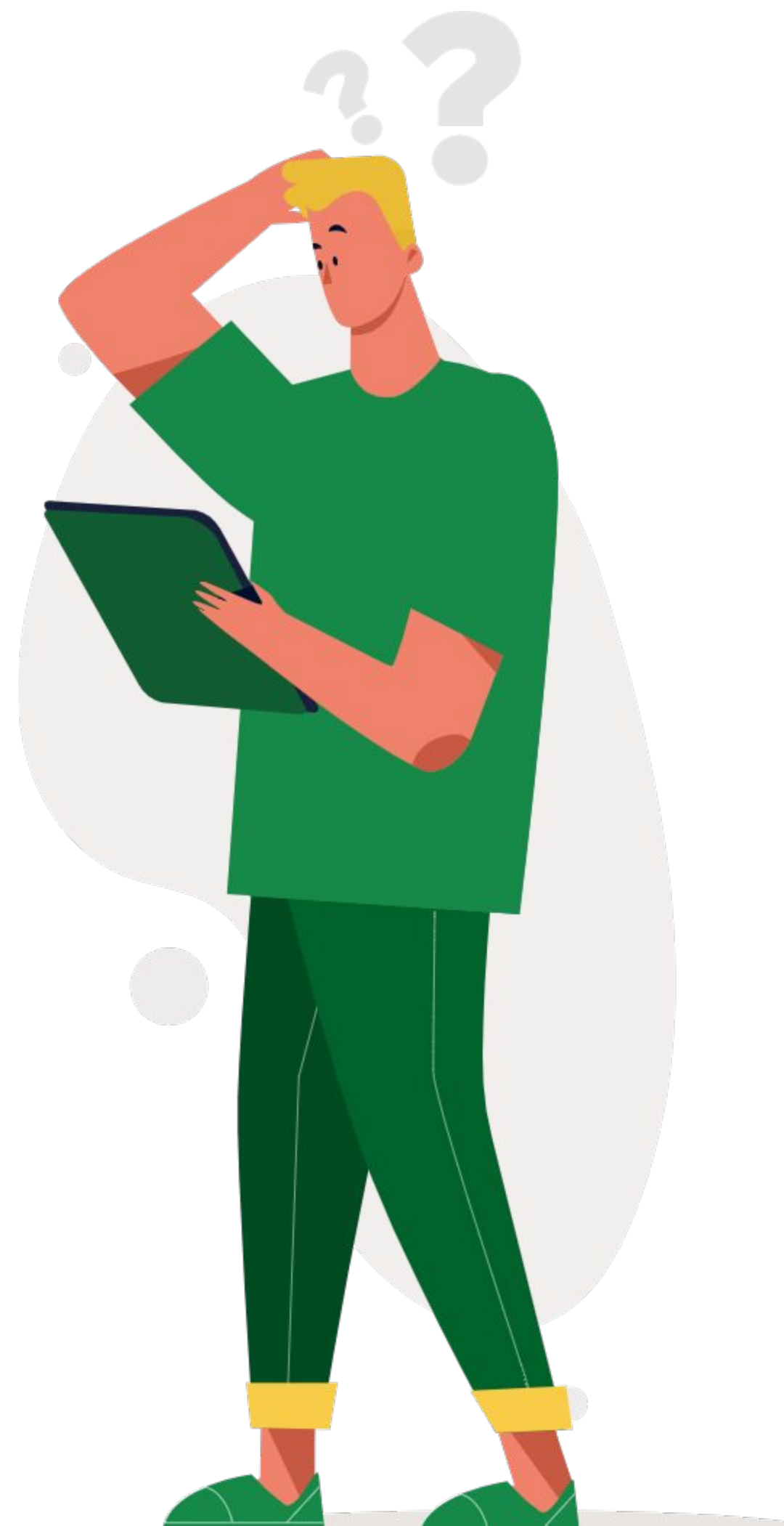


Моя Школа

сертифицированные IT курсы

Введение в тестирование



Преподаватель

Имя: **Татьяна Русецкая**

Компания: **ISSOFT**



Давайте знакомиться!

