



# Agenda

- Классификация видов тестирования
- Agile Testing Quadrants
- Ещё несколько определений
- Практика

# тестирования:

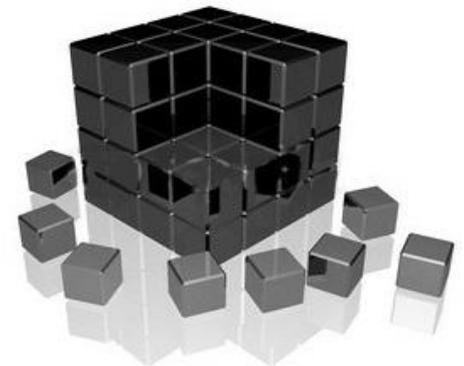
- По знанию внутренностей системы
- По объекту тестирования
- По субъекту тестирования
- По времени проведения тестирования
- По критерию “позитивности” сценариев
- По степени изолированности тестируемых компонент
- По степени автоматизированности тестирования
- По степени подготовленности к тестированию

# СИСТЕМЫ

- Тестирование черного ящика  
(Black Box Testing)
- Тестирование серого ящика  
(Grey Box Testing)
- Тестирование белого ящика  
(White Box Testing)

# Ящика

- ❑ Black Box (closed box, opaque box, behavioral) Testing
  - Тестировщик производит тестирование, не имея информации о том, как устроена система изнутри
  - Идеи для тестирования идут от предполагаемого поведения пользователей

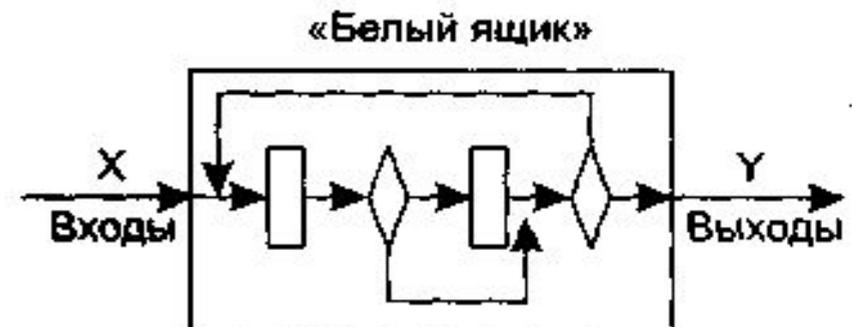


# 1.2. Тестирование белого ящика

## □ White (Glass box, structural) Box Testing

- Противоположность методу Черного ящика
- Тестирование производится на основании информации, как устроена система изнутри

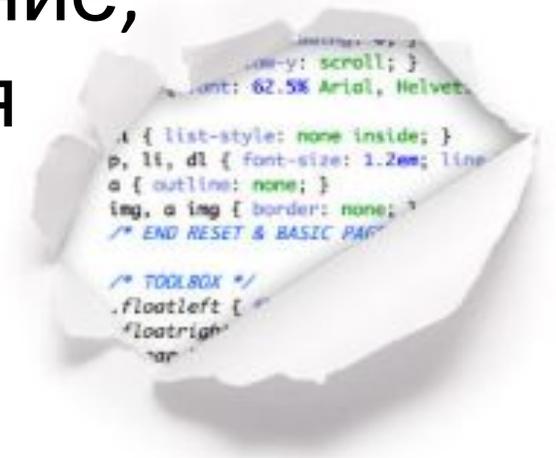
- Обычно производится самими программистами



# 1.3. Тестирование серого ящика

## □ Grey Box Testing

- Подход сочетает в себе как белый, так и черный ящики
- Это информированное тестирование, ориентированное на пользователя



## 2. По объекту тестирования

- Функциональное тестирование
- Тестирование пользовательского интерфейса
- Тестирование локализации
- Тестирование скорости и надежности
- Тестирование безопасности
- Тестирование удобства использования
- Тестирование совместимости
- Тестирование инсталляции
- Тестирование документации

# тестирование

- Один из процессов жизненного цикла программного продукта
- Проверяет соответствие реализованных функций требованиям и просто ожиданиям пользователя.
- Проверяется каждая из функций приложения и все они в комплексе
- Исследуются все сценарии использования

# тестирование

- Преимущества функционального тестирования:
  - имитирует фактическое использование системы;
- Недостатки функционального тестирования:
  - возможность упущения логических ошибок в программном обеспечении;
  - вероятность избыточного тестирования.

