
Test Task 4 QA

Roman Alekseev 13 May 2020





Тестовое задание для интервью кандидатов на позицию QA Specialist

Теория:

1. Основные принципы Agile-тестирования. На какой стадии разработки программного продукта необходимо обеспечивать качество? Чем QA специалист отличается от тестировщика?

❖ Принципы Agile-Testing:

- **Тестирование продвигает проект вперед.** Непрерывное тестирование – единственный способ обеспечить непрерывный прогресс. Agile Testing обеспечивает обратную связь на постоянной основе, и конечный продукт отвечает требованиям бизнеса.
- **Тестирование не является этапом** – Agile команда проводит тестирование вместе с командой разработчиков, чтобы убедиться, что функции, реализованные во время данной итерации, действительно выполнены. Тестирование не ведется для более поздней фазы.
- **Все тестируют.** В гибком тестировании приложение тестирует вся команда, включая аналитиков, разработчиков и тестировщиков. После каждой итерации даже заказчик проводит приемочное тестирование.
- **Сокращение циклов обратной связи** – В Agile Testing бизнес-команда узнает о разработке продукта для каждой итерации. Они участвуют в каждой итерации. Непрерывная обратная связь сокращает время отклика обратной связи, и, следовательно, затраты на его устранение уменьшаются.

-
- **Содержите код в чистоте** . Дефекты исправляются по мере их появления в одной и той же итерации. Это гарантирует чистый код на любом этапе разработки.
 - **Облегченная документация** – вместо исчерпывающей документации по тестированию Agile Testers –
 - Используйте многоразовые контрольные списки, чтобы предложить тесты
 - Сосредоточьтесь на сути теста, а не на случайных деталях.
 - Используйте легкие стили документации / инструменты.
 - Запишите идеи испытаний в чартеры для пробных испытаний.
 - Используйте документы для нескольких целей.
 - **Использование одного тестового артефакта для ручных и автоматических тестов.** Один и тот же артефакт тестового скрипта можно использовать для ручного тестирования и в качестве входных данных для автоматических тестов. Это устраняет необходимость в документации по тестированию вручную, а затем эквивалентный сценарий теста автоматизации.
 - **«Готово Готово», а не просто выполнено** – говорят, что в Agile функция выполняется не после разработки, а после разработки и тестирования.
 - **Test-Last против Test Driven** – Тестовые случаи пишутся вместе с требованиями.
 - ❖ **Agile** – это методология итеративной разработки, в которой действия по разработке и тестированию выполняются одновременно. Тестирование не является отдельной фазой и тесно связано со всеми фазами разработки, такими как требования, проектирование, кодирование и генерация тестовых наборов. Тестирование происходит одновременно через жизненный цикл разработки. Кодирование и тестирование выполняются в интерактивном режиме и постепенно, что приводит

к качественному конечному продукту, который соответствует требованиям заказчика.

❖ Международный стандарт системы менеджмента качества ISO 9000:2015

Пункт 3.2 раскрывает два определения:

- Обеспечение качества (3.2.10) — часть управления качеством, направленная на обеспечение уверенности в том, что требования к качеству будут выполнены.
- Контроль качества (3.2.11) — часть управления качеством, ориентированная на выполнение требований к качеству.

Из этих определений следует, что мы либо обеспечиваем качественный продукт, либо проверяем продукт на соответствие качеству.

Из данных требований заключаем: 1) QA-specialist - тот кто помогает команде разработки выпускать (обеспечить) качественный продукт.

2) Тестировщик - это тот кто на последнем этапе разработки лишь проверяет качество продукта, но не влияет на процесс обеспечения качества и разработки, не видит полной картины процесса и поэтому не может взять на себя ответственность за качество выпускаемого продукта.

2. Что такое спецификация? Как QA специалист применяет спецификацию в своей работе?

Спецификация требований к программному обеспечению (SRS) – это документ, в котором содержится полное описание ожидаемых результатов работы системы.

Спецификацию получаем от заказчика проанализировав, исследовав его требования и переводя их на качественно новый, более детализированный уровень, на котором данными требованиями и будем руководствоваться. На основании спецификации QA специалист и создает необходимые тесты.

