

Введение в тестирование



Profit from the Cloud

01 Июля 2014

Стельмашенко Светлана

Содержание:

- Введение
- Цикл разработки ПО
- Методологии разработки ПО

Введение.



Что, зачем и почему



Тестирование?

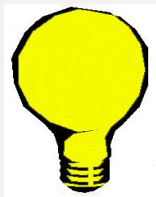
- Исследование
- Валидация
- Верификация
- Систематизация
- Общение
- Контроль
- Понимание бизнес-процессов
- Участие в дизайне системы
- Целенаправленное истощение нервной системы девелоперов и ПМ-ов



Цикл разработки ПО

- — это путь от идеи до поддержки готового продукта.
- Чем более отлажены каждая из стадий цикла и координация между ними, тем эффективнее работает компания, тем выше качество и тем счастливее пользователи.

Обычно это выглядит так:



- Дизайн + доки /
- Путь к цели
- Кодирование
- Тестирование + фикс
- Релиз
- Идея
- Цель

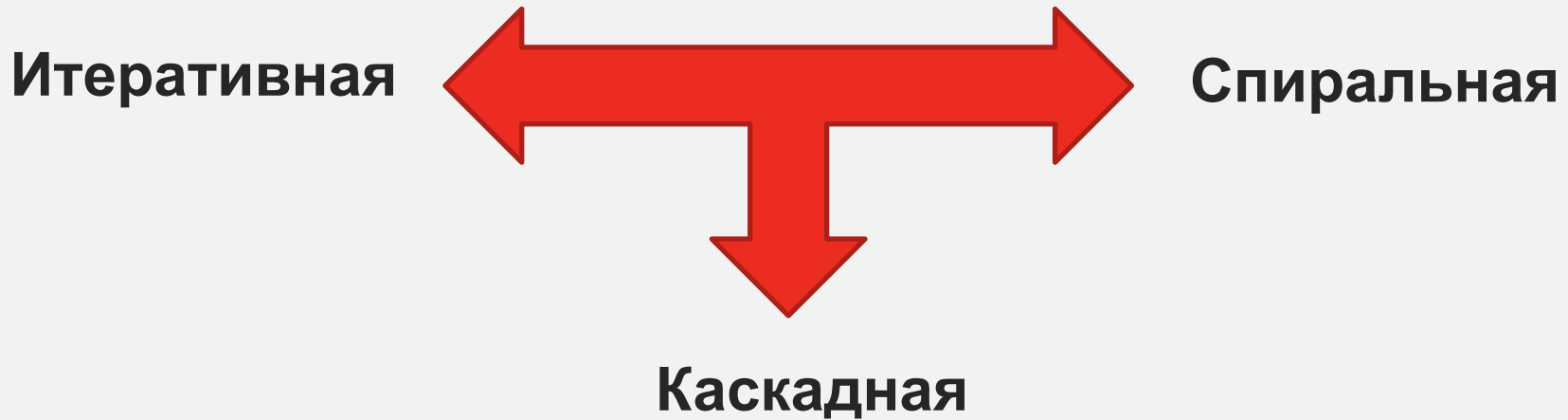




Задачи инженера контроля качества:

