

## Вопросы для повторения

- Что такое тестирование?
- Что делает тестировщик?
- Что такое качество?
- Что значит ТЗ и его назначение ?
- Что такое баг(Определение) ?

Ручное тестирование

## Урок 2



# Место тестирования в процессе разработки ПО

Цикл разработки ПО. Цикл тестирования ПО.  
Классификация видов и направлений  
тестирования.

# План урока

1. Общие знания о циклах разработки ПО.
2. Общие знания о цикле тестирования ПО.
3. Что мы можем тестировать?
4. Какие виды тестирования бывают?

К концу урока мы будем знать разные виды тестирования, этапы в процессе тестирования.



Жизненный цикл ПО начинается с идеи.

Жизненный цикл ПО – непрерывный процесс, который начинается с принятия решения о необходимости создания продукта и заканчивается с его полным изъятием из эксплуатации.

# Требования к ПО



# Разработка и тестирование



## Эксплуатация и поддержка





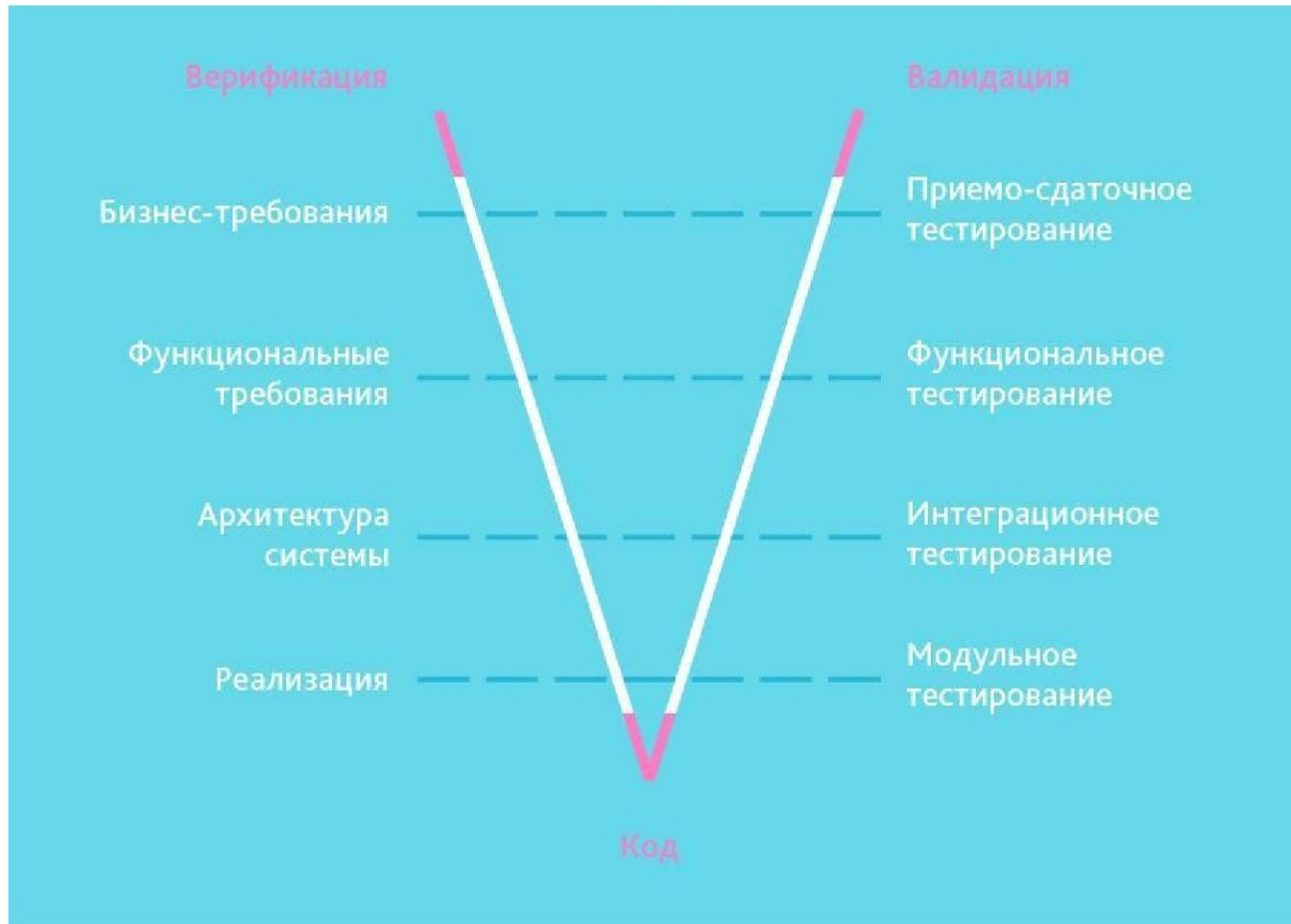
# Каскадная модель разработки



Цикл  
разработки  
программного  
обеспечения



# V-модель



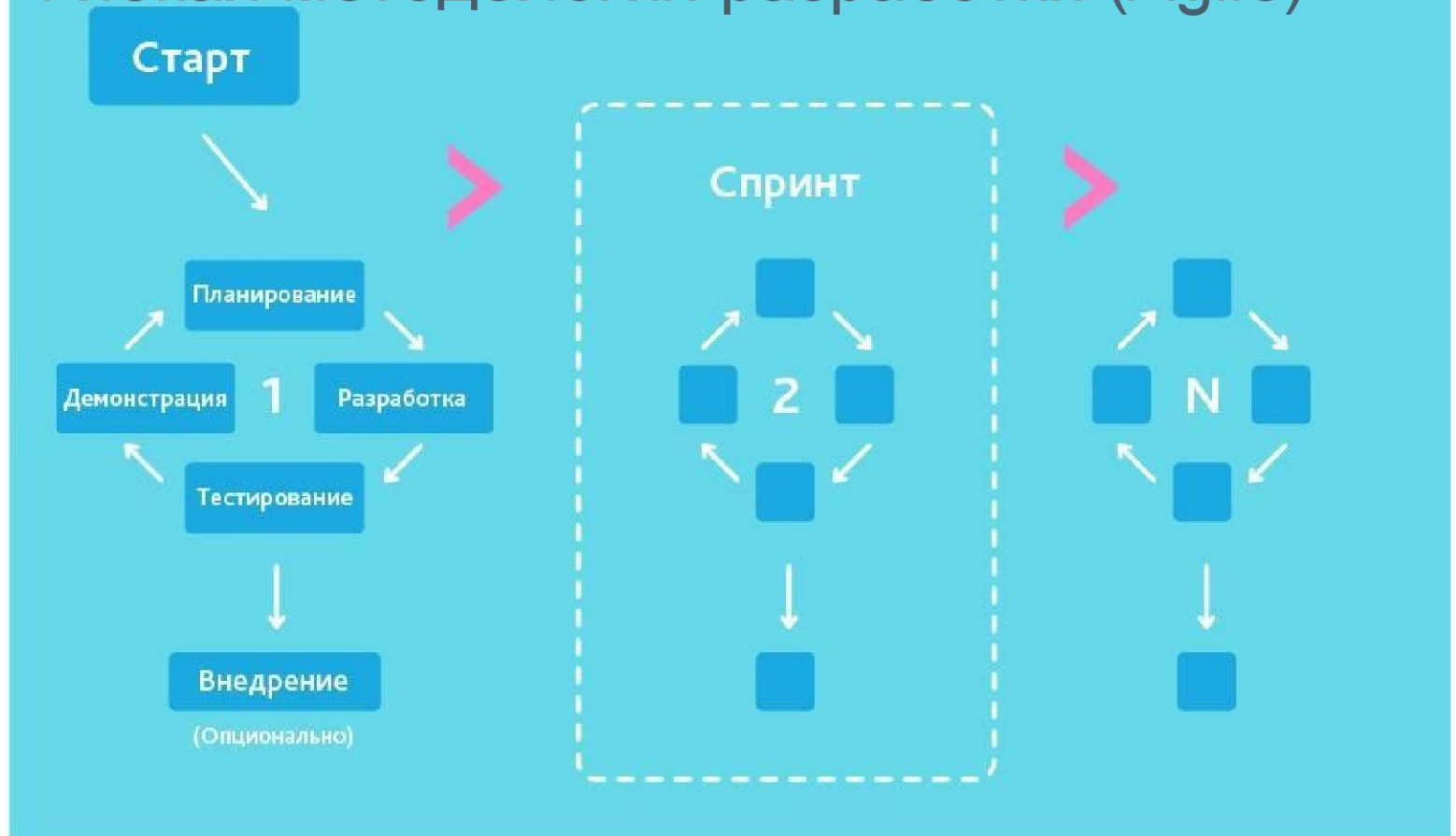
## Инкрементная модель



## Итеративная модель



# Гибкая методология разработки (Agile)



Спринт – итерация, ограниченная по времени, по окончании которого должна быть получена новая рабочая версия продукта.