Спецификация — это текстовый файл с описанием того, что нужно протестировать в тестовых данных. В ней указывается какие результаты должна получить программа.

Тестовый код находит реальные, вычисленные на живом коде результаты. А тестовый движок производит сверку спецификации и вычисленных результатов. При этом в тестовый лог записывается информация какие спецификации и свойства были протестированы, каков результат тестирования и сколько времени оно заняло.

3. Что такое тест-кейс? Позитивные негативные тест-кейсы. Приведите пример, когда негативные тест-кейсы проверяются в первую очередь. Сколько ожидаемых результатов может иметь один тест-кейс?

❖ **Test Case** — это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Атрибуты тест-кейса

- Уникальный идентификатор тест-кейса
- Название
- Предусловия
- Шаги (описание последовательности действий, которая должна привести нас к ожидаемому результату)
- Ожидаемый результат (что мы ожидаем увидеть после выполнения шагов.)

Виды Тестовых Сценариев:

Тест кейсы разделяются по ожидаемому результату на позитивные и негативные:

- Позитивный тест кейс использует только корректные данные и проверяет, что приложение правильно выполнило вызываемую функцию.

-
- Негативный тест кейс оперирует как корректными так и некорректными данными (минимум 1 некорректный параметр) и ставит целью проверку исключительных ситуаций.

- ❖ Например нужно проверить, что вызываемая приложением функция не выполняется при срабатывании валидатора.

- ❖ Каждый выполненный тест-кейс, дает нам один из трех результатов:

1. **Положительный результат**, если фактический результат равен ожидаемому результату,

2. **Отрицательный результат**, если фактический результат не равен ожидаемому результату. В этом случае, найдена ошибка.

3. **Выполнение теста блокировано**, если после одного из шагов продолжение теста невозможно. В этом случае так же, найдена ошибка.

Как правило одним тест-кейсом проверяется одна конкретная задача, и для этой задачи должен быть только один ожидаемый результат.

(Но существуют и тест-кейсы с несколькими результатами – когда результаты заданы для каждого шага выполнения проверки, а не только для последнего.)

4. Что такое тест-план? Особенности составления тест-плана при Agile-разработке.

- ❖ **Test Plan** - это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.

Отвечает на вопросы:

Что надо тестировать? Что будете тестировать? Как будете тестировать? Когда будете тестировать? Критерии начала тестирования? Критерии окончания тестирования?

❖ Особенности составления тест-плана при Agile-разработке.

Планирование тестирования должно начинаться во время планирования выпуска и обновляться во время каждого спринта. Планирование тестирования должно охватывать следующие задачи –

- Определение объема тестирования, степени тестирования, целей тестирования и спринта.
- Выбор среды тестирования, инструментов тестирования, данных тестирования и конфигураций.
- Назначение тестирования функций и характеристик.
- Планирование тестовых заданий и определение частоты тестов.
- Определение методов испытаний, методов, инструментов и данных испытаний.
- Определение предпосылок, таких как задачи предшественника, экспертиза и обучение.
- Определение зависимостей, таких как функции, код, системные компоненты, поставщик, технология, инструменты, действия, задачи, команды, типы тестов, уровни тестов и ограничения.
- Установление приоритетов с учетом важности и зависимости клиента / пользователя.
- Прибытие в срок и усилия, необходимые для тестирования.
- Определение задач на каждом спринте.

5. Что такое баг? Приоритет и серьезность бага: в чем отличие?
Приведите пример бага с наивысшим приоритетом и
наименьшей серьезностью. Что такое регрессионный баг?

❖ Баг – это несоответствие фактического результата выполнения программы ожидаемому результату. Дефекты обнаруживаются на этапе тестирования программного обеспечения (ПО), когда специалист проводит сравнение полученных результатов работы программы (компонента или дизайна) с ожидаемым результатом, описанным в спецификации требований.

❖ Severity vs Priority

Серьезность (Severity) — это атрибут, характеризующий влияние дефекта на работоспособность приложения.

Приоритет (Priority) — это атрибут, указывающий на очередность выполнения задачи или устранения дефекта. Это инструмент менеджера по планированию работ. Чем выше приоритет, тем быстрее нужно исправить дефект.

