Предмет: Поддержка и тестирование программных модулей

Преподаватель: Мария Александровна Горбунова

Жизненный цикл ПО начинается с идеи о задумке какого – либо ПО. Конец – когда ПО выводится из эксплуатации. Стадии:

1. Анализ требований
2. Проектирование
3. Разработка/кодирование
4. Тестирование
5. Ввод в эксплуатацию и доработка

Тестирование – это процесс, позволяющий убедится в том, что в программе нет ошибок, при этом, невозможно отыскать абсолютно все ошибки в программном продукте. Из этого выходит, что тестирование – процесс, позволяющий убедится в том, что программа выполняет свое назначение.

Тестирование – наблюдение за функционированием ПО в специфических условиях с целью определения степени соответствия ПО к требованиям.

Тестирование – одна из дисциплин RUP (Методология разработки ПО). Она ориентирована на оценку качества с помощью следующих методов:

1. Поиск и документирование дефектов качества
2. Общая рекомендация относительно качества
3. Проверка выполнения основных предположений и требований на конкретных примерах
4. Проверка, что продукт функционирует так, как было запроектировано
5. Проверка, что требования были выполнены соответствующим образом

Тестирование – это проверка соответствия программы требованиям, осуществляемая путём наблюдения за её работой в специальных искусственно созданных ситуациях, выбранных с целью гарантии требуемого качества продукции.

Качество ПО – способность программного продукта при заданных условиях удовлетворять установленным или предполагаемым потребностям.

Качество ПО – весь объем признаков и характеристик программ, который относится к их способности удовлетворять установленным или предполагаемым потребностям.

Качество ПО – степень, в которой система, компонент или процесс удовлетворяет потребностям или ожиданиям заказчика или пользователя.

Модель качества

Качество ПО

Функциональность:

1. Функциональная исправность
2. Соответствие стандартам
3. Функциональная совместимость
4. Безопасность
5. Точность

Надежность:

1. Завершенность
2. Восстанавливаемость
3. Устойчивость к отказам

Удобство использования:

1. Удобство изучения
2. Понятность
3. Удобность и простота использования

Эффективность:

1. Эффективность по времени
2. Эффективность использования ресурсов

Удобство сопровождения:

1. Стабильность
2. Анализируемость
3. Контролепригодность
4. Изменяемость

Портативность:

1. Удобство установки
2. Заменяемость
3. Совместимость

ГОСТ – каталог государственных стандартов

ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программной продукции. Характеристика качества и руководства по их применению.

ГОСТ 28195-89. Оценка качества программных средств. Общие положения.

ГОСТ 28806-90. Качество программных средств. Термины и определения.

Тест – специальная искусственно созданная ситуация, выбранная определенным образом, и описание того, какие наблюдения за работой программы нужно сделать для проверки её соответствия некоторому требованию.

Тестовый случай – артефакт либо документ, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Тестовый случай используется для ручного или автоматического тестирования.

Ошибка – возникает в результате деятельности людей, связанной с разработкой ПО. Ошибки бывают в требованиях, в коде, в дизайне

Дефект – несоответствие поведения программы требованиям или ожидаемому поведению, или несоответствие документации требованиям

Отказ – проявление дефекта в ходе эксплуатации программы. Аварийный отказ – невозможность продолжать эксплуатацию программы

Верификация – процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа. Т.е. выполняются ли наши цели, сроки, задачи по разработке проекта, определенные в начале текущей фазы.

Валидация – определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе.

Цели и задачи тестирования:

1. Предоставить информацию заказчику, разработчику, руководителю проекта
2. Информация: обнаруженные несоответствия и рекомендации по их исправлению, рекомендации к усовершенствованию
3. С помощью: средств управления тестированием, отчётов, встреч

Задачи тестировщика:

* Определение миссии тестирования
* Проверка подхода к тестированию
* Проверка стабильности выпуска
* Тестирование и оценка
* Достижение приемлемого результата миссии
* Совершенствование методов и средств тестирования

Планирование тестов:

1. Определение требований к тестам
2. Оценка рисков
3. Выбор стратегии тестирования
4. Определение ресурсов
5. Создание расписания/последовательностей
6. Разработка плана тестирования
7. Определение критериев завершения и успешности тестирования

Дизайн и разработка тестов:

* Анализ объёма работ
* Определение тестовых случаев
* Обзор и оценка тестового покрытия
* Определение тесто-критичной функциональности в дизайне и модели реализации
* Запись тест-кейсов или программирование текстовых скриптов
* Создание/подготовка внешних наборов данных

Процесс выполнения тестов:

* Выполнение тестов
* Оценка выполнения тестов
* Восстановление после сбойных тестов
* Проверка результатов
* Исследование неожиданных результатов
* Запись ошибок

Оценка тестов:

* Анализ дефектов
* Оценка критериев завершения и успешности тестирования

Эффективность тестирования:

Фаган (1976) Число ошибок, обнаруженных в ходе инспекции/общее число ошибок в продукте до инспекции \* 100%

Джонс (1986) Дефекты, обнаруженные тестированием или инспекцией/дефекты, имеющиеся во время тестирования или инспекции \* 100% = Обнаруженные дефекты/обнаруженные + не обнаруженные дефекты (обнаруженные позже) \* 100%

Участники процесса тестирования:

1. Менеджер проекта – обеспечение ресурсами, координация работ
2. Разработчик, технический писатель – исправление найденных ошибок
3. Архитектор – обеспечение целостности проекта в процессе исправления ошибок
4. Интегратор (Релиз менеджер) – контроль и выпуск версий ПО
5. Аналитик – установка приоритетов, связанных с необходимостью и сложностью исправления найденных дефектов
6. Тестировщик – несет ответственность за процесс тестирования в целом

Роли участников группы тестирования:

1. Руководитель группы тестирования – ключевая роль тестировщика в группе, несет ответственность за организацию процесса тестирования в проекте, планирование и контроль действий по тестированию
2. Тестировщик – аналитик – несет ответственность за формирование тестовых спецификаций и анализ итогов
3. Разработчик тестов – несет ответственность за разработку тестов, установку и сопровождение инфраструктуры тестирования, создание стенда для проведения тестирования в соответствии с планом тестирования
4. Исполнитель тестов – несет ответственность за фактическое исполнение тестов и документирование выявленных дефектов

Определения модели жизненного цикла программной системы по ГОСТ: Модель жизненного цикла – структура, состоящая из процессов, работ и задач, включающих в себя разработку, эксплуатацию и сопровождение программного продукта, охватывающая жизнь системы от установления требований к ней до прекращения ее использования (ГОСТ 12207, 1999).

Четыре обобщенные фазы жизненного цикла:

1. Концепция (инициация, идентификация, отбор)
2. Определение (анализ)
3. Выполнение (практическая реализация, производство и т.д.)
4. Закрытие (завершение)

Циклы тестирования:

Полный цикл – обычно совпадает с интеграцией разработки или соответствует ее определенной части.

Частный цикл – проводится для конкретной сборки объекта тестирования.

Полный цикл тестирования: Начало – Планирование – Разработка – Выполнение – Анализ и отчет о результатах тестирования – Завершение.

1. Жизненный цикл
2. Тестирование
3. Качество ПО
4. ГОСТ
5. Тест/тестовый случай/ошибки/дефекты
6. Цели и задачи тестирования
7. Эффективность тестирования
8. Участники процесса тестирования
9. Роли участников процесса тестирования
10. Циклы тестирования