Спринты имеют одинаковую продолжительность на протяжении всей жизни продукта.

# Спиральная модель



# Жизненный цикл тестирования



# Этап планирования



1. Что мы тестируем? А что не тестируем?
2. Как много работы?
3. Есть ли все необходимое для тестирования?
4. Выбрана ли стратегия тестирования?

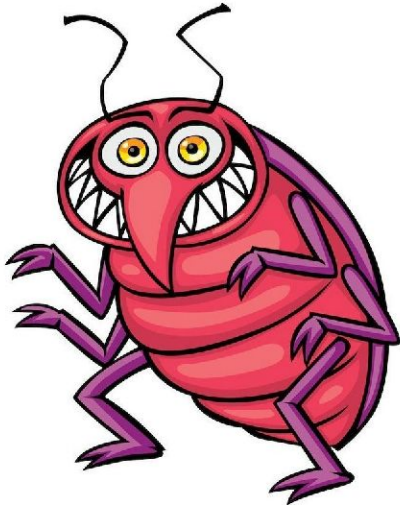


# Этап разработки тестов (тест-дизайн)



1. Анализируем документацию.
2. Придумываем тесты.
3. Фиксируем тесты.

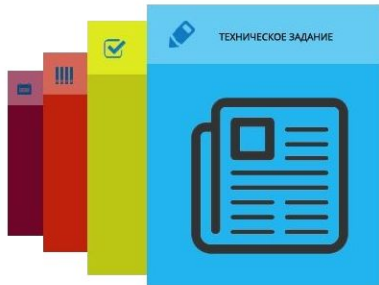
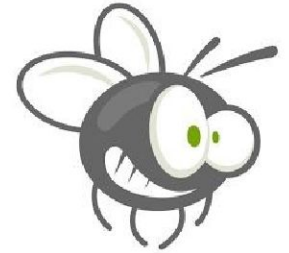
# Выполнение тестов



# Оценка результатов тестирования



# Что тестируют кроме ПО?



- Требования к ПО и документацию;
- Дизайн;
- Код.

```
bool IsBukva(char symbol)
{
    switch(symbol)
    {
        case 'a': return 1; break;
        case 'b': return 1; break;
    }
}
```

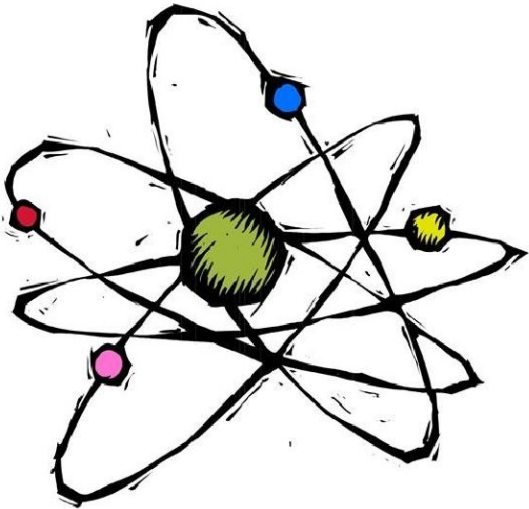
# Тестирование требований. Завершенность



- “Пароли должны храниться в зашифрованном виде”  
*Каков алгоритм шифрования?*
- “Экспорт осуществляется в форматы PDF, PNG и т.д.”  
*Что понимается под “и.т.д.”?*
- “... см. выше”  
*А куда – выше? Насколько выше?*



# Тестирование требований. Атомарность



“Когда пользователь входит в систему, должно отображаться приветствие; когда пользователь вошел в систему, должно отображаться имя пользователя; когда пользователь выходит из системы, должно отображаться прощание”.

*Все три ситуации заслуживают отдельных и куда более детальных описаний требований.*

# Тестирование требований. Непротиворечивость



“После успешного входа в систему пользователя, не имеющего права входить в систему...”