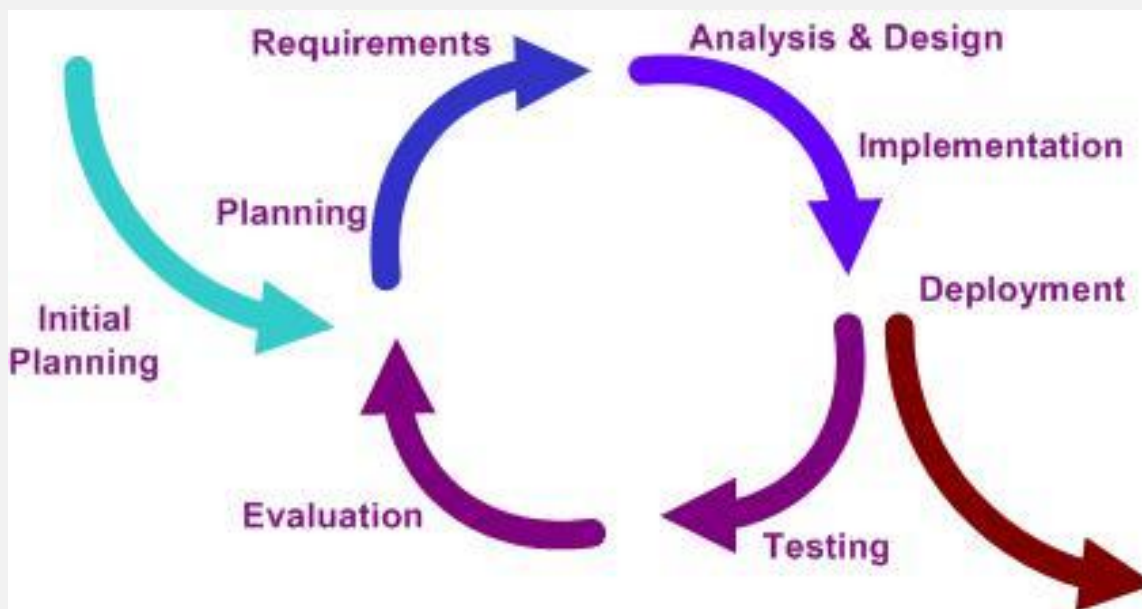
- - определение функционала продукта на основании анализа рынка и отзывов от клиентов.
- - статическое тестирование (требований, спецификаций, документации), взаимодействие с ПМ-и.
- - составление и согласование плана тестирования с ПМ-и разработчиками
- - ручное и автоматизированное тестирование.
- - разработка инструментов для внутреннего использования.
- - мониторинг качества текущего билда по результатам прохождения тестов
- - оперативная помощь клиентам
- - взаимодействие с партнёрами

Методологии разработки ПО (модели процесса)



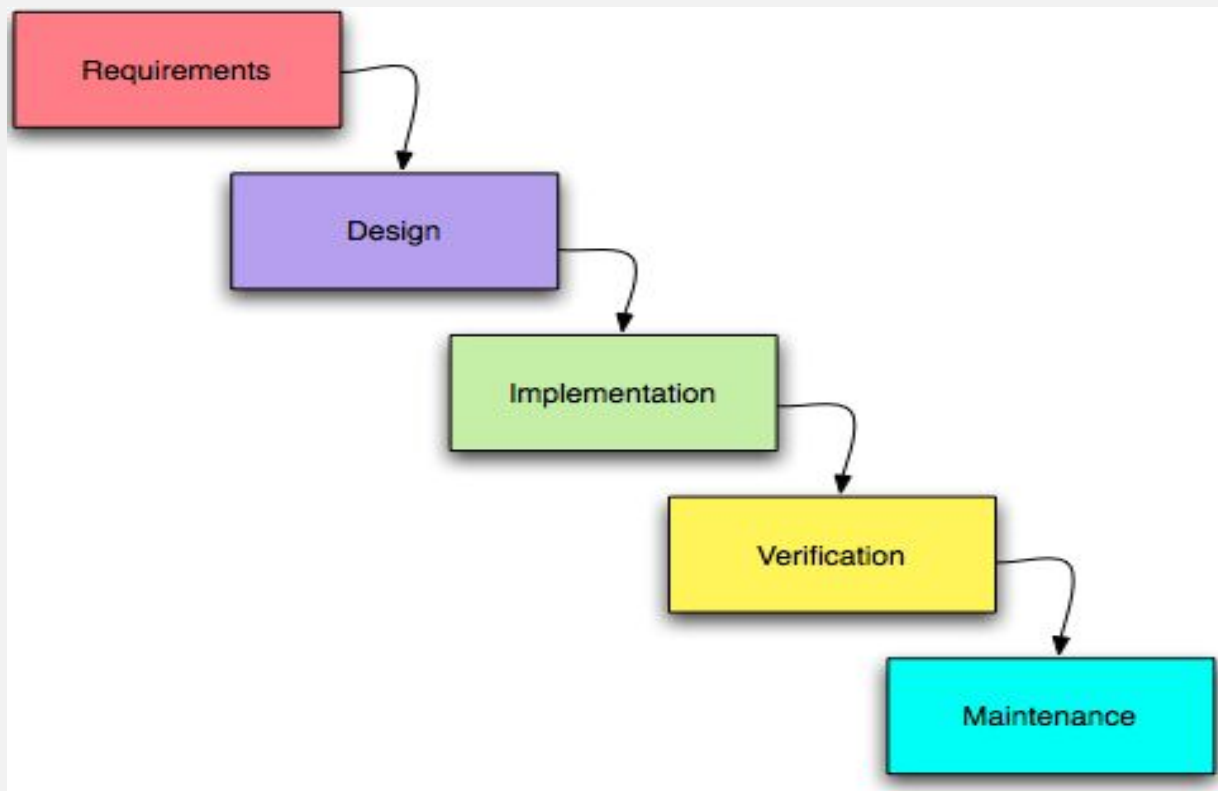
Итеративная (*agile*)

Работа ||-но с непрерывным анализом результатов.
Проект в каждой фазе проходит повторяющийся цикл:
Планирование - Реализация - Проверка - Оценка.



