

# Лекция 1. Введение в Software Quality Engineering



## ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2016

Павел Степанов

Старший преподаватель

Кафедра компьютерной математики и программирования ГУАП

# 1. Введение

- Качество можно не только измерять, но и создавать
- Обеспечение качества – комплексный и сложный процесс
- Наличие контроля качества существенно увеличивает шанс проекта на успех (Capers Jones. Estimating Software Costs).
- Управление качеством является вариантом управления рисками
- Реальность, зачастую, печальна
  - Контроля качества нет
  - Либо он выполняется вручную неквалифицированным персоналом

## 2. Содержание

- Понятие качества
- Понятие тестирования
- Управление качеством как управление рисками
- Процесс тестирования

### 3. Определение качества

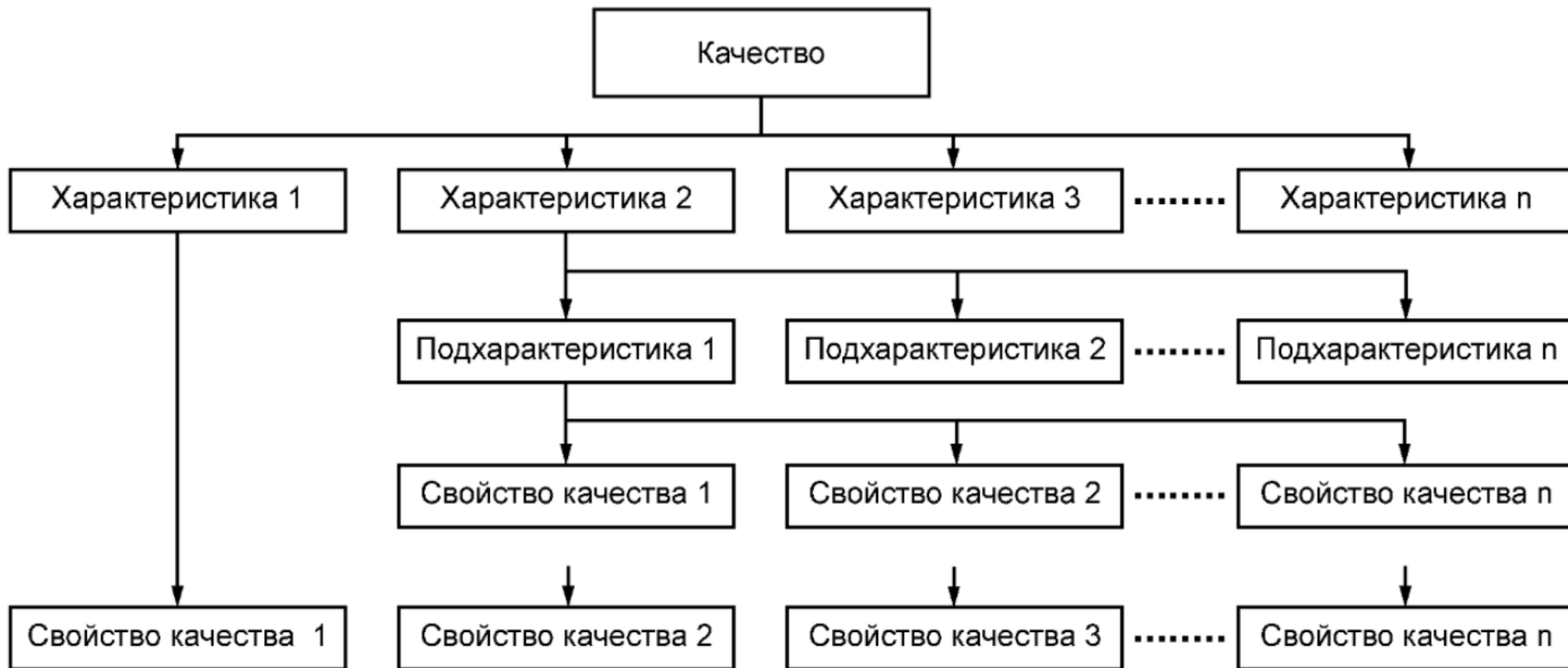
- ГОСТ Р ИСО/МЭК 25010-2015

- Качество - степень удовлетворения системой заявленных и подразумеваемых потребностей различных заинтересованных сторон