# Non-functional testing

- Non-functional testing is the testing of a software application for its non-functional requirements.
- Non-functional testing includes:
  - Load testing
  - Localization testing
  - Performance testing
  - Recovery testing
  - Security testing
  - Usability testing

## 2.2. Тестирование пользовательского интерфейса

### □ Для стандартных объектов UI

- Text box
- Text area
- Numeric field
- Drop down list
- Combo box
- Check box
- Date field
- существуют стандартные проверки... Какие?

## 2.3. Тестирование локализации

- Тестирование локализации – это различные проверки, связанные с адаптацией приложения для пользователей из других стран



## 2.4. Тестирование скорости и надёжности

- Это тестирование для определения времени задержки в обработке как функции от приложенной нагрузки
- Это тестирование является составной частью процессов жизненного цикла программного продукта и проводится с целью получения статистических и аналитических данных, характеризующих ПП в условиях, приближенных к реальной эксплуатации

## 2.4.4 Тестирование скорости и надёжности

□ Тестирование скорости и надёжности можно разделить на:

- Load testing
- Stress testing
- Performance testing
- Volume testing
- Stability testing

# Load testing

- A type of performance testing conducted to evaluate the behavior of a component or system with increasing load, e.g. numbers of parallel users and/or numbers of transactions, to determine what load can be handled by the component or system.

# Stress testing

- A type of performance testing conducted to evaluate a system or component at or beyond the limits of its anticipated or specified work loads, or with reduced availability of resources such as access to memory or servers.

# Volume testing

## /Performance testing

- Задачей объемного тестирования является получение оценки производительности при увеличении объемов данных в базе данных приложения.

# Stability testing

- Задачей тестирования стабильности (надежности) является проверка работоспособности приложения при длительном (многочасовом) тестировании со средним уровнем нагрузки.

## 2.5. Тестирование безопасности

- Целью тестирования безопасности является определение правильности ограничения доступа к различным объектам системы согласно пользовательским правам



## 2.5. Тестирование безопасности

- Тестирование контроля доступа
- Тестирование авторизации пользователей
- Тестирование валидации ввода
- Тестирование надежности шифрования данных
- Тестирование правильности обработки ошибок
- Тестирование на переполнение буфера
- Тестирование конфигурации сервера

## 2.6. Тестирование удобства использования

- ❑ Usability testing оценивает приложение с точки зрения конечного пользователя и может включать в себя такие проверки как:
  - Человеческий фактор
  - Эстетика
  - Непротиворечивость пользовательского интерфейса
  - Онлайновая и контекстно-зависимая помощь
  - Пользовательская документация
  - Обучающие материалы



## 2.6. Тестирование удобства использования

- Внешний вид экранов и меню, функций помощи, стиля иконок, расположение их в удобном порядке для подтверждения того, что система может быть изучена и использована с минимальными усилиями
- Часто эта группа тестов относится к категории некритичных, но когда речь идет, например, о рыночном готовом продукте – пренебрегать удобством эксплуатации весьма опасно

## 2.6. Usability testing. Assessment

- efficiency
- accuracy
- recall
- emotional response

## 2.7. Тестирование совместимости

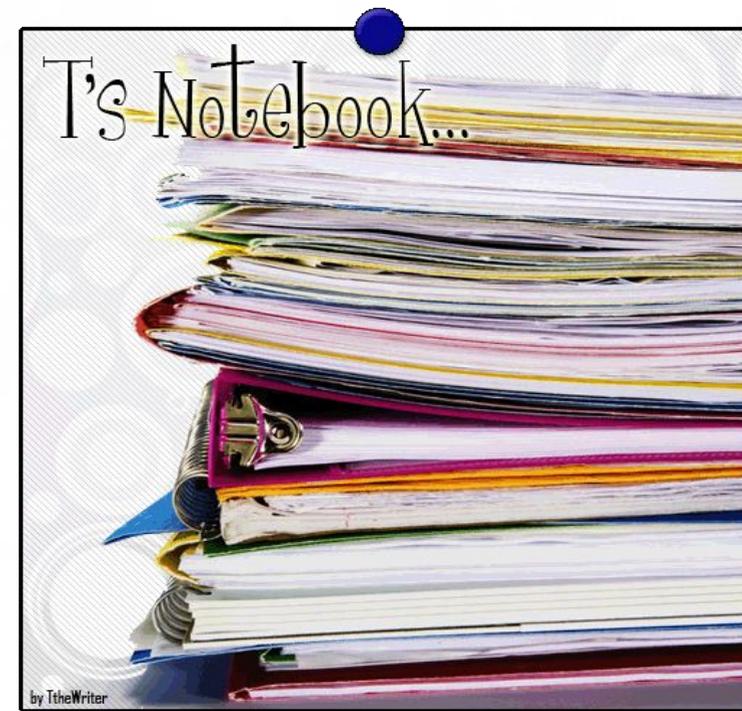
- Тестирование на совместимость (compatibility testing) проводится для проверки того, что программный продукт работает без ошибок на конфигурациях с различными операционными системами, браузерами, почтовыми клиентами
- Кроме того, может проверяться совместимость с hardware, etc.