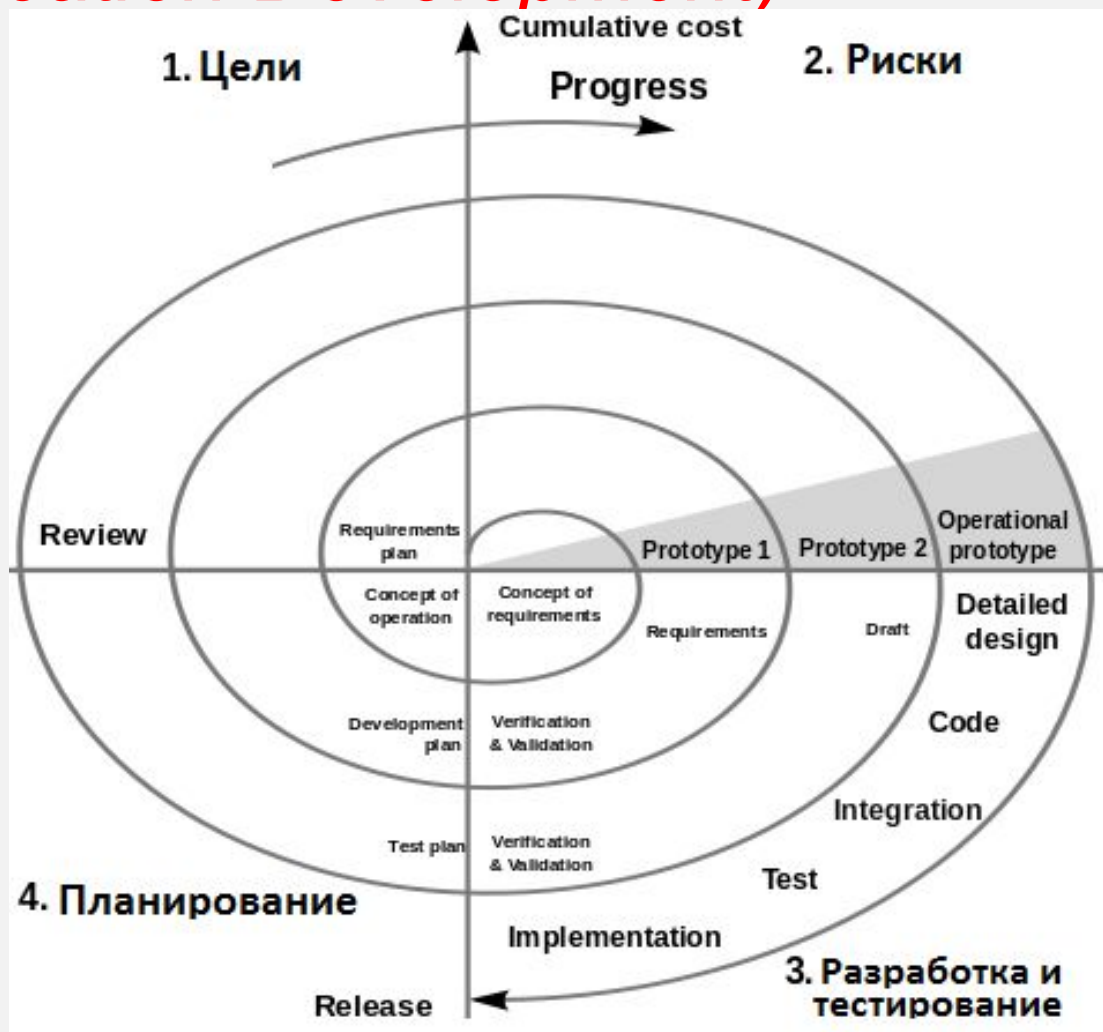
Каскадная (*waterfall*)

переход от одной фазы разработки к другой происходит только после полного и успешного завершения предыдущей фазы. Переходов / перекрытия этапов нет.