Severity выставляется QA специалистом Priority – менеджером, тимлидом или заказчиком

Градация Серьезности дефекта (Severity)

S1 Блокирующая (Blocker)

Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна. Решение проблемы необходимо для дальнейшего функционирования системы.

S2 Критическая (Critical)

Критическая ошибка, неправильно работающая ключевая бизнес логика, дыра в системе безопасности, проблема, приведшая к временному падению сервера или приводящая в нерабочее состояние некоторую часть системы, без возможности решения проблемы, используя другие входные точки. Решение проблемы

необходимо для дальнейшей работы с ключевыми функциями тестируемой системой.

S3 Значительная (Major)

Значительная ошибка, часть основной бизнес логики работает некорректно. Ошибка не критична или есть возможность для работы с тестируемой функцией, используя другие входные точки.

S4 Незначительная (Minor)

Незначительная ошибка, не нарушающая бизнес логику тестируемой части приложения, очевидная проблема пользовательского интерфейса.

S5 Тривиальная (Trivial)

Тривиальная ошибка, не касающаяся бизнес логики приложения, плохо воспроизводимая проблема, малозаметная посредством пользовательского интерфейса, проблема сторонних библиотек или сервисов, проблема, не оказывающая никакого влияния на общее качество продукта.

Градация Приоритета дефекта (Priority)

P1 Высокий (High)

Ошибка должна быть исправлена как можно быстрее, т.к. ее наличие является критической для проекта.

P2 Средний (Medium)

Ошибка должна быть исправлена, ее наличие не является критичной, но требует обязательного решения.

P3 Низкий (Low)

Ошибка должна быть исправлена, ее наличие не является критичной, и не требует срочного решения.

-
- ❖ Допустим поехала верстка на главной странице приложения. Для клиента верстка важна. С точки серьезности такая ошибка будет оценена как Trivial. А приоритет будет High.

(В итоге поле серьезности/приоритета ошибки будет обладать значением Critical или High.)

- ❖ Regression bugs - это баги которые возникают после внесения изменений в программу (например добавляется новая функциональность): перестаёт работать то, что должно было продолжать работать в уже реализованном функционале. Поиском таких багов занимается регрессивное тестирование.

Регрессионное тестирование — это вид тестирования направленный на проверку изменений, сделанных в приложении или окружающей среде (починка дефекта, слияние кода, миграция на другую операционную систему, базу данных, веб сервер или сервер приложения), для подтверждения того факта, что существующая ранее функциональность работает как и прежде.

6. Автоматизированное тестирование. Преимущества и недостатки.

- ❖ Для автоматизации используются программные инструменты, имитирующие действия QA специалиста при ручной проверке. Такие инструменты используют в качестве входной информации тест-кейсы, записанные на формальном языке, операторы которого соответствуют действиям пользователя - например вводу команд, перемещению курсора, активизации пунктов меню и других интерфейсных элементов. Среди инструментов, предназначенных для автоматизированного функционального тестирования пользовательских интерфейсов, наиболее часто используются Selenium, RobotFramework, Watir, Zombie.js, PhantomJS. Это инструменты для создания наборов функциональных тестовых скриптов для проверки веб-приложений.

Автоматизация тестирования приложений основана на написании кода. Методика автоматизированного тестирования позволяет создать ожидаемый сценарий, а затем сравнить его с реальным и указать расхождения.

Автоматизация наиболее применима в сложных приложениях с большой функциональной частью. Особой популярностью пользуется автоматизация тестирования труднодоступных мест приложения, валидационных форм, базовых операций, часто используемой функциональности.

Плюсы автоматизированного тестирования

- **Нагрузка на приложение.** Когда используется автоматизированное тестирование, становится возможным моделирование большой нагрузки, которая приближена к реальной ситуации.
- **Временной фактор.** Ручное тестирование – это долгий и ресурсоемкий процесс, в то время как код для сценария пишется один раз.
- **Повторяемость.** Код автотестов может быть использован неоднократно, особенно при внедрении новой функциональности.