- Утилитарное определение

- Качество – это оценка готовности продукта к выпуску (переходу на следующий этап жизненного цикла)

# 4. Модели качества



# 5. Типовые модели качества

- Модель качества при использовании
  - Результативность
  - Производительность
  - Удовлетворенность
  - Свобода от риска
  - Покрытие контекста
- Модель качества продукта
  - Функциональная пригодность
  - Уровень производительности
  - Совместимость
  - Удобство использования
  - Надежность
  - Защищенность
  - Сопровождаемость
  - Переносимость
- Модель качества данных

## 6. Тестирование

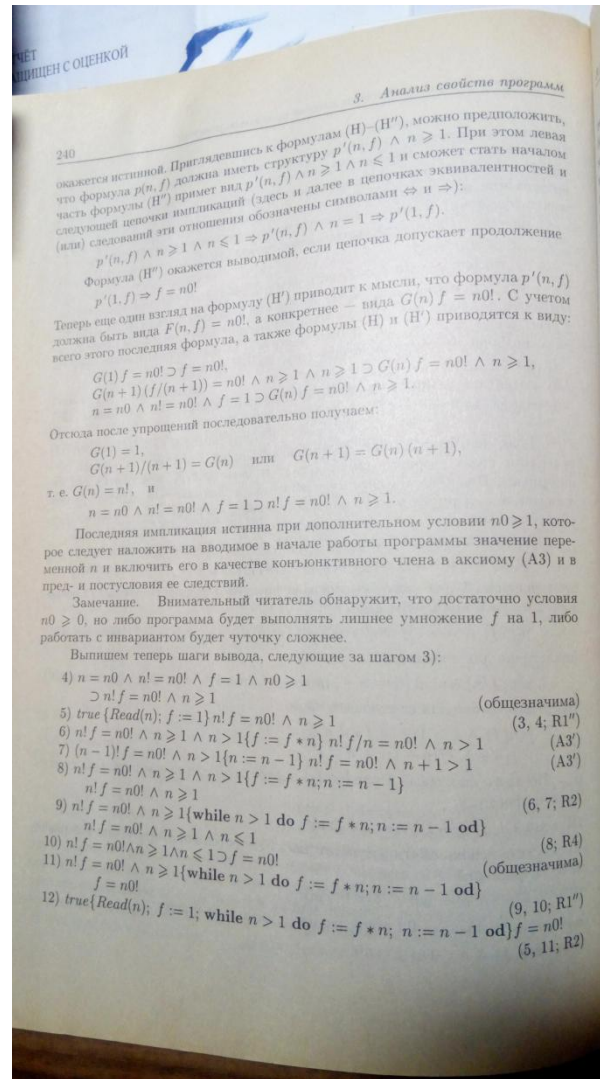
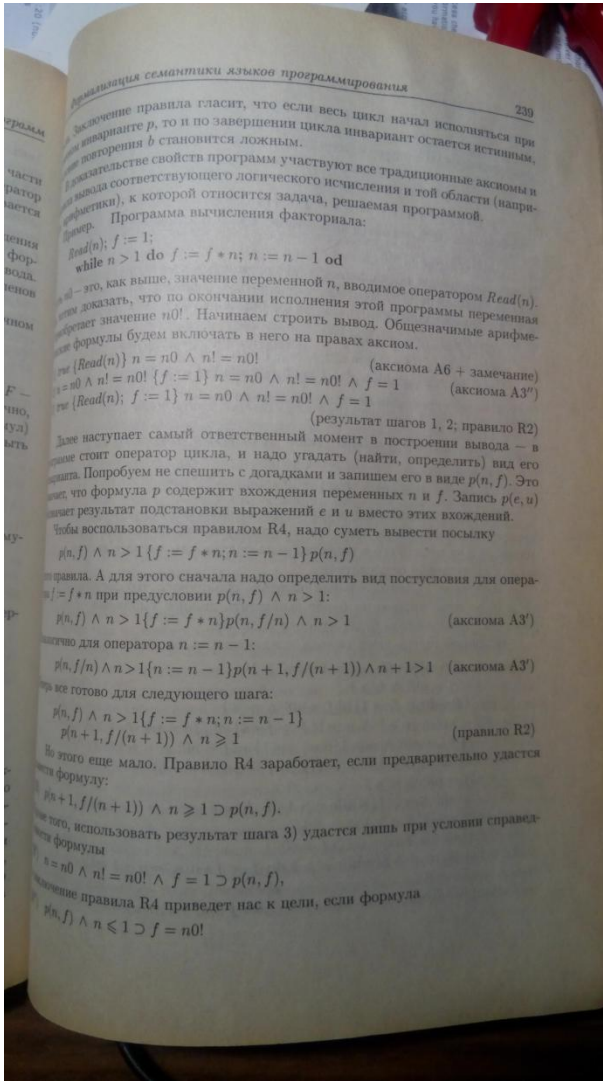
- Тестируемость (testability): Степень, с которой объективный и физически реализуемый тест может быть спроектирован для определения того, что требование выполняется (ГОСТ Р ИСО/МЭК 12207-2010)
- Тестирование программного обеспечения (Software Testing) - проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. [IEEE Guide to Software Engineering Body of Knowledge, SWEBOK, 2004]
- Фактически, тестирование – процесс измерения качества.

