

# Manual QA

Lecture 1. Качество программного обеспечения.

Дорофеев Максим

# Дорофеев Максим

- Около 4-х лет в тестировании ПО
- Руководжу командой тестировщиков, состоящей из 9 человек
- Archer Software, Agilie, SoftServe

# Расскажите о себе

- Какое хобби?
- С чем ассоциируется тестирование?
- Какие качества могут помочь стать хорошим QA

# Профессия тестировщика: Портрет тестировщика

## Коммуникабельность

Умение  
концентрироваться

Усидчивость

Изобретательность

Логическое  
мышление



Внимательность

Небезразличие к  
судьбе продукта

Умение поставить  
себя на место  
пользователя

# Профессия тестировщика: Цели и задачи тестирования ПО

Ожидаемый результат



Фактический результат



# Профессия тестировщика: 4 пути развития вашей карьеры





# Профессия тестировщика: Так с чего начать?

Google

тестирование



Все   Картинки   Видео   Новости   Карты   Ещё   Настройки   Инструменты

Результатов: примерно 21 400 000 (0,63 сек.)

Google

testing



Все   Картинки   Карты   Новости   Видео   Ещё   Настройки   Инструменты

Результатов: примерно 706 000 000 (0,43 сек.)

## Преимущество занятий на курсах:

- Проработанная система обучения, прошедшая испытания временем и приносящая результаты;
- Практикующие преподаватели;
- Поддержка во время и после обучения;
- Налаженная коммуникация школы с HR-специалистами - помощь в трудоустройстве.

# Профессия тестировщика: Мифы о тестировании

Миф первый:

## Социальный портрет QA Engineer:

- образование:  
гуманитарное

vs

техническое

- возраст:  
можно  
начинать,  
если мне 30+?

Миф второй:

**Работать QA Engineer - это просто:**

давить кнопки и писать документацию

Миф третий:

**Работа QA Engineer'a - это рутинная работа**



Миф четвертый:

Попасть в тестирование просто

# Миф пятый:

На курсах меня научат всему,  
я могу ничего не делать!

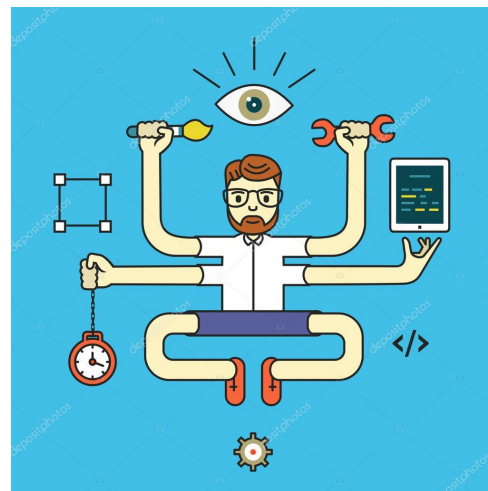
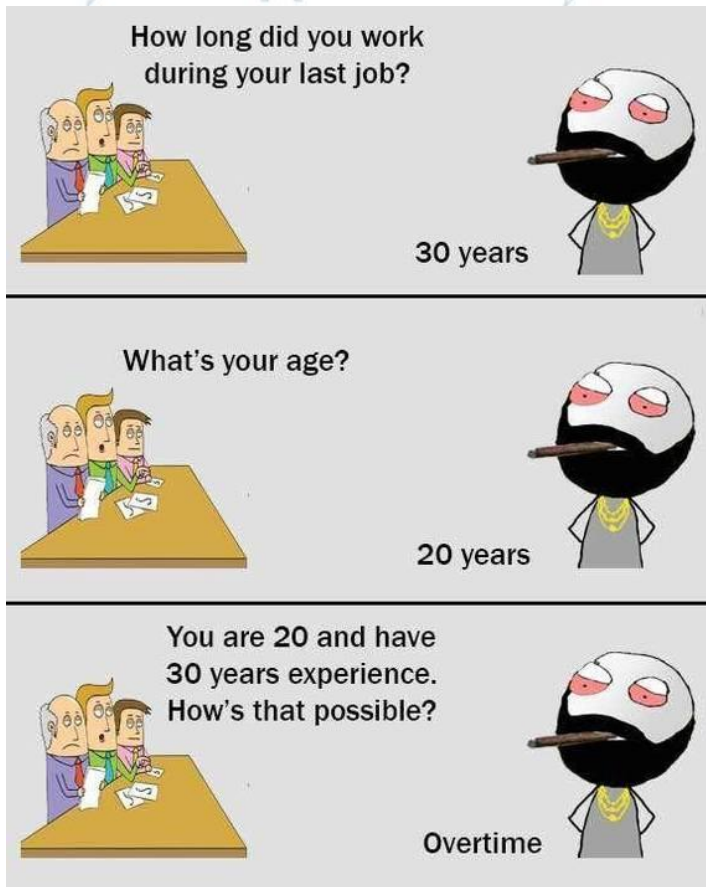
# Формат обучения для лучшего освоения материала и его запоминания:

- **Всегда говорить, если что-то непонятно**
- **Задавать вопросы**
- **Обсуждать на занятии и за пределами**
- **Учиться заниматься самостоятельно**
- **Быть готовым к срезу знаний**

# Преимущества и сложности работы в IT

# Сложности

OVERTIME



САМОРАЗВИТИЕ

# Преимущества



- **Комфортное и современное место работы**
- **Интересные задачи и постоянное развитие**
- **Возможность командировок за границу**
- **Релокация**
- **Компенсация английского**
- **Компенсация спортивного зала**
- **Страховка**
- **Оплачиваемый отпуск/больничный**
- **Кофе, печенье и тд.**

# Качество ПО. ISO 8402.

Совокупность характеристик программного обеспечения, относящихся к его способности удовлетворять **установленные** и **предполагаемые** потребности.

# Профессиональный подход к качеству.

- Соответствие требованиям;
- Пригодность к использованию.

## Цели тестирования.

- Собрать актуальную информацию о соответствии производимого продукта требованиям;
- Предоставить данную информацию всем заинтересованным лицам.

# Quality Control - Testing.

Самый низкий уровень - прохождение тест - кейсов и локализация дефектов.

# Quality Control - QC.

Совокупность действий, проводимых над продуктом в процессе разработки, для получения информации о его **актуальном состоянии** в разрезах: "готовность продукта к выпуску", "соответствие зафиксированным требованиям", "соответствие заявленному уровню качества продукта".

# Quality Assurance - QA.

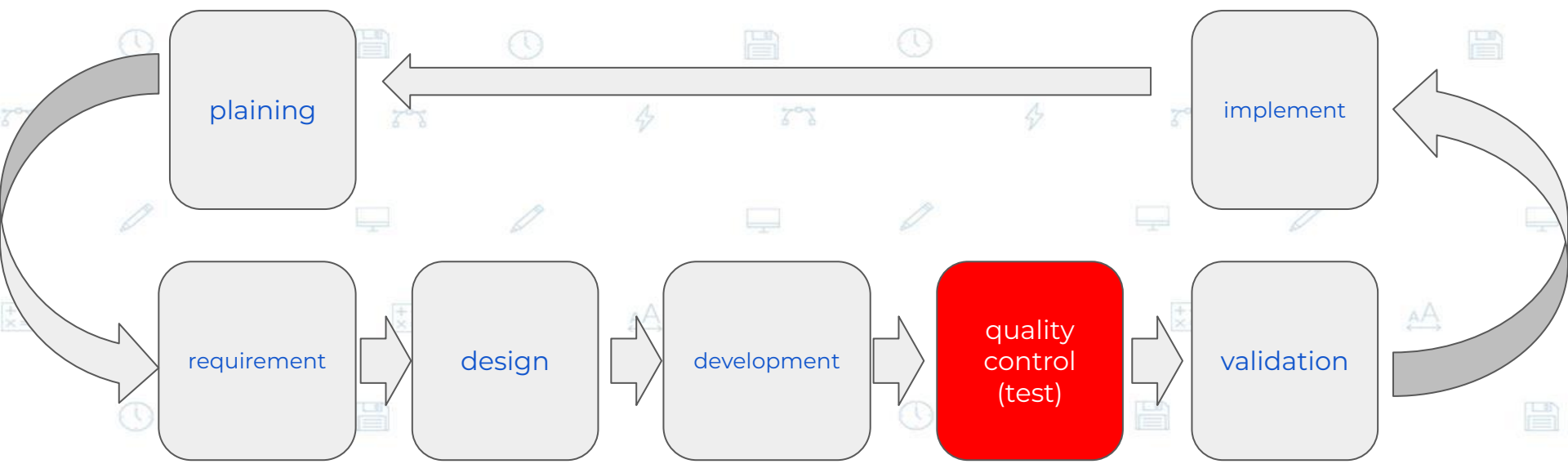
Совокупность мероприятий, охватывающих **все технологические этапы разработки**, выпуска и эксплуатации программного обеспечения (ПО) информационных систем, предпринимаемых на разных стадиях жизненного цикла ПО, для обеспечения требуемого уровня качества выпускаемого продукта.