Минусы автоматизированного тестирования

- **Отсутствие обратной связи.** (Автоматизированное тестирование не способно предоставить обратную связь относительно качества продукта – оно лишь выполняет запрограммированные сценарии.)
- **Отсутствие тестирования глазами пользователя.** (Иногда в приложении остаются ошибки, которые могут быть не покрыты автотестами.)
- **Отсутствие возможности тестирования цвета, дизайна и эргономики.** (данный пункт не является первостепенным, но может значительно повлиять на качество продукта.)
- **Надежность.** (Автоматизированные тесты могут упасть по многим причинам, например, при большой загруженности тестовой машины или при проблемах с сетью.)
- **Стоимость.** (Для небольших проектов инструменты автоматизированного тестирования могут оказаться достаточно

затратными, поэтому более рационально их использовать для долгосрочных проектов.)

1. Опишите процесс тестирования канцелярской скрепки, с отсутствием на неё спецификации и ограничением времени на проверку не более 2 часов.

Прежде всего начнем с negative testing протестируем скрепку на Covid-19, на радиацию и токсичность.

Далее можем приступить к positive testing. На все у нас 2 часа. Оборудование готово (Бумага, Вибротест, Морозильная и Жаровая камеры) Итак:

Шаг 1 кто будет ее использовать? И зачем?

Учитель чтобы скрепить листы бумаги (Данный ответ даст общее представление о требующихся тестах)

Шаг 2 каковы варианты использования?

Скрепка используется для скрепления листов без повреждения бумаги

Шаг 3 каковы границы действия предмета?

Ограничение до 35 листов бумаги и до 65 листов с небольшими повреждениями

Границы распространяются и на окружающую среду. Скрепка работает при температурах до 65 градусов И на холоде до -65 градусов

Шаг 4 каковы условия отказа?

При скреплении более 65 листов скрепки повреждают бумагу

Также происходит при температуре окружающей среды выше 65 градусов и при понижении температуры более - 65 градусов

Шаг 5 как осуществить тестирование?

Пора автоматизировать процесс тестирования Использовать вибротест чтобы определить нормальное каждодневную эксплуатацию скрепки в течении 365 дней

2. Составьте Test Suite для функционального тестирования формы авторизации. Форма имеет два поля для ввода логина "Login" и пароля "Password", а также кнопку "Sign in" для подтверждения входа в систему (приложение или сайт).

Будем создавать Test-Suite при помощи программы TestLink

(<https://github.com/TestLinkOpenSourceTRMS/testlink-code>)

Создадим Test-Suite, назовём его Testing Module-A Далее Создадим кейсы TC001-4 TC002-4 TC003-4

Тест-кейс № TC001-4

Использование зарегистрированного значения «test» в поле «Password», соответствующее значению «test» в поле «Login»

Шаги:

Зайти на www.dev_post.com

Ввести в поле «Login» test

Ввести в поле «Password» test

Нажать на кнопку «Sign».

Ожидаемый результат:

Появляется сообщение «Ваш вход в систему подтвержден! Добро пожаловать!»

Тест-кейс TC002-4 Ноль на ноль

Шаги:

Зайти на www.dev_post.com

Ввести в поле «Login» 0

Ввести в поле «Password» 0

Нажать на кнопку «Sign».

Ожидаемый результат:

Появляется сообщение «Ваш вход в систему не подтвержден! Попробуйте снова»

Тест-кейс TC003-4 Пустые значения

Шаги:

Зайти на www.dev_post.com

Ничего не вводить в поле «Login»

Ничего не вводить в поле «Password»

Нажать на кнопку «Sign».

Ожидаемый результат:

Появляется сообщение «Ваш вход в систему не подтвержден! Поля «Login» и «Password» не заполнены Попробуйте снова»

3. Составьте позитивный и негативный Test Case для функциональной проверки формы обратной связи. Форма имеет обязательные поля “Name”, “Email”, “Message”, а также кнопку “Send”. После нажатия на кнопку выводится сообщение о результате операции.

- Тест-кейс No 1.

- Шаги:

Зайти на www.dev_post.com (логин – test, пароль – test).

Перейти на вкладку «Contacts».