# тестирование

- Режимы инсталляционного тестирования (installation testing):
  - Install
  - Uninstall or Remove
  - Modify (Add/Remove components)
  - Repair
  - Upgrade

# документации

- Цель тестирования документации – определение, насколько точна, полна, достоверна и ясна документация, которая поставляется с ПП
- Поставляемая документация:
  - Инструкция пользователя
  - Инструкция по установке
  - Help
  - И так далее ...



# 3. По субъекту тестирования

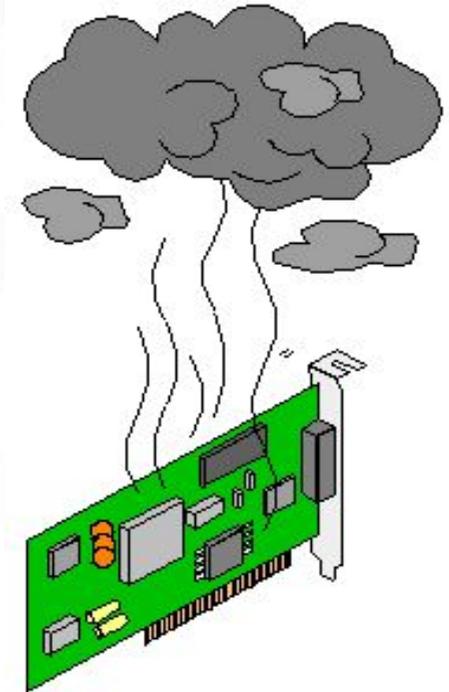
- **Альфа-тестирование**  
Проводится сотрудниками компании
- **Бета-тестирование**  
Вид тестирования, который производится обычно потенциальными пользователями (проверка в “боевых условиях”)

# тестирования

- Тест приемки (Smoke test)
- Тестирование новых функций
- Регрессионное тестирование
- Тест сдачи (acceptance test)
- Maintenance testing

# Smoke test

- ❑ Короткий тест, проверяющий основную функциональность программного продукта и его работоспособность, ограниченный по времени, по результатам которого ведущий инженер по тестированию принимает решение о приемке версии ПП на тестирование



# New feature testing

- Целью является проверка того, что новая функциональность работает корректно

# Regression testing

- Регрессионное тестирование – повторное проведение тестов для проверки того, что изменения, внесенные в программу, не повлияли на функционал, который не изменялся.

Regression:  
"when you fix one bug, you  
introduce several newer bugs."

- Не путать с retesting!



# Full regression test

- Полный регрессионный тест включает в себя все ранее уже проверенное
- Может проводиться на этапе, когда продукт уже заявлен как готовый к поставке
- Может включать в себя перепроверки ранее описанных багов

# Sanity testing

- Узконаправленное тестирование, достаточное для доказательства того, что конкретная функция работает согласно заявленным в спецификации требованиям.
- Является подмножеством регрессионного тестирования.
- Используется для определения работоспособности определенной части приложения после изменений, произведенных в ней или окружающей среде.
- Обычно выполняется вручную.