Расскажите о себе:

- 1. Почему выбрали профессию тестировщика или чем интересно данное направление?**
- 2. Кем работали или чем занимались до курсов по тестированию?**
- 3. Как видите компанию своей мечты?**
- 4. (необязательный, но важный) Как обстоят дела с английским?**

О курсе

2 блока - технический и основной по 8-мь занятий каждый + «микс» с практическим блоком

Основной блок:

- **Введение в тестирование, основные термины, аксиомы, принципы**
- **Жизненный цикл ПО и жизненный цикл тестирования**
- **Виды, техники и классификация тестирования**
- **Баг-трекинг-овые системы**
- **Тестовая документация (+Тестирование требований и документации)**
- **Техники тест-дизайна**
- **Web-тестирование**
- **Мобильное тестирование**



О курсе

2 блока - технический и основной по 8-мь занятий каждый + «микс» с практическим блоком

Технический блок:

- **Web-технологии**
- **API тестирование**
- **Chrome dev tools**
- **База данных**
- **Сети, протоколы**
- **Виртуализация**

План занятия

1

Базовые понятия тестирования

2

Кто такой тестировщик?

3

10 заповедей тестировщика

4

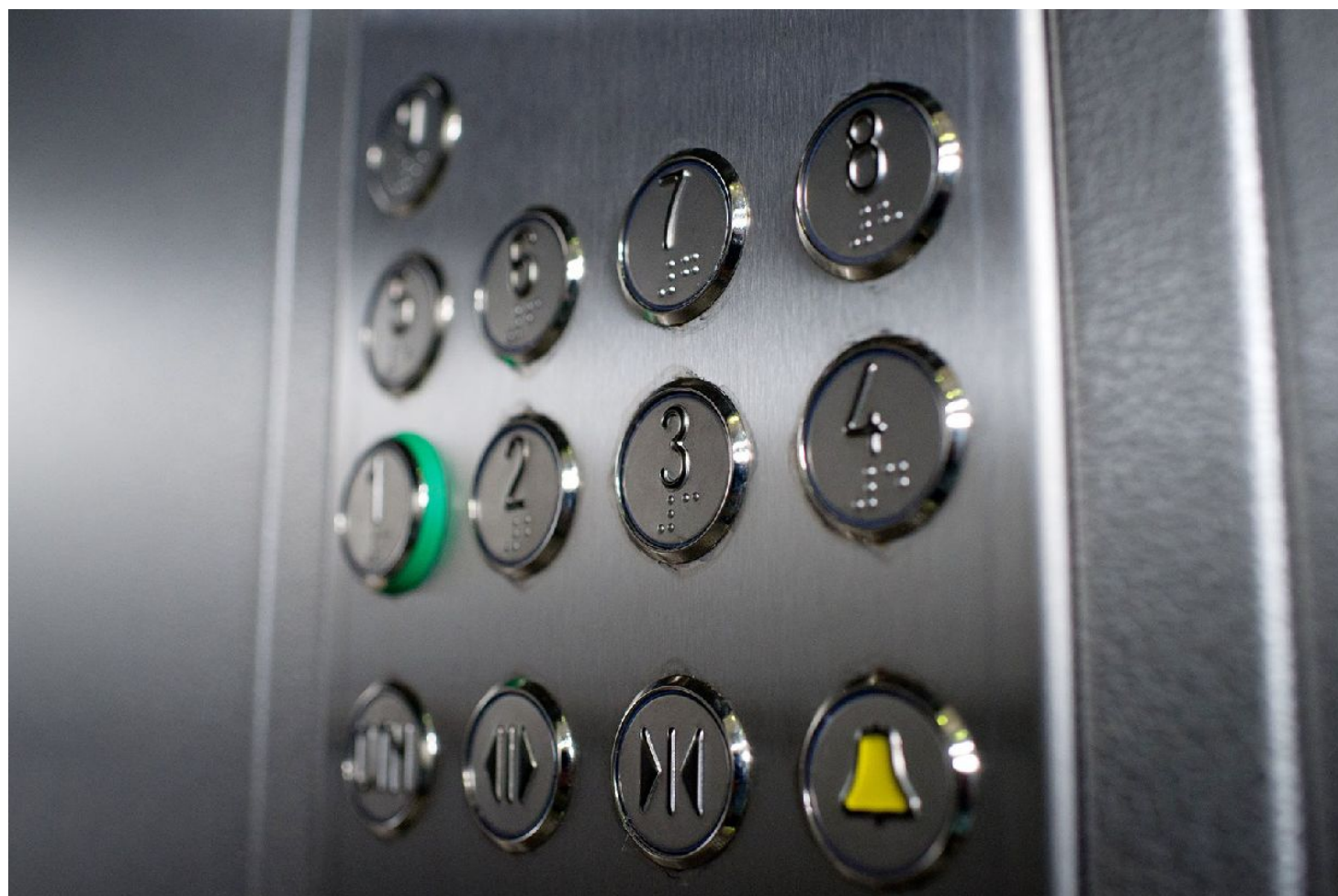
7 принципов тестирования

5

Аксиомы тестирования

6

**Testing, QC QA —
разница этих понятий**



Ты тоже пытался нажать все кнопки одновременно?

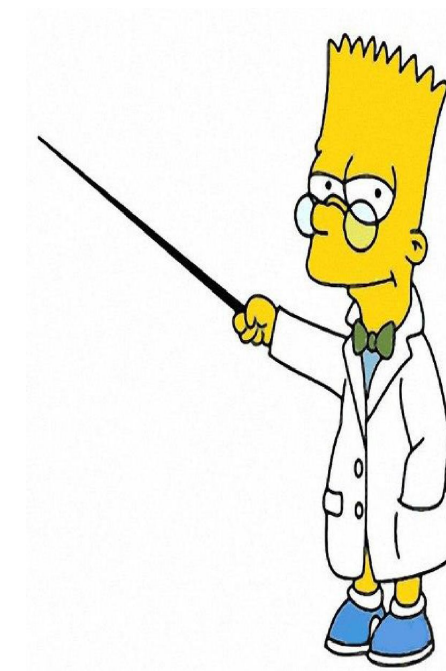


Актуальность профессии:

Обеспечение качества - является неотъемлемой частью производства, т.к. позволяет делать продукт конкурентоспособным на рынке.