Спиральная *RAD (Rapid Application Development)*

- Постадийное прототипирование, на каждой стадии уточняются цели и характеристики проекта, акцент на анализ и проектирование



Методы и виды тестирования

1. По объекту тестирования:

- Функциональное тестирование (functional testing)
- Тестирование производительности (performance testing)
 - Нагрузочное тестирование (load testing)
 - Стресс-тестирование (stress testing)
 - Тестирование стабильности (stability / endurance / soak testing)
- Юзабилити-тестирование (usability testing)
- Тестирование интерфейса пользователя (UI testing)
- Тестирование безопасности (security testing)
- Тестирование локализации (localization testing)
- Тестирование совместимости (compatibility testing)

2. По знанию системы:

- Тестирование чёрного ящика (black box)
- Тестирование белого ящика (white box)
- Тестирование серого ящика (grey box)

3. По степени автоматизации:

- Ручное тестирование (manual testing)
- Автоматизированное тестирование (automated testing)
- Полуавтоматизированное тестирование (semiautomated testing)

4. По степени изолированности компонентов:

- Компонентное (модульное) тестирование (component/unit testing)
- Интеграционное тестирование (integration testing)
- Системное тестирование (system/end-to-end testing)

5. По времени проведения тестирования:

- Альфа-тестирование (alpha testing)
 - Дымовое тестирование (smoke testing)
 - Тестирование новой функциональности (new feature testing)
 - Подтверждающее тестирование (confirmation testing)
 - Регрессионное тестирование (regression testing)
 - Приёмочное тестирование (acceptance testing)
- Бета-тестирование (beta testing)*

* С привлечением клиентов

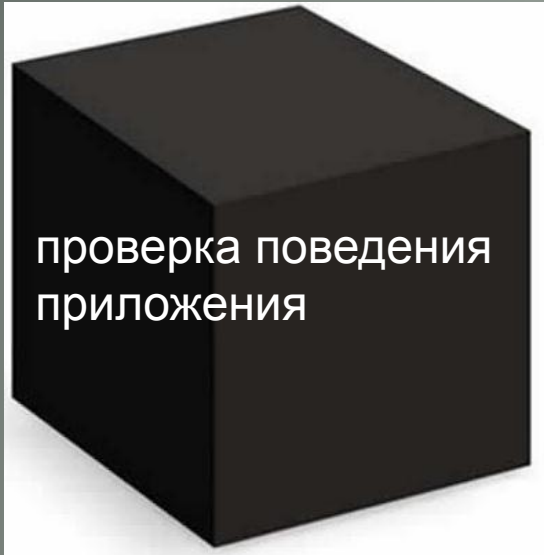
6. По признаку позитивности сценариев:

- Позитивное тестирование (positive testing)
- Негативное тестирование (negative testing)

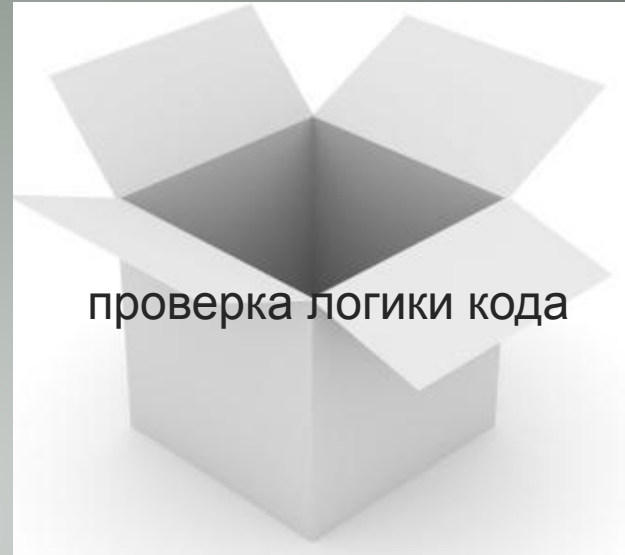
7. По степени подготовленности к тестированию:

- Тестирование по документации (formal testing)
- Тестирование ad-hoc или интуитивное тестирование (ad hoc testing)

Black vs White



- Эквивалентное разбиение
- Анализ граничных значений
- Диаграммы причинно-следственных связей
- Системное тестирование.
- Функциональное тестирование
- Регрессионное тестирование
- Тестирование безопасности.
- Тестирование перегрузок.
- Тестирование производительности.
- Тестирование удобства использования



- Unit тесты
- Ввод неверных значений
- Тестирование обработки ошибок
- Утечка памяти
- Комплексное тестирование модулей
- Мутационные тесты
- Исследование покрытия

- Вопросы?