# Acceptance testing

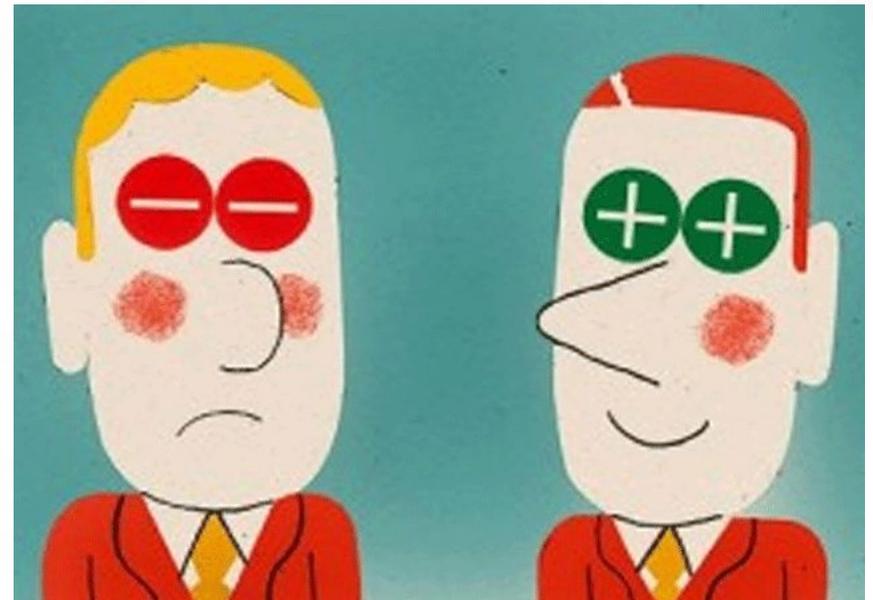
- ❑ Приёмо-сдаточное тестирование - это набор тестов, по которому осуществляется приемка продукта (возможно заказчиком)

# Maintenance testing

- Тестирование системы во время ее эксплуатации**

## 5. По критерию позитивности сценариев

- Позитивное тестирование
- Негативное тестирование



# Позитивное тестирование

- ❑ Класс тестов, которые проверяют, что программа делает то, что должна делать (результатом ожидается УСПЕХ)

# Негативное тестирование

- ❑ Класс тестов, которые проверяют, что программа НЕ делает то, что НЕ должна делать (ожидается НЕУСПЕХ в результате)

# 0. По степени изолированности компонент

- Модульное тестирование (unit)
- Компонентное тестирование (component)
- Интеграционное тестирование (integration)
- Системное тестирование (system)

# Unit testing

- ❑ Тестирование на самом нижнем уровне (unit – часть исходного кода, которая не включает никаких вызываемых подпрограмм или функций). Выполняется в основном разработчиками

# Component testing

- ❑ **Тестирование отдельных компонентов (модулей) общей системы**

# Integration testing

- ❑ Тестируется взаимодействие отдельных модулей между собой (In the small)
- ❑ In the big – тестируется взаимодействие системы с внешним окружением

# Integration testing

- Bottom Up Integration (снизу вверх)
- Top down integration (сверху вниз)
- “Big bang” integration (большой взрыв)

# System testing

- ❑ Тестирование системы в целом для обнаружения общих системных багов, таких как потеря ресурсов, синхронизация и временные проблемы, конфликты разделенных ресурсов.

# СИСТЕМЫ

- Ручное
- Автоматизированное
- Смешанное\полуавтоматическое

# Ручное

- Выполняется без привлечения средств автоматизации
- Выполняется, обычно, по подготовленным тест кейсам



# тестирование

- Выполняется с использованием специализированных программных продуктов
- Требуется высокая квалификация тестировщиков и навыки программирования



oe



## 8. По подготовленности к тестированию

- Тестирование по тест кейсам (documented testing)
- Интуитивное тестирование (ad hoc testing)

# Documented testing

- Тестирование по уже разработанной тестовой документации
- Выполняемые тесты определены заранее