# 7. История вопроса

- Формальное доказательство свойств программы
- Исчерпывающее тестирование



# 8. Критика формального доказательства



формальное  
доказательство  
свойств  
факториала  
(Функции из двух  
строк).  
Лавров С.С.  
Программировани  
е. Математические  
основы, средства,  
теория. 2001

## 9. Критика исчерпывающего тестирования

- Современные программы настолько велики, что исчерпывающее тестирование невозможно за сколь-нибудь разумное время

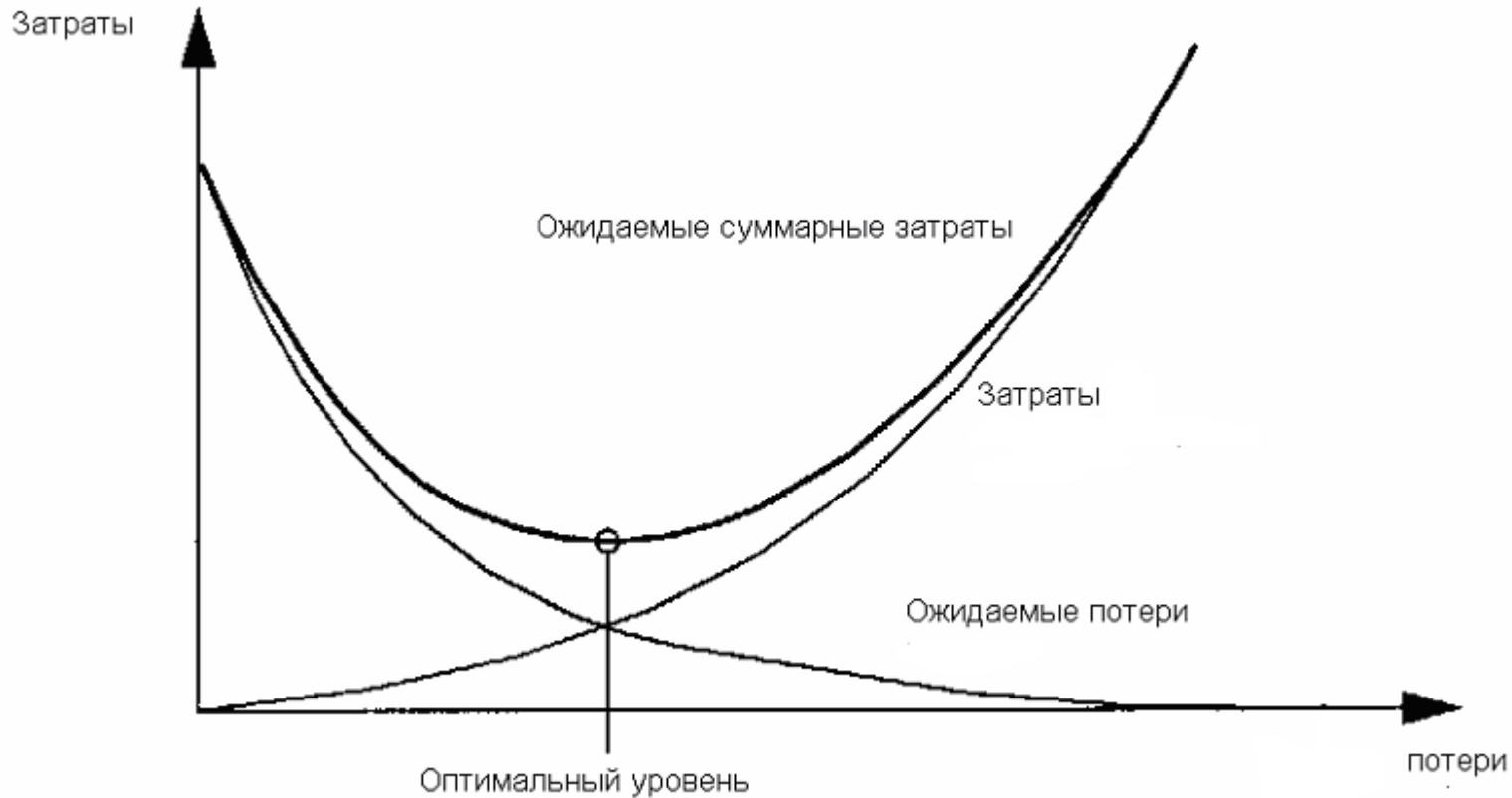


Маргарет Хамильтон, главный инженер НАСА, и код миссии Аполлон