Тенденции для развития профессии тестировщика:

- Автоматизация станет важным навыком в тестировании.
- Повсеместное использование тестирования в облаке.
- Рост потребности в тестировщиках больших данных.
- IoT тестирование (это комбинация различных платформ, протоколов, аппаратного обеспечения и сетевых средств).



Моя  Школа

сертифицированные IT курсы

Кто такой тестировщик?



Тестировщик (или специалист по тестированию программного обеспечения) — это специалист, который проверяет качество программного обеспечения и уровень его соответствия заранее определённым потребностям и ожидаемым результатам.

Он руководствуется технической документацией, а иногда и просто здравым смыслом. Ведь тестировщик пытается посмотреть на приложение глазами **не только разработчика, но и пользователя.**

Поэтому иногда он «вмешивается» в процесс разработки и предлагает добавить в программу более удобные функции.



Какие этапы вы, скорее всего, пройдёте на работе тестировщиком «с нуля»

1. Первые 2-3 месяца — обучение
2. От нескольких месяцев до 2 лет — ручное тестирование. Вы будете выполнять большинство видов тестов в ручном режиме, начнёте составлять свои тест-планы и взаимодействовать с разработчиками
3. От 2 лет и далее — квалифицированный специалист, готовый работать в офисе и на стороне клиента, тестировать крупные проекты, а не только фрагменты и модули, глубоко знающий свою специализацию и не очень глубоко все остальные, способный работать не только с профