

---

# **Инженеры SQA и тестирования**

описание, классификация,  
признаки

---

---

# Немного о себе

- Окончил СПбГТУ «Политех» по специальности «Робототехника»
  - В IT-индустрии с 1993 г., с 2000 специализируюсь в области тестирования и SQA
  - Кроме тестирования и QA, имею опыт разработчика, специалиста поддержки, сетевого инженера, IT-преподавателя
  - Работал в российских и иностранных компаниях, в т.ч. за рубежом; размер компаний – от стартапов до транснациональных IT-гигантов (Motorola)
  - Предметные области: системы управления корабельным вооружением, судовые системы динамического позиционирования, телекоммуникации, мобильные приложения, видео по запросу, fin-tech
  - В настоящий момент работаю QA Tech Lead в fin-tech компании, со специализацией в области автоматизации тестирования
  - С 2006 года веду курс по основам тестирования в Высшей Инженерной Школе (подразделение СПбПУ им. Петра Великого)
-

---

# Предмет разговора

- Каковы основные характеристики и навыки специалистов в области контроля качества ПО?
  - Какие разновидности существуют? Как их классифицировать?
  - Какие требования к разного рода специалистам по качеству ПО могут предъявлять работодатели?
  - Какие ключевые слова могут встретиться в заявках работодателей и резюме соискателей, и что они означают?
-

---

Чем занимаются специалисты SQA?

**S**oftware **Q**uality **A**ssurance

Обеспечивают качество программного  
продукта

---

# Что является основой обеспечения качества ПО?

- Конечная инстанция – пользователь (end user, customer). Он голосует рублём (долларом, евро, юанем) и/или количеством установок
- Основной критерий качества программного продукта – он должен понравиться пользователям («User should be happy»)
- Все практики контроля качества ПО преследуют одну цель: продукт должен понравиться пользователю
- Одна из частных целей – минимизация количества ошибок в ПО, попадающих к пользователю
- Средство достижения этой цели – «функциональное тестирование»:
  - все пункты спецификации требований реализованы в продукте
  - поведение продукта не противоречит этим требованиям и здравому смыслу (последнее – особенно важно!)

---

# Продукция фирмы Apple

Пример ориентации разработки IT-продуктов на конечного пользователя:

- iPod
  - iPhone/iPad + iOS
  - MacBook + MacOS
-

---

# Как обеспечить качество?

- «Безошибочное проектирование», «пассивные методы» - Quality Assurance (QA): установка, поддержание и совершенствование оптимальных производственных процессов создания программных продуктов
  - «Выявление и устранение ошибок», «активные методы» - Quality Control (QC): оценка качества конечного продукта - тестирование и эксплуатация
-

# Два класса специалистов, обеспечивающих качество

- **Quality Assurance** – «процессные инженеры»
  - Контроль производственных процессов: формирование, описание, адаптация под меняющиеся реалии бизнеса и технологий, согласование с менеджментом, внедрение, помощь персоналу
  - Контроль документации: создание шаблонов, правил, надзор за следованием этим шаблонам и правилам
  - Подготовка и проведение сертификаций (CMM/CMMI, ISO-9000)
- **Quality Control** – собственно «тестировщики», «тестеры»
  - Выполнение программ с целью поиска ошибок
  - Любая деятельность, направленная на обнаружение ошибок в программном обеспечении
  - Техническое исследование с целью получения информации, связанной с качеством тестируемого продукта

---

# Кто востребованнее, кого чаще ищут?

- В правильно построенной софтверной компании нужны и те, и другие
  - Тестировщиков (QC) нужно больше, чем процессных инженеров (QA)
  - На практике чаще всего ищут всё-таки тестировщиков
  - Имейте в виду, что обычно «QA» зовут именно тестировщиков, хотя это и не совсем верно с «академической» точки зрения
-

---

## Набор навыков, обязательный для всех тестировщиков (по крайней мере с уровня Middle)

- Общая компьютерная грамотность
  - Знание теории тестирования и владение профессиональной терминологией
  - Умение анализировать спецификации требований, пользовательские сценарии, приёмочные критерии для создания соответствующих наборов тестов (тестовых спецификаций).
  - Владение необходимым набором техник и приёмов
  - Умение документировать и сопровождать найденные ошибки, проверять их исправление («верифицировать»)
  - Владение необходимым набором профессиональных инструментов как минимум в качестве пользователя
  - По крайней мере общие представления о предметной области, для которой разрабатывается тестируемый продукт
-

---

# Необходимый набор профессиональных инструментов тестировщика

- Баг-трекеры (системы сопровождения найденных ошибок): Jira, RedMine, Mantis, Bugzilla, ClearQuest
  - Системы хранения и выполнения тестов: TestLink, TestRail, PractiTest, SpiraTest , qTest, Xstudio
  - Средства коллективной работы: Confluence, GoogleDocs
  - Системы контроля версий : ClearCase, CVS, MS SourceSafe, Subversion, Git, Mercurial, BitBucket, Tortoise(SVN/CVS/Hg)
  - Системы непрерывной интеграции (CI): Jenkins, TeamCity
-

---

# Нетехнические навыки тестировщика

- Коммуникационные навыки и «soft skills»
  - Высокий уровень грамотности и владения языками (русским, английским)
-

---

# Технические специализации

---

по типам выполняемого тестирования

# Функциональное тестирование

- Выполнение тестов вручную согласно тестовой спецификации (тест-плану)
- Спецификация или тест-план составляется на основании:
  - функциональных требований
  - use cases («вариантов использования»)
  - приёмочных критериев (acceptance criteria of user story, терминология SCRUM/Agile)
- Частью функционального тестирования является сопровождение ошибок
  - документирование обнаруженных отклонений с помощью багтрекеров
  - проверка после исправления программистами (верификация)

---

# Автоматизированное тестирование

- Выполнение программы осуществляется не инженером, а другой, специализированной программой (скриптом), написанной, в свою очередь, инженером-«автоматизатором»
  - SDET: Software DEveloper in Test
  - Unit Testing: Nunit/xUnit, TestNG, Watir
  - Graphical User Interface: Selenium WebDriver, QTP, Segue, IBM Rational Robot
  - Application Programming Interface: RestSharp, SoapUI, Swagger, Postman, Curl, CTESK, dotTEST
  - Behavior Driving Development: Cucumber/Gherkin, SpecFlow
-

---

# Нагрузочное тестирование

- Performance, Load, Stress testing
  - Как правило, речь идёт об имитации нагрузки на web-серверы.
  - Самый популярный инструмент – JMeter
  - Разновидность автоматизированного тестирования
-

---

# Тестирование безопасности

- Аутентификация и авторизация
  - Несанкционированный доступ к информационным системам
  - Целостность информационных систем
  - Очень часто речь идёт о Web-сервисах
-

---

# Технические специализации по типам приложений

Вне зависимости от типа приложения, обычно  
требуется выполнять ручное функциональное  
тестирование

---

---

# Web

- HTML, HTML5, CSS, REST, SOAP, JSON, JavaScript, Ajax, WebSphere
  - Apache, IIS
  - TomCat, JSP
  - СУБД (DBMS):
    - Реляционные: Oracle, MS SQL Server, PostgreSQL
    - Нереляционные: MongoDB: noSQL, document-oriented DBMS.
  - Системы развёртывания: Docker, Octopus
-

---

# Мобильные

- Android
  - iOS
  - реже - Windows Phone
  - Продвинутый уровень:
    - Умение снимать внутренние логи
    - ADB
    - SQLite
    - Robotium
    - Appium
-

---

# Настольные (desktop)

- Windows
  - Linux (реже – Unix): Ubuntu, RedHat, CentOS
  - MacOS
  - Приложения на этих платформах
-

---

# Телекоммуникации

- ЛВС
    - Ethernet
    - WiFi
    - маршрутизация
  - WAN
    - GigabitEthernet
    - ATM
    - PON
    - PPOE, PPTP, L2TP
  - Сотовая связь
    - GSM
    - GPRS
    - LTE, 4/5G
    - iDEN
  - Bluetooth (LE)
  - NFC
-

---

# Встраиваемые (embedded)

- Оборудование:
    - Банкоматы
    - БС сотовых сетей
    - Бытовая техника
  - Операционные системы:
    - WxWorks
    - RTOS
    - QNX
  - Internet of Things
-

---

# Классификация по стажу

Релевантные навыки

---

---

# Trainee

- от 0: как правило, фирмы берут и обучают для себя «на вырост»
  - общее владение компьютером и желание работать в IT
-