# 10. Управление качеством как риском

- Для крупного банка и для opensource разработчика низкое качество продукта несет разные риски
  - Вы можете написать на коленке программу, но не продукт
- Само наличие контроля качества существенно увеличивает шанс проекта на успех (Capers Jones. Estimating Software Costs).
- Разные подходы к контролю качества
  - Если вы делаете ядерный реактор
  - Если вы делаете Gmail (концепция “достаточно качественного” ПО)
  - Если вы делаете что-то для community

# 11. Важность оценки качества



Затраты на тестирование и потери от низкого качества продукта

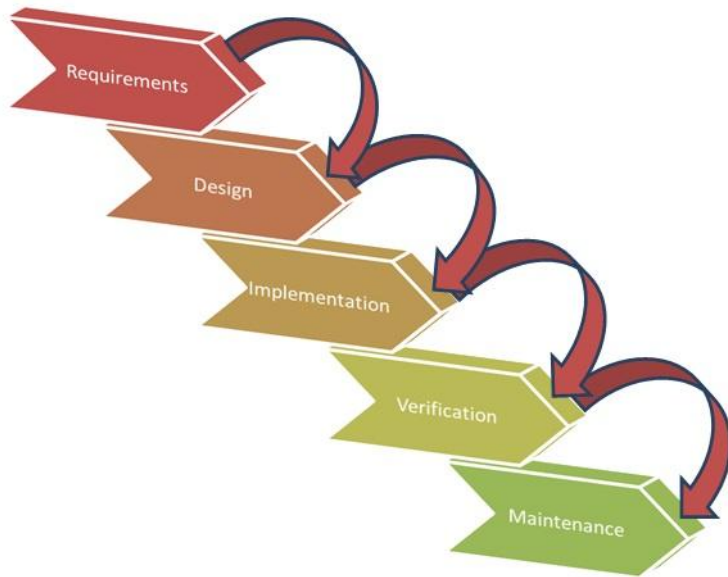
## 12. Шесть правил процесса тестирования

