

Тренинги Intel Delta Course

«Дополнительные главы по Software Engineering»



Тестирование: Основные понятия, определения и идеи

Виценко А.Ю., Крюков А.К., Моренов О.А., Пряхин И.В., Семенов Д.С.,
Чиликин Е.В. Intel



Варианты Определения Дефекта

Дефект – отличие поведения программы от

- Спецификаций продукта:

Калькулятор считает, что $2 \times 2 = 5$

- Ожиданий пользователя, не определенных спецификацией продукта:

Невозможность вывести документ на печать в программе обработки документов

Дефект или нет?

- FAT32 не может работать с файлами размером больше 4ГБ
- Не можем перекодировать фильм за долю секунды
- Программа не удаляет за собой временные файлы
- «Стандартный» калькулятор не умеет считать синусы

Понятие Программы

Программа – набор инструкций и данных, исполнимых компьютером

Тогда, **Дом** – набор стройматериалов, собранных согласно чертежам

Дом – постройка, предназначенная для жилья людей
В таком определении есть цель и заинтересованные лица (**stakeholder**)

Заинтересованное лицо (**stakeholder**) - любой человек, на которого влияет:

- Успех или неудача проекта создания программы
- Действия или бездействие программы

Программа –

- Общение
- между людьми или компьютерами
- Разнесенными во времени и пространстве
- Содержащее инструкции, которые могут быть исполнены компьютером

Дефект

Дефект - атрибут программы, который:

- Уменьшает ценность для «важных» заинтересованных лиц
- Увеличивает ценность для «враждебных» заинтересованных лиц
- Без соответствующей компенсирующей пользы

“Any threat to the value of the product to any stakeholder who matters.”

--

James Bach

Цена Дефекта ПО

Чем раньше найден дефект – тем меньше цена его исправления

Дефект стоит денег:

- Возможное уменьшение продаж продукта
- Возможное возмещение убытков покупателям
- Увеличение расходов на службу поддержки
- Судебные издержки
- Потеря имиджа компании

Дефект стоит времени:

- Поиск причин проблемы
- Исправление кода
- Повторное тестирование продукта

Примеры «Дорогих» Ошибок

- Взрыв на советском газопроводе в результате диверсии ЦРУ (1982)
- Ошибка в управляющем коде рентгеновского аппарата «Therac-25» привела к смерти 5 пациентов (1980)
- «Пинг смерти». Неверно сформированный пакет пинга приводил к «падению» операционной системы (1995/96)
- Веерное отключение электричества в Северной Америке из-за ошибки в управляющем ПО (2003)
- Использование целых чисел в расчетах механизма детонации ракет «Патриот»
- Все ключи, сгенерированные версиями OpenSSL, выпущенными 2006-2008, уязвимы из-за ошибки в генераторе случайных чисел
- Зависание бортовой аппаратуры израильских самолетов при полетах над Мертвым Морем
- Патч онлайн игры «Eve Online» удалил файл boot.ini на компьютерах всех пользователей (2007)

Способы повышения качества ПО

Проактивные:

- Повышение квалификации разработчиков
- Внедрение эффективных процессов разработки ПО
 - Проектная методология разработки
 - Использование стандартов кодирования
 - Методологии контроля изменений
 - ...
- Выравнивание требований к продукту с ожиданиями заказчика или потребителя

Реактивные:

- Инспекции кода
- Использование статических анализаторов кода
- Динамическое тестирование
- Исправление найденных ошибок

Зачем тестировать?

- Демонстрация отсутствия ошибок в программе?
 - Демонстрация корректности исполнения программой предусмотренных сценариев?
 - Убедиться, что программа выполняет свое предназначение?
 - Убедиться, что у программы нет «побочных эффектов»?
 - Понять ценность программы для потребителя?
- ...

Что такое хороший тест:

Тест, подтверждающий отсутствие ошибки?

Тест, находящий ошибку?

Понятие Тестирования

Тестирование это:

- «Процесс поиска всех мыслимых ошибок или недостатков в рабочем продукте» [Майерс, 1979]
- Проверка работоспособности продукта
- Оценка качества продукта
- Информирование руководства о качестве продукта с целью улучшения управления рисками

Тестирование – процесс исследования программы с целью нахождения в ней угроз качеству продукта

“Testing can be used to show the presence of bugs, but never their absence!” [E.W. Dijkstra]

«Тестирование может быть использовано для демонстрации наличия ошибок, но никогда для их отсутствия» [Дейкстра]

Требования к тестированию

- Повторимо
- Систематизировано
- Документировано

Полное Тестирование

Возможно ли обнаружить все ошибки в программе?

Пример: проверим перемножение 2 целых чисел

Число комбинаций - $2^{32} * 2^{32}$

Если один тест идет 0.000001 сек, то все тесты идут 213503982 дня

Невозможно создать набор тестов, гарантирующий отсутствие ошибок

Экономическая целесообразность создания полного набора тестов



**В общем случае невозможно найти все ошибки в программе.
Но это не означает, что к этому не нужно стремиться.**

Краткая История Тестирования

До 1956 года тестирование=отладка.

1957-1978 Тестирование – демонстрация выполнения требований.
Осознание невозможности «исчерпывающего» тестирования.

1979–1982 «Период разрушения», цель – поиск ошибок.

1983–1987 Цель – оценка продукта и измерение качества.
Появление первых систем автоматизированного тестирования

1988-сейчас. Соответствие спецификациям, нахождение дефектов,
предотвращение дефектов.

Принципы Тестирования (Майерс)

1. Необходимая часть тестового сценария – определение ожидаемого результата
2. Программист должен избегать тестирования собственных программ
3. Организации следует избегать тестирования собственных программ
4. Вдумчиво изучайте результаты каждого теста
5. Тестовые сценарии должны разрабатываться для некорректных входных данных, так же как и для правильных и ожидаемых
6. Определение, что программа делает то, что должна – лишь половина дела. Другая половина – проверка, что программа не делает того, чего не должна
7. Избегайте исключения тестовых сценариев
8. Не планируйте тесты в предположении, что ошибки не будут найдены
9. Вероятность нахождения ошибок в секции программы прямо пропорциональна количеству уже найденных там ошибок
10. Тестирование – исключительно творческая и интеллектуальная задача

Мифы О Тестировании

- Цель тестирования – демонстрация отсутствия ошибок в продукте
- Есть код, который нет нужды тестировать
- Тестирование – это просто. Все способны тестировать
- Тестирование – случайный, не систематизируемый процесс
- Тестированию не нужно учиться
- Тестирование = отсутствие карьерного роста
- Продукт может быть протестирован полностью
- Тестирование позволяет найти все дефекты продукта
- Все дефекты должны быть исправлены
- Автоматизированный тест эквивалентен аналогичному выполненному человеком

Материалы и источники

1. The Art of Software Testing. Glenford J. Myers
2. Methods of software Quality Improvement. Amir Sejminovic, Farshid Atachi
3. Блог Майкла Болтона (<http://www.developsense.com/blog>)
4. Блог «255» ступеней (<http://www.developsense.com/blog>)
5. Википедия:
 1. http://en.wikipedia.org/wiki/List_of_software_bugs
 2. http://en.wikiversity.org/wiki/Software_testing/history_of_testing
6. <http://www.exforsys.com/tutorials/testing/software-testing-myths.html>
7. TestingGeek <http://www.testinggeek.com/what-is-not-software-testing-exploring-myths>
8. Alan Myrvold blog <http://testapprentice.com/2011/02/21/software-testing-myths/>
9. <http://www.wired.com/software/coolapps/news/2005/11/69355?currentPage=all>