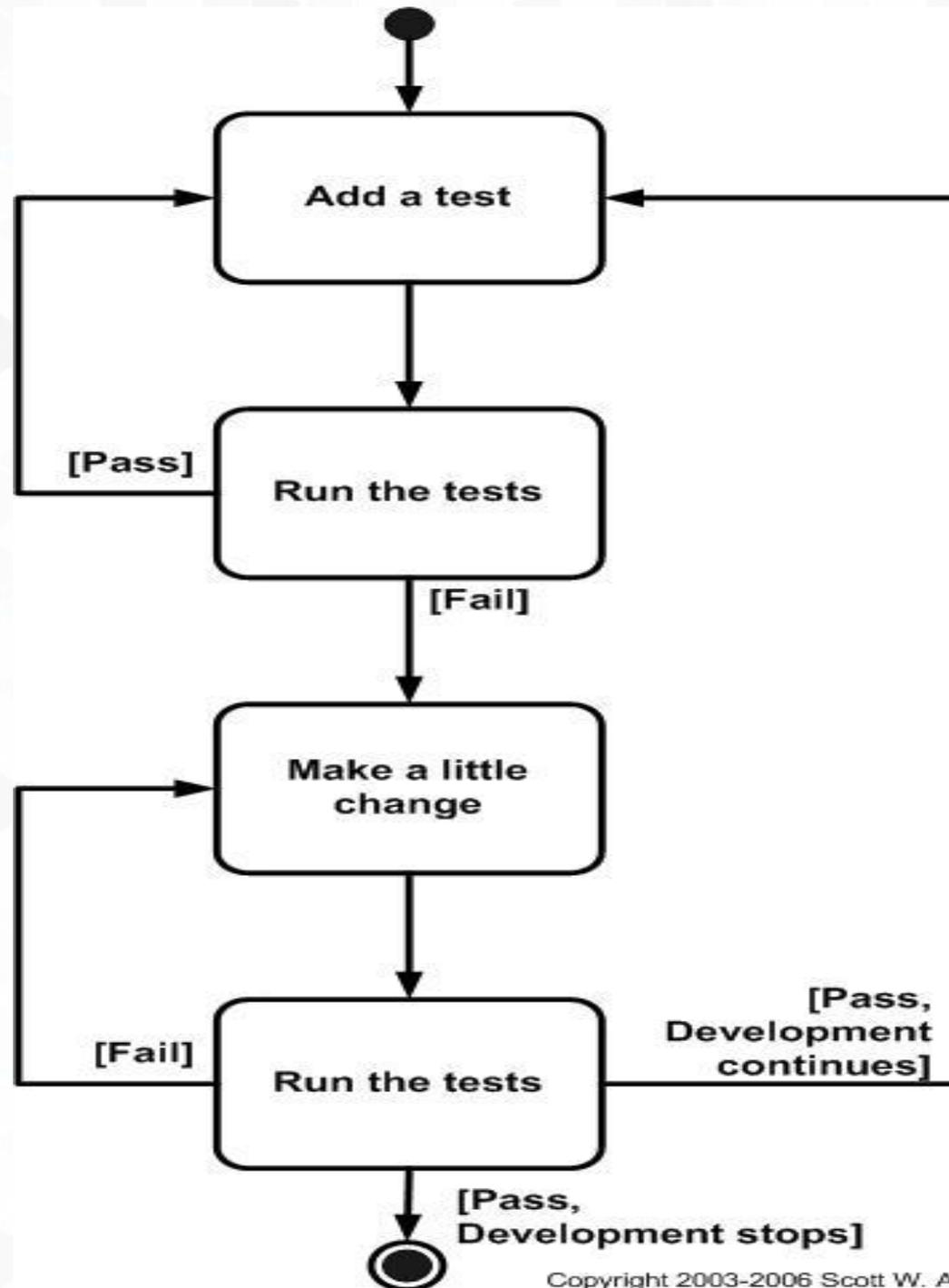
# Ad hoc testing

- ❑ Тестирование может происходить без сценария, когда тестировщик бессистемно перебирает различные варианты работы системы