# Как проходит процесс тестирования ПО.

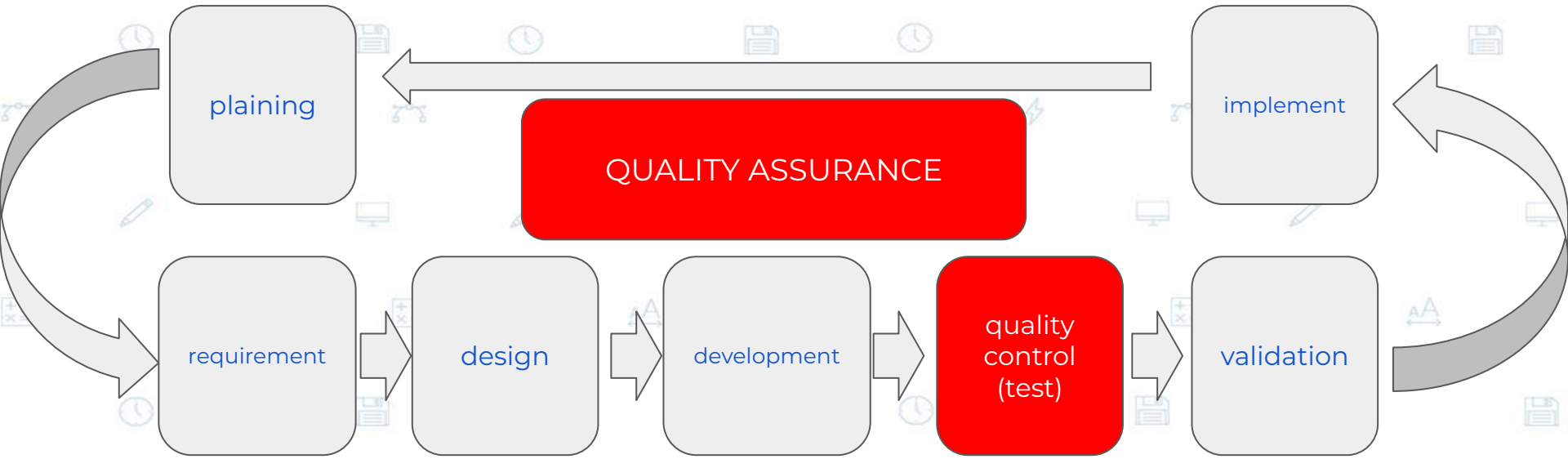
- Тестировщик получает программу;
- Тестировщик Наблюдает за работой программы, в определенных, искусственно созданных условиях;
- На выходе получает информацию о соответствиях и несоответствиях;
- Далее эта информация используется для того, чтобы улучшить уже существующий продукт. Либо для того чтобы изменить требования к разрабатываемому продукту.



# Development life cycle



# QA vs QC



# Software testing.

Одна из техник контроля качества, включающая в себя:  
Активности по планированию работ (**Test Management**),  
проектированию тестов (**Test Design**), выполнению  
тестирования (**Test Execution**) и анализу полученных  
результатов (**Test Analysis**).

# Basics. verification vs validation.

**Верификация** (verus - “верный”) -> правильность;

Отвечает на вопрос “**правильно ли мы это делаем?**”

**Валидация** (validus - “здравый”) -> польза, ценность.

Отвечает на вопрос “**правильную ли работу мы делаем?**”

# Характеристики качества ПО. Функциональность.

Определяется способностью ПО решать задачи, которые соответствуют зафиксированным и предполагаемым потребностям пользователя, при **заданных условиях** использования ПО.

# Характеристики качества ПО.

## Надежность.

Способность ПО выполнять требуемые задачи в обозначенных условиях на протяжении заданного **промежутка времени** или указанное **количество операций**.

# Характеристики качества ПО. Удобство использования.

Возможность легкого понимания, изучения,  
использования и привлекательности ПО для  
**пользователя.**

# Характеристики качества ПО. Эффективность.

Способность ПО обеспечивать требуемый **уровень производительности** в соответствии с выделенными ресурсами, временем и другими обозначенными условиями.



# Характеристики качества ПО. Удобство сопровождения.

Легкость, с которой ПО может анализироваться, тестироваться, изменяться для **исправления дефектов**, для реализации новых требований, для облегчения дальнейшего обслуживания и адаптироваться к имеющемуся окружению.

# Характеристики качества ПО. Портативность.

Характеризует ПО с точки зрения **легкости его переноса** из одного окружения (software/hardware) в другое.

# Характеристики качества ПО. Модель качества программного обеспечения.

КАЧЕСТВО

## Функциональность:

- функциональная исправность;
- соответствие стандартам;
- функциональная совместимость;
- безопасность;
- точность

## Надежность:

- завершенность;
- восстанавливаемость;
- устойчивость к отказам

## Удобство

### использования:

- удобство изучения;
- понятность;
- удобство и простота использования

## Эффективность:

- эффективность по времени;
- эффективность использования ресурсов

## Удобство

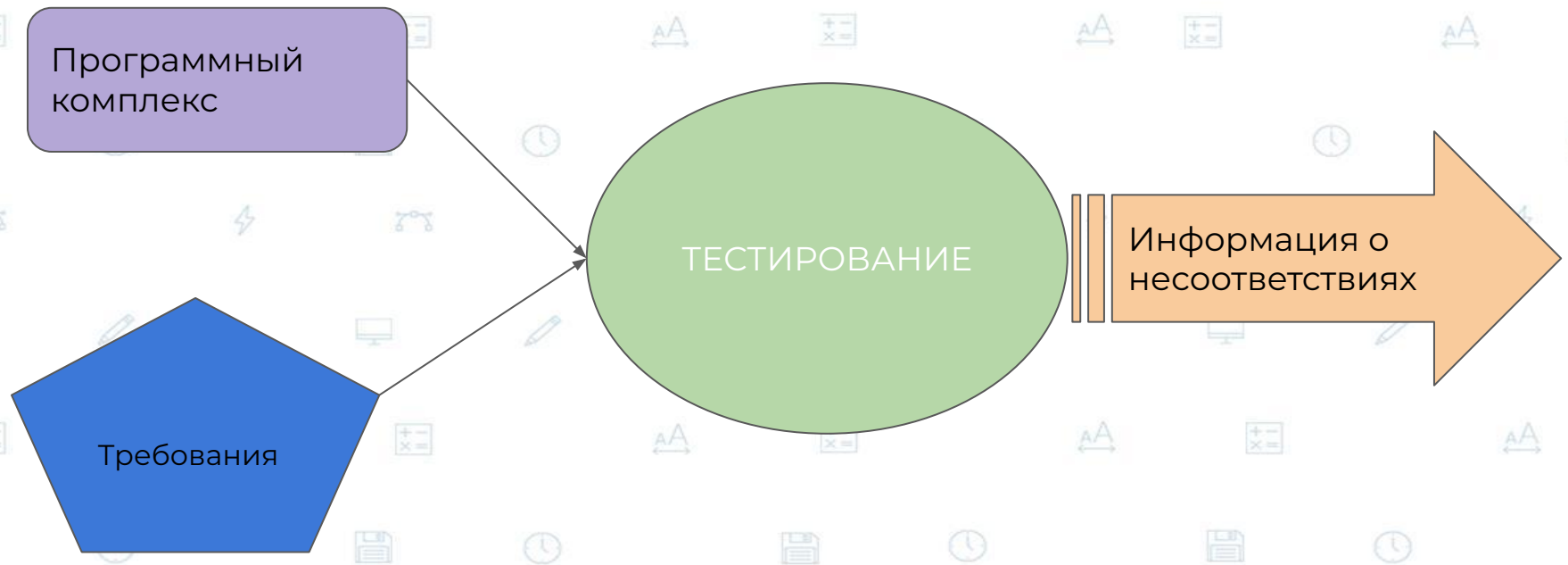
### сопровождения:

- стабильность;
- анализируемость;
- контролепригодность;
- изменяемость

## Портативность:

- удобство установки;
- заменяемость;
- совместимость

# Процесс тестирования.



# Принципы или законы тестирования

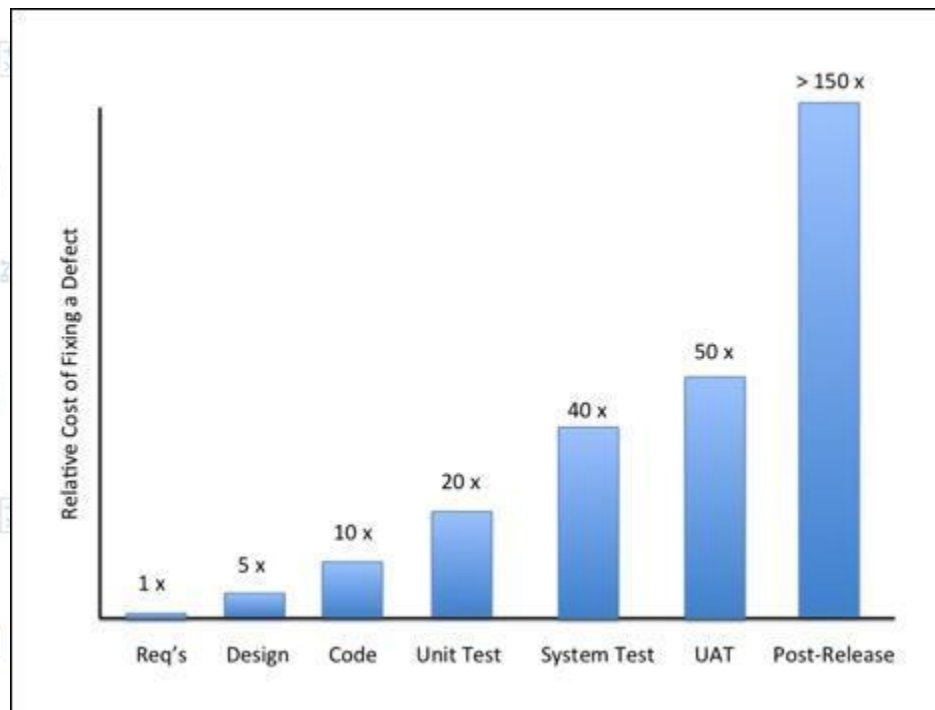
## I. Testing is context dependent

Тестирование контекстно зависимо. Различные продукты тестируют по-разному. Тестирование ракеты отличается от тестирования лопаты

## II. Early testing

Тестирование должно начинаться как можно раньше в процессе разработки системы и должно быть сфокусировано на определенных задачах

# Стоимость ошибки:





### III. Exhaustive testing is impossible

Тестирование всех комбинаций входных данных и предусловий не является возможным. Вместо исчерпывающего тестирования используются риски и приоритеты для фокусировки тестирования

Сколько нужно тестировать?

Чтобы ПОЛНОСТЬЮ протестировать поле, которое может принимать только 1 цифру необходимо:

10 (цифры 0-9) позитивных сценариев + 26 (uppercase alphabetical) + 26 (lowercase alphabetical) + ~ 6 ( символы пунктуации) = как минимум 68 тестов



## IV. Defect clustering

Большой процент багов из общего количества багов содержится в одних и тех же модулях. Кластеры дефектов меняются со временем



## На заметку: Принцип Парето 20/80

Из общего количества возможных тестов всегда необходимо выбирать 20% таких, которые найдут 80% всех багов

## V. Pesticide paradox

Когда одни и те же тесты повторяются снова и снова, на каком-то этапе они перестанут находить новые баги. Для предотвращения этого явления, тестовые сценарии нужно регулярно ревьюить и менять, писать новые тесты