- Качество процесса тестирования определяет успех тестирования
- Тестирование должно внедряться на самых ранних этапах, например, при анализе требований
- Инструментальные средства тестирования должны быть использованы как можно раньше
- Необходимы люди, занимающиеся улучшением процесса тестирования
- Тестирование – область для профессионалов
- Необходимо создавать культуру “созидательного разрушения”

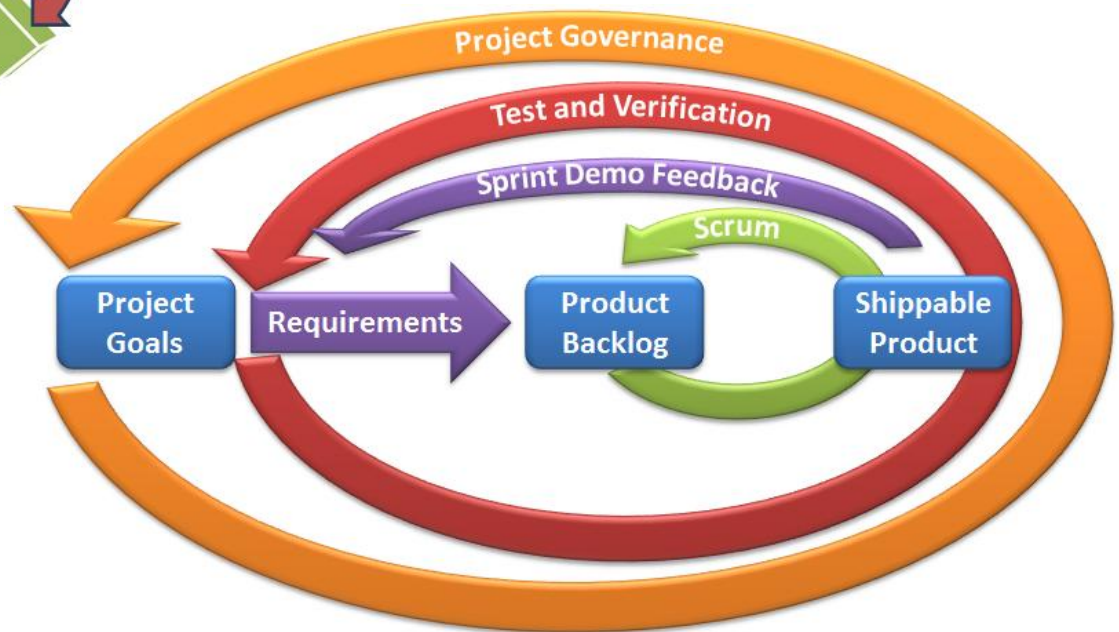
## 13. Процесс тестирования

- Что тестировать?
- Как тестировать?
- Когда тестировать?
- Что делать с результатами тестирования?

# 14. Тестирование и процесс



Где именно мы должны иметь тестирование на этих картинках?



# 15. Когда начинать тестирование

## Стоимость исправления ошибок





## 16. Инструменты

- Написать автоматический тест – сложно, исполнить – легко
- Чем больше автоматических тестов, тем быстрее можно узнать об отказе продукта
  - Чем раньше баг известен, тем проще его поправить

## 17. Управление изменениями

- Управление новой функциональностью
- Управление ошибками
  - Необходим четко проработанный процесс обработки ошибок
    - тяжесть отказа и приоритет
    - ILW метрика (impact-workaround-likelihood)

## 18. Анализ качества после выпуска

- Даже если проект уже закончен, может быть нужно анализировать его качество
  - Для исправлений новой версии
  - Для улучшения качества работы SQE

# 19. Q&A