# Agile Testing Quadrants (ATQ)



# TDD



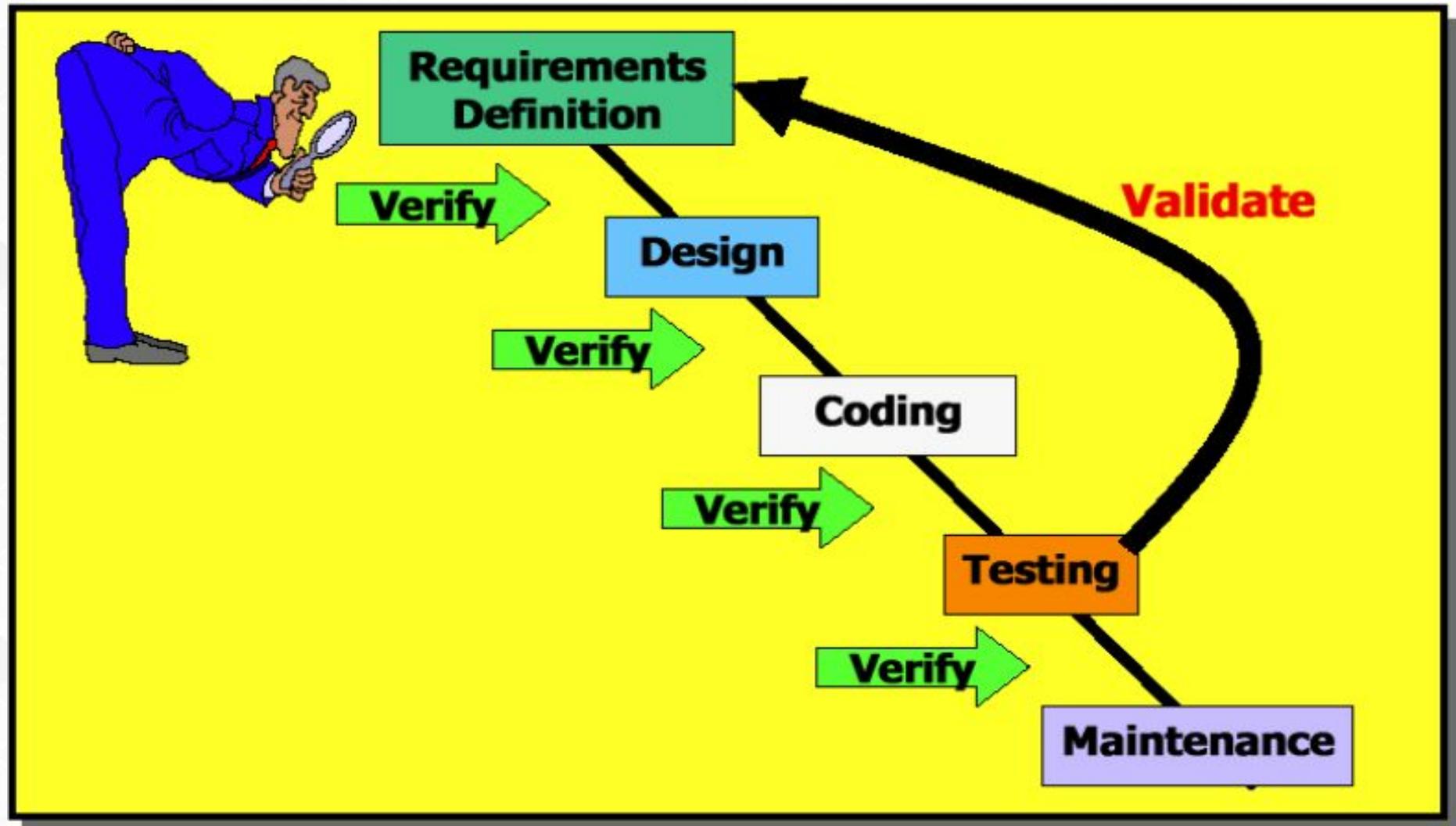
# Верификация (Verification)

- Верификация – процесс проверки того что **любой созданный проектом артефакт** соответствует своим специфическим требованиям, т.е. *проверяет что **артефакт** создан правильно*
- Целью верификации является достижение гарантии того, что верифицируемый объект (требования, документация или программный код) соответствует требованиям и удовлетворяет проектным спецификациям и стандартам.
- **Тестирование – один из методов верификации ПО**

# Валидация (Validation)

- Валидация – процесс проверки того что продукт или его компоненты соответствует своему предполагаемому предназначению, находясь в предполагаемых условиях
- Валидация проверяет, что создан правильный продукт, и в итоге проверяет соответствие продукта ожиданиям заказчика
- Хорошо проведённое тестирование – обнаруживает **несоответствия** продукта ожидаемым потребностям пользователя
- Основной результат валидирования – **Улучшение** или запрос на изменение

# Verification & Validation



# Практика

- Протестировать карандаш, распределить кейсы по видам тестирования.

# тестирования:

- По знанию внутренностей системы
- По объекту тестирования
- По субъекту тестирования
- По времени проведения тестирования
- По критерию “позитивности” сценариев
- По степени изолированности тестируемых компонент
- По степени автоматизированности тестирования
- По степени подготовленности к тестированию

