



Верификация ПО

Тестирование в жизненном цикле программного проекта

К.А.Кулаков

Петрозаводск — 2017

Команда проекта

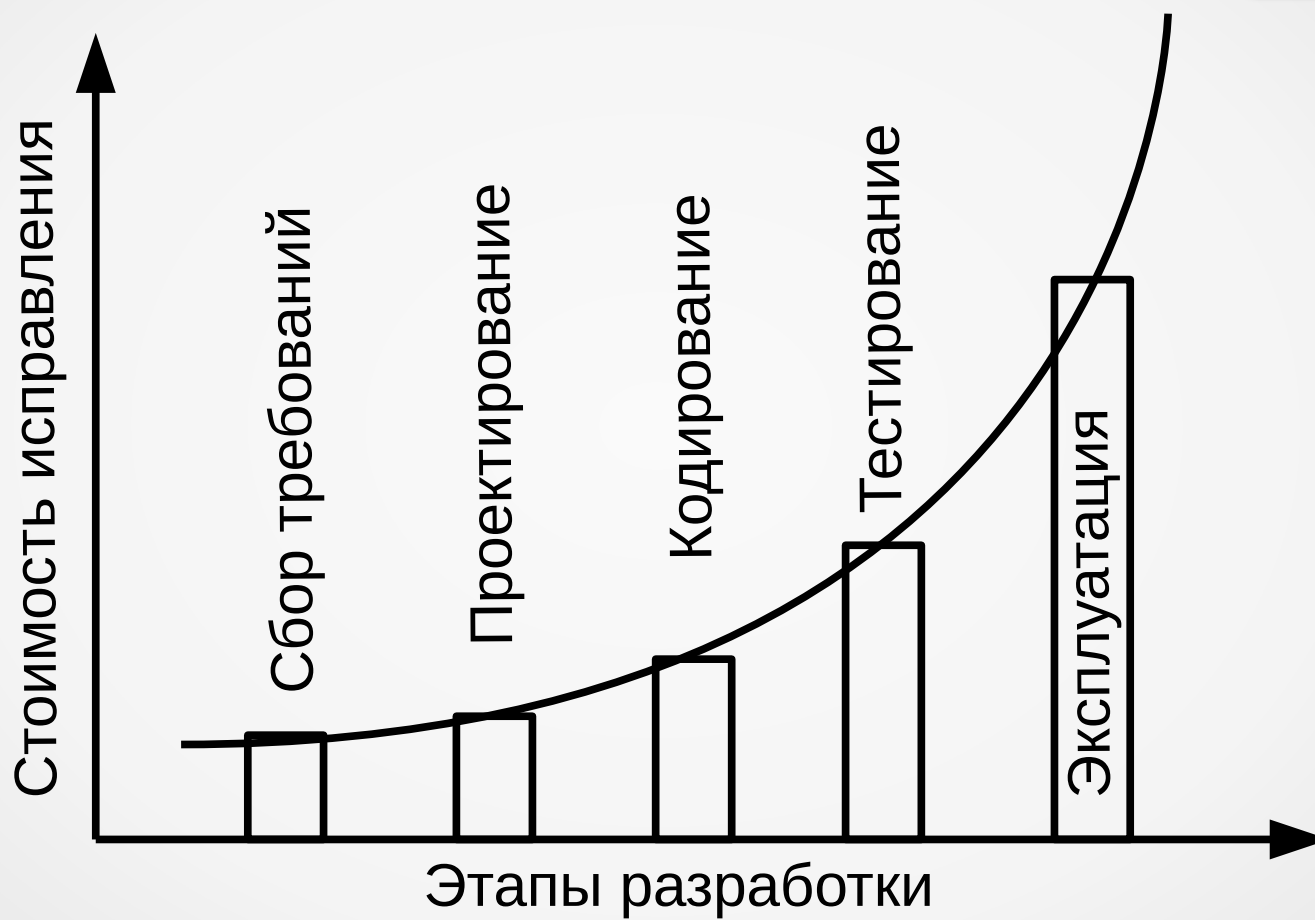
- Руководитель проекта (качество, планирование)
- Проектировщики
 - Разработчик архитектуры
 - Специалист по анализу предметной области
 - Специалист по анализу человеческого фактора, или специалист по эргономике
 - Программист пользовательского интерфейса
 - Ведущие программисты
- Менеджер по маркетингу (соответствие долгосрочной стратегии, конкурентоспособность, маркетинг)
- Техническая поддержка (минимизация вопросов от пользователей)
- Технические писатели (writers) — это члены группы документирования (documentation group), разрабатывающие руководство пользователя и справку
- Тестировщики (testers)
- другие специалисты: по компьютерной графике, надежности, защите, аппаратному обеспечению, а также юристы, бухгалтера и т.д.

Этапы жизненного цикла проекта

- Планирование и анализ требований
- Проектирование
- Кодирование и написание документации
- Тестирование и исправление недостатков
- Сопровождение (после выпуска) и усовершенствование

!!!Тестирование и модификация на всех этапах!!!

Стоимость исправления ошибки



Планирование

- Определение целей
- Анализ требований
- Определение функциональных характеристик программного продукта

- Тестирование на этапе планирования
 - "тестируются" идеи.
 - К анализу привлекаются специалисты по маркетингу, руководители проекта, главные конструкторы и специалисты по анализу человеческого фактора.
 - Члены группы тестирования участвуют в этой работе очень редко.
 - Группа аналитиков читает черновики проектных документов и собирает информацию, которая может оказать помощь в их оценке и дальнейшем планировании.
 - способы: сравнительный анализ, дискуссионные группы и обследование объекта.
 - Результаты каждой из этих процедур могут привести к значительному пересмотру существующих планов.

Планирование

- Цели анализа требований
 - Адекватны ли эти требования?
 - Полны ли они?
 - Совместимы ли требования между собой?
 - Выполнимы ли они?
 - Разумны ли они?
 - Поддаются ли они тестированию?

Проектирование

- Внешний дизайн
 - одна из самых важных частей, могут появиться вопросы от пользователя при использовании
- Внутренняя структура
 - Проектирование программной архитектуры (декомпозиция, п/системы, модули, интерфейсы)
 - Проектирование организации данных
 - Описание алгоритмов
 - Моделирование (модели, прототипы)

Проектирование

- Тестируются только идеи
- Тестирование на этапе проектирования
 - Действительно ли проект хорош?
 - Соответствует ли проект требованиям?
 - Полон ли проект?
 - Достаточно ли он реалистичен?
 - Хорошо ли описана в проекте подсистема обработки ошибок?

Проектирование

- Способы тестирования
 - Совещания аналитиков
 - Обзорное совещание. (демонстрация)
 - Инспекционное совещание. (анализ и проверка)
 - Рецензионное совещание. (вопросы)
 - Анализ псевдокода
 - поиск лишних частей, анализ обращений и т.д.

Реализация

- Последовательное написание кода
- Первичная проверка разработчиком
- План реализации в соответствии со стратегией сборки
- Статический анализ кода и документации
 - Соответствие спецификации языка программирования и стандартам
 - Нет дорогостоящего запуска
 - Статический анализ не заменяет динамического тестирования

Тестирование

- Разработка тестов белым ящиком
 - Направленность тестирования
 - Полный охват кода
 - Управление потоком
 - Отслеживание целостности данных
 - Внутренние граничные точки
 - Тестирование, определяемое выбранным алгоритмом

Тестирование

- Разработка тестов черным ящиком
 - проверка работы сложных объектов;
 - проверка работы на некорректных данных;
 - тестирование с точки зрения пользователя;
 - создание тестов параллельно с кодом.

Тестирование

- Готовая система — сложный объект
 - Очень трудно выявить источник ошибки.
 - Трудно организовать исправление ошибок.
 - Процесс тестирования плохо автоматизирован.
- Разбиение сложного объекта на простые
 - проверка простых объектов (модульное тестирование)
 - проверка взаимосвязей (интеграционное тестирование)

Тестирование

- Стандартная процедура тестирования
 - Планирование (стратегия, серии тестов, ...)
 - Приемочное тестирование (проверка работоспособности)
 - Проверка стабильности программы (сколько циклов может выдержать)
 - Функциональное и системное тестирование, сверка и аттестация продукта (сверка с требованиями)
 - Бета-тестирование (пользователи)
 - Тестирование целостности готового продукта и тестирование распространяемых копий (пустые пакеты, диски, отсутствие нужных компонент)
 - Окончательная приемка и сертификация

Тестирование

- Примеры тестов
 - Лабораторные испытания (наблюдение за работой)
 - Граничные условия
 - Производительность (наиболее частые операции)
 - Переходы между режимами (состояния)
 - Эксплуатация в реальном режиме (проверка в условиях рабочего места)
 - Нагрузочные испытания (активность, объем входных данных, ресурсы)
 - Многопользовательская и многозадачная работа
 - Обработка ошибок
 - Защита
 - Совместимость и преобразование форматов данных (в т.ч. со старыми версиями)
 - Аппаратные конфигурации
 - Установка и обслуживание
 - Эффектные тесты (показать нестабильность публике)

Тестирование

- Оценка качества работы разработчиков и тестировщиков
- Регрессионное тестирование: повторный запуск тестов после внесения модификаций
- Псевдоотладка: намеренное включение нескольких ошибок в код, оценка качества тестирования
- Мутационное тестирование: внесение маленького изменения и проверка обнаружения в тестах

Тестирование

- Метрики покрытия кода тестами
 - Критерий охвата функций (Function coverage): каждая функция вызывается хотя бы раз.
 - Критерий охвата строк (Statement coverage): самый слабый, каждая строка должна выполняться.
 - Критерий охвата ветвлений (Decision / Branch coverage): более основательный, каждое ветвление проверяется по всем направлениям.
 - Критерий охвата условий (Condition coverage): более строгий, проверка всех составляющих логического условия (каждое атомарное булево выражение приняло значения и "истина" и "ложь").
 - Критерий охвата параметров (Parameter Value coverage): проверка всех значений параметров метода.
 - Критерий охвата путей (Path coverage): все возможные пути в коде были пройдены.
 - Критерий охвата циклов (Loop coverage): Все циклы исполнялись 0, 1, ..., N раз.

Сопровождение

- На сопровождение программного обеспечения затрачивается 67% его общей стоимости.
 - 20% бюджета сопровождения тратится на исправление ошибок
 - 25% уходит на адаптацию продукта к новому аппаратному обеспечению и новой программной среде
 - 6% тратится на исправление документации
 - 4% тратится на повышение производительности
 - 42% тратится на внесение изменений и усовершенствований
 - 3% на другие нужды