## VI. Testing shows presence of defects

Тестирование может показать, что ПО содержит дефекты, но не может доказать, что дефектов нет. Тестирование может уменьшить вероятность нахождения в ПО не найденных дефектов, но даже если дефекты не были обнаружены, это все равно не доказательство правильно работы ПО

## VII. Absence of errors fallacy

Нахождение и исправление дефектов не поможет, если ПО не удовлетворяет потребности и нужды пользователя



# Процесс тестирования.



качество

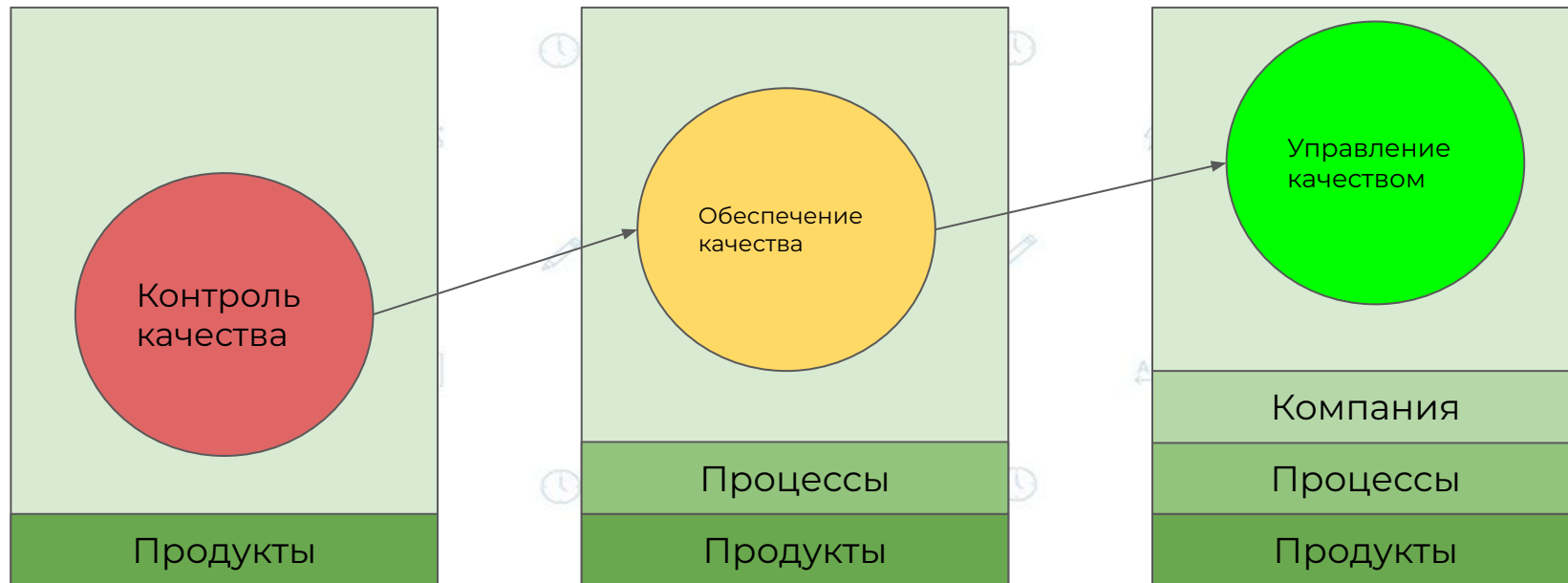


надежность



количество ошибок

# Эволюция повышения качества ПО.





# КНИГИ



# Контакты

**телефон:** 093-908-05-15

**скайп:** maxim.dorofeev3

**почта:** dorofeev.maxim90@gmail.com

# Вопросы и ответы





# Ссылки.

[Процесс разработки программного обеспечения ВИКИ](#)

<http://habrahabr.ru/post/111674/>

<http://alexeybulat.blogspot.com/2008/01/blog-post.html>

<http://alexeybulat.blogspot.com/2007/12/qa-qc-testing.html>

<http://www.protesting.ru/qa/quality.html>

<https://www.guru99.com/software-testing-life-cycle.html>

<http://www.protesting.ru/testing/>

<https://kurspc.com.ua/node/103>