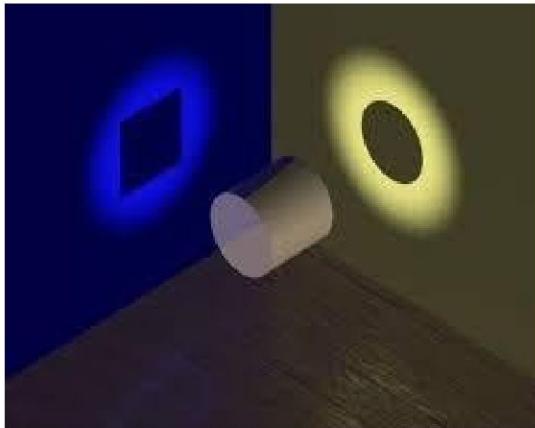
*Как он успешно вошел в систему, если не имел такого права?*

“712.а Кнопка “Close” всегда должна быть красной» и «36452.x Кнопка “Close” всегда должна быть синей»

*Так все же красной или синей?*



# Тестирование требований. Однозначность



“Приложение должно поддерживать передачу больших объемов данных”.

*Насколько больших?*

“В случае необходимости оптимизации передачи больших файлов система должна эффективно использовать минимум оперативной памяти, если это возможно”.

*Звучит красиво, но непонятно*

# Тестирование требований. Выполнимость



“Настройка параметров для подключения к базе данных должна поддерживать распознавание символов из жестов, полученных с устройств трехмерного ввода”.

*Долго, дорого и бесполезно для конечного пользователя.*

“Система поиска должна заранее предусматривать все возможные варианты поисковых запросов и кэшировать их результаты”

*В принципе нереализуемо.*

# Тестирование требований. Обязательность



- Требование было добавлено “на всякий случай”, хотя реальной потребности в нем не было и нет.
- Требованию выставлены неверные значения приоритета по критериям важности и/или срочности.
- Требование устарело, но не было переработано или удалено.

# Тестирование требований. Прослеживаемость



- Требования не пронумерованы, не структурированы, не имеют оглавления и работающих перекрестных ссылок.
- При разработке требований не были использованы инструменты и техники управления требованиями.
- Набор требований неполный, носит отрывочный характер с явными пробелами.

# Тестирование требований. Модифицируемость



- Изменение требования порождает противоречивость.
- Требования изначально противоречивы. В такой ситуации внесение изменений усугубляет ситуацию.
- Требования представлены в неудобной для обработки форме.

# Тестирование требований. Проранжированность



- Неверно выставлен приоритет по важности для конечного пользователя.
- Неверно выставлен приоритет по стабильности требований, которые в самое ближайшее время могут претерпеть кардинальные изменения.
- Неверно выставлен приоритет по срочности, нарушая последовательность реализации функций ПО, ожидаемых заказчиком.



# Тестирование требований. Корректность и проверяемость



- Опечатки.
- Неаргументированные требования к дизайну и архитектуре.
- Плохое оформление текста и сопутствующей графической информации, грамматические, пунктуационные и иные ошибки в тексте.
- Неверный уровень детализации.



# Что тестируют кроме ПО?



- Требования к ПО и документацию;
- Дизайн;
- Код.

```
bool IsBukva(char symbol)
{
    switch(symbol)
    {
        case 'a': return 1; break;
        case 'b': return 1; break;
    }
}
```

# Тестирование дизайна

Где я?



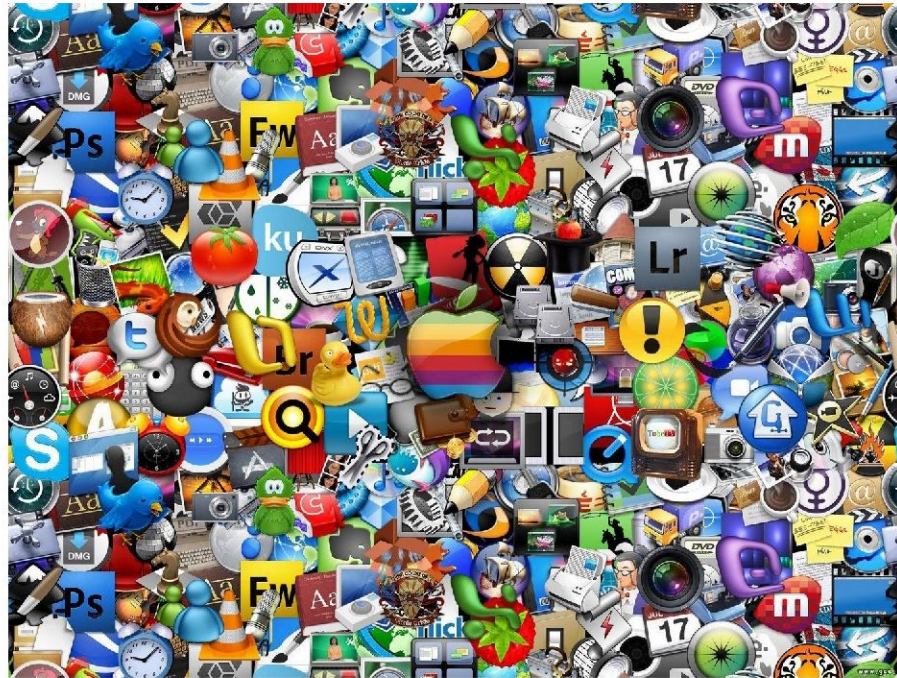
Тестирование дизайна

Что делать дальше?