Ввести в поле «Name» John

Ввести в поле «Email» emailtest@post.ru

Ввести в поле «Message» Lorem porto cavalli lucano

Нажать на кнопку «Send».

Ожидаемый результат: Появляется сообщение «Ваше сообщение успешно отправлено»

-
- Тест-кейс No 2.

Шаги:

Зайти на www.dev_post.com (логин – test, пароль – test).

Перейти на вкладку «Contacts».

Ввести в поле «Name» John

Ничего не вводить в поле «Email»

Ввести в поле «Message» Lorem porto cavalli lucano

Нажать на кнопку «Send».

Ожидаемый результат: Появляется сообщение «Ваше сообщение не отправлено
Поле «Email» не заполнено»

-
4. Какое минимальное количество тест-кейсов потребуется для проверки функционала “Родительский контроль” с ограничениями G, PG, R, NC-17, 18+, если в наличии 40 каналов, 15 из которых имеют возрастной рейтинг G, 10 – PG, 10 – R, 3 – NC-17 и 2 – 18+.

Классов эквивалентности тестируемых ограничений 5 поэтому минимальное количество тест-кейсов будет 5 Один для G, один для PG, один для R, один для NC-17, один для 18+ Итого 5

5. Кратко опишите процесс проведения Compatibility Testing сайта по следующим критериям: операционная система, разрешение экрана, браузер.

Тестирование совместимости (Compatibility Testing) — вид нефункционального тестирования, основной целью которого является проверка корректной работы продукта в определенном окружении. Окружение может включать в себя следующие элементы:

1. Аппаратная платформа 2. Операционная система (Unix, Windows, MacOS) 3. Браузеры (Internet

Explorer, Firefox, Opera, Chrome, Safari, Яндекс.Браузер) 4. Различное разрешение экрана.

Тестирование совместимости проводим в 2 этапа: 1) На первом этапе проверяется взаимодействие выпускаемого продукта с окружением, в которое он будет установлен, на различных аппаратных средствах. (Например запускаем сайт в браузерах компьютеров разных операционных систем, в браузерах планшетов и в мобильных устройствах) 2) На втором этапе выпускаемый продукт проверяется с позиции его конечного пользователя и конфигурации его рабочей станции.

6. Составьте чек-лист для проведения Smoke Testing плеера, используемого на с а й т е <https://www.youtube.com>.

Чек-лист YouTube плеера на мобильных устройствах

Чек-лист	Galaxy S5	Moto G4	Iphone X
Запуск	Ok	Ok	Ok
Пауза	Ok	Ok	Ok
Воспроизведение	Ok	Ok	Ok
Прокрутка вперед	Ok	Ok	Ok
Прокрутка назад	Ok	Ok	Ok
Во весь экран	Ok	Ok	Ok
Выход из полного экрана	Ok	Ok	Ok
Выключение звука	Ok	Ok	Ok
Включение звука	Ok	Ok	Ok
Выключение	Ok	Ok	Ok

7. Найдите и оформите баг-репорт, используя методику Exploratory Testing, на с а й т е <https://www.vmedia.ca>.

Bug #9196

Серьезность: Medium

Назначен на: <Руководитель проекта>

Выполнил QA specialist Roman Alekseev

Дата 13.05.2020

Описание: В футере сайта Спрайт LinkedIn при нажатии открывает недоступную страницу сайта компании на LinkedIn.com

Окружение:

Mac OS X 10.15.3 browser Chrome 81.0.4044 Screen resolution 1440x900
Current viewport 1440x747

Шаги:

1. Зайти на сайт
<https://www.vmedia.ca>
3. Найти внизу сайта иконку LinkedIn
4. Нажать на иконку LinkedIn
- 5 Перейти на страничку компании Vmedia на сайте LinkedIn.com

Ожидаемый результат:

Открывается страница сайта компании Vmedia на LinkedIn.com

Результат: Открывается недоступная страница <https://www.linkedin.com/company/unavailable/> с сообщением «Приносим извинения. Эта страница недоступна.»

Скриншоты прилагаю

<https://is.gd/lw9pwf>

<https://is.gd/RwMP5L>