---

# Junior

- от полугода до 2 лет
  - «продвинутый» пользователь, желание работать в IT
  - знание теоретических вопросов (что такое «баг», «верификация», «релиз», «процесс»)
  - какой-нибудь опыт работы с:
    - багтрекером
    - версионным контролем
    - системой управления тестами
-

# Middle

- От 2 до 7 лет
- Хорошее знание теории, чтение профильной литературы, возможно посещение курсов и тренингов
- Профессиональный пользователь инструментов
  - багтрекеры
  - версионный контроль
  - управление тестами
  - управление требованиями
- Пользователь по крайней мере реляционных СУБД:
  - Oracle
  - MS SQL Server
  - PostgreSQL
  - SQL Server Management Studio
  - Toad
- Глубокая специализация на каком-то (или 2-3) видах приложений; желательно знакомство со всем (почти) спектром
- Работа в разных компаниях/проектах, по разным типам процессов (SCRUM, Waterfall, RUP, CMM/CMMI, ...)

---

# Senior

- От 7 лет
  - Блестящее знание теории и нюансов. Участие в профильных мероприятиях, возможно преподавание
  - Опыт работы в разных компаниях/проектах
  - Опыт руководства хотя бы 1 человеком, хотя бы временно
  - Набор технических навыков (см. далее)
-

---

# Senior – технические навыки

- Владение на уровне администратора хотя бы одним инструментом каждого типа
    - баг-трекеры
    - версионный контроль
    - непрерывная интеграция
    - управление тестами
    - управление требованиями
    - средства коллективной работы
  - Опыт работы с различными типами приложений, операционных систем (Windows + Linux must have!)
-

---

# Senior – технические навыки

- Работа с СУБД:
    - реляционными (MS SQL, Oracle, PostgreSQL, Maria)
    - других типов (MongoDB + Robomongo)
  - Хотя бы поверхностное знакомство с сетевыми технологиями:
    - LAN\WAN\Mobile
    - Адресация, маршрутизация, уровни (OSI), стеки
  - Очень желательно (практически обязательно):
    - владение хотя бы основами программирования
    - опыт нагрузочного и/или автоматизированного тестирования
-

---

# Guru

- не менее 15 лет
  - реально знает о тестировании ВСЁ!
  - ... и все умеет...
-

---

# Административная классификация

Обязанности

---

---

# Engineer

Выполняет поставленные руководством задачи в рамках должностной инструкции:

- выполняет приложение с целью поиска ошибок
  - собирает техническую информацию о качестве приложения
  - поставленные задачи могут носить исследовательский характер: определить пригодность той или иной технологии, практики, инструмента.
-

---

# QA Tech Lead

- Руководит группой тестировщиков на проекте; ставит задачи и контролирует их исполнение
  - Непосредственно участвует в тестировании (как инженер)
  - Помогает тестировщикам в техническом плане: применение техник и инструментов, решение технических затруднений
  - Собирает и предоставляет начальству (QA Manager) метрики группы
-

---

# QA Manager

- Определяет техническую политику в подведомственных подразделениях
  - Ставит и контролирует задачи группам тестировщиков на различных проектах в подведомственных подразделениях
  - Собирает и анализирует метрики разных проектов/групп
  - Взаимодействует с процессными инженерами в части поддержания и улучшения процессов (практик) тестирования
-

---

# Guru, evangelist

- Предлагает, пропагандирует новые подходы и идеи не только среди технического персонала, но и среди менеджмента
  - Помогает в решение сложных технических задач, часто – с помощью нетривиальных новаторских подходов
  - Служит локальным центром технической экспертизы
-

# Обязательно ли тестировщику уметь программировать?

- В реальности большая часть тестировщиков программировать не умеет (по крайней мере профессионально)
- Обязательно:
  - Для соискателей на должности инженеров по нагрузочному и автоматизированному тестированию
  - Для соискателей на позицию уровня Senior или QA Manager (хотя бы базовые навыки, хотя бы на одном ЯП)

---

# Ориентировочные цифры по зарплатам

- Инженер: от 30 т.р. (junior) до 90 (Senior)
  - Инженер по автоматизированному тестированию: 70 – 150 т.р.
  - Tech Lead: 60 -170 т.р.
  - Менеджер: 80 – 200 (и более) т.р.
-