# Тестирование дизайна

## Лавина



# Тестирование дизайна

Что я сделал?





# Тестирование кода



- Модульное тестирование (юнит-тесты).
- Интеграционное тестирование.
- Ревью программного кода (Code review).

# Виды тестирования





По запуску кода на исполнение ↗

Статическое

Тестирование требований

Code review

Динамическое

Модульное тестирование

Интеграционное тестирование

Приёмочное тестирование

Статическое – без запуска ПО.

Динамическое – с запуском ПО.

## По доступу к коду и архитектуре приложения

Метод белого ящика

Юнит-тестирование

Полный доступ к коду и архитектуре

Метод чёрного ящика

Тестирование через интерфейс пользователя

Тестирование по требованиям

Метод серого ящика

Частичный доступ к структуре приложения (БД)

## По степени автоматизации ↑

### Ручное тестирование

Выполнение тестов человеком

Тестирование происходит медленно

Человека нельзя клонировать

Человек сам модернизирует тесты

### Автоматизированное тестирование

Тесты выполняет программа

Тесты выполняются быстро

Одни и те же тесты можно выполнять на разных конфигурациях

Автоматизированные тесты нужно поддерживать и модернизировать

## По уровню детализации приложения

Модульное тестирование

Тестирование отдельного модуля ПО

Юнит-тестирование

Интеграционное тестирование

Тестирование связи между модулями

Тестирование связи между частями ПО

Системное тестирование

Тестирование функциональных требований системы

Модульное – тестируется часть приложения.

Интеграционное – тестируются связи между модулями приложения или связи между системами.

Системное – тестируется система в целом.

## По степени важности тестируемых функций

Дымовое тестирование

Минимальный набор тестов на явные ошибки

Критерий продолжения более глубокого тестирования

Тестирование критического пути

Тестирование часто используемого функционала

Расширенное тестирование

Проверка всей функциональности по требованиям



## По принципам работы с приложением ↗

Позитивное тестирование

Тестирование допустимых и корректных данных

Негативное тестирование

Тестирование на некорректных данных

Тестирование непредусмотренных ситуаций

## По цели тестирования ↑

Функциональные  
виды тестирования

Функциональное тестирование

Арифметические операции в калькуляторе

Операция покупки в интернет-магазине

Тестирование безопасности

Роли в системе

Защита от взлома

Защита от скриптов

Тестирование взаимодействия

Использование внешних сервисов

Использование внешних компонентов и систем

Нефункциональные  
виды тестирования

Связанные  
с изменениями в коде

# По цели тестирования ↗

## Функциональные виды тестирования

- Нагрузочное тестирование
  - Производительность ПО под нагрузкой
  - Изменение скорости отклика под нагрузкой
- Стресс-тестирование
  - Высокая интенсивность выполнения операций в ПО
- Тестирование стабильности и надёжности
  - Многочасовая эксплуатация
  - Поиск утечек памяти
- Объёмное тестирование
  - Увеличение размера данных в БД
- Инсталляционное тестирование
  - Установка
  - Обновление
  - Удаление
- Юзабилити-тестирование
  - Удобство использования
  - Понятность
  - Привлекательность
- Тестирование на отказ и восстановление
  - Восстановление сети
  - Восстановление после сбоя
- Конфигурационное тестирование
  - Различные ОС
  - Различные драйвера

## Связанные с изменениями в коде

## По цели тестирования ↑

Функциональные  
виды тестирования

Нефункциональные  
виды тестирования

Связанные  
с изменениями в коде

Дымовое тестирование

Регрессионное тестирование

Тестирование сборки

Санитарное тестирование

Подтверждение работоспособности  
ранее разработанной функциональности

Тестирование по ранее разработанным  
тест-кейсам, чек-листам

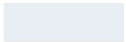
Аналог дымового тестирования

Приёмка новой версии в тестирование

Может быть глубже чем дымовое тестирование

Подмножество регрессионного тестирования

Проверка работоспособности части приложения



# Практика