональное тестирование | Тестирование, основанное на анализе функциональной спецификации компонента или системы |
| Нагрузочное тестирование | Тип тестирования, проводимый с целью оценки поведения компонента или системы при возрастающей нагрузке, например количестве параллельных пользователей и/или интенсивности операций, а также определения какие ресурсы системы и в каком объеме будут при этом использованы. |
| Тестирование производительности | Процесс тестирования с целью определить производительность программного продукта. Производительность - степень, с которой система или компонент выполняет заложенные в нее функции в установленных рамках на время обработки и пропускную способность. |
| Тестирование стабильности | Тестирование, целью которого является оценка работы системы при длительной средней нагрузке |
| Тестирование удобства использования | Это метод тестирования направленный на установление степени удобства использования, обучаемости, понятности и привлекательности для пользователей разрабатываемого продукта в контексте заданных условий |
| Тестирование интерфейса пользователя | Предполагает проверку соответствия приложения требованиям к графическому интерфейсу, профессионально ли оно выглядит, выполнено ли оно в едином стиле |
| Тестирование безопасности | Стратегия тестирования, используемая для проверки безопасности системы, а также для анализа рисков, связанных с обеспечением целостного подхода к защите приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным |
| Тестирование совместимости | Подход, основной целью которого является обеспечение качественной работы конечного продукта с другими программами, операционными системами, аппаратным обеспечением и т.д. |
| **По знанию системы:** | |
| Тестирование чёрного ящика | Тестирование, функциональное или нефункциональное, без знания внутренней структуры компонента или системы |
| Тестирование белого ящика | Тестирование, основанное на анализе внутренней структуры компонента или системы |
| Статическое тестирование | Проводится без запуска программы |
| Динамическое тестирование | Проводится с запуском программы |
| **По степени автоматизированности:** | |
| Ручное тестирование | Функциональное тестирование, при котором все шаги тестов выполняются вручную |
| Автоматизированное тестирование | Процесс верификации программного обеспечения, при котором инициализация и выполнение тестов выполняется с помощью инструментов для автоматизации тестирования |
| **По степени изолированности компонентов:** | |
| Компонентное (модульное) тестирование | Метод тестирования, при котором проверяется минимально возможный для тестирования компонент, например, отдельный класс или функция. Чаще всего модульное тестирование осуществляется разработчиками |
| Интеграционное тестирование | Метод тестирования, при котором проверяются интерфейсы между компонентами, подсистемами. |
| Системное тестирование | Метод тестирования, при котором проверяется интегрированная система на ее соответствие требованиям. |
| Приемочное тестирование | Формальный процесс тестирования, который проверяет соответствие системы требованиям и проводится с целью: определения удовлетворяет ли система приемочным критериям, вынесения решения заказчиком или другим уполномоченным лицом принимается приложение или нет |
| **По уровню независимости тестирования:** | |
| Альфа тестирование | Имитация реальной работы с системой штатными разработчиками, либо реальная работа с системой потенциальными пользователями/заказчиком. Чаще всего альфа-тестирование проводится на ранней стадии разработки продукта, но в некоторых случаях может применяться для законченного продукта в качестве внутреннего приемочного тестирования. |
| Бета тестирование | Выполняется будущими пользователями для того, что бы получить обратную связь о продукте. Иногда распространяются версии с ограничениями по времени или по функциональности для некоторой группы лиц, с тем, что бы убедиться, что продукт содержит достаточно мало ошибок. |
| **По времени проведения тестирования** | |
| Регрессионное тестирование | Тестирование уже протестированной программы, которое проводится после модификации для уверенности в том, что процесс модификации не внес или не активизировал ошибки в областях, не подвергавшихся изменениям. Проводится после изменений в коде программного продукта или его окружения |
| Повторное тестирование | Тестирование, во время которого исполняются тестовые сценарии, выявившие ошибки во время последнего запуска, для подтверждения успешности исправления этих ошибок |
| **По признаку позитивности сценариев:** | |
| Позитивное тестирование | Это тестирование на данных или сценариях, которые соответствуют нормальному (штатному, ожидаемому) поведению системы. Основной целью является проверка того, что при помощи системы можно сделать то, для чего она создавалась |
| Негативное тестирование | Это тестирование на данных или сценариях, которые соответствуют нештатному поведению тестируемой системы - различные сообщения об ошибках, исключительные ситуации. Основной целью является проверка устойчивости системы к воздействиям различного рода, валидация неверного набора данных, проверка обработки исключительных ситуаций (как в реализации самих программных алгоритмов, так и в логике бизнес-правил) |
| **По степени подготовленности к тестированию:** | |
| Тестирование по документации | Подход к тестированию, при котором тестовые сценарии разрабатываются на основе целей и условий тестирования, вытекающих из требований, то есть тесты, проверяющие определенные функции или оценивающие нефункциональные атрибуты системы, такие как надежность или практичность. |
| Тестирование по тест-кейсам | Тестирование производится по заранее написанным тестовым сценариям |
| Исследовательское тестирование | Тестирование без применения документаций и тестовых сценариев. Данный вид тестирования заранее не определяется в плане тестирования, и такие тесты создаются, выполняются и модифицируются динамически, по мере необходимости. |

**Порядок выполнения работы**

1. Составить тест-план для тестирования приложения «Калькулятор».
2. Спроектировать тестовую модель (тест-кейсы) для любых трех операций (например, сложение, умножение, деление).
3. Провести тестирование по тестовой модели и тест-плану.
4. Описать найденные дефекты.
5. Составить отчет по завершению тестирования.
6. **Содержание отчета по лабораторной работе**
7. Цель работы
8. Тест-план
9. Тестовая модель
10. Дефекты
11. Отчет по завершению тестирования
12. Выводы по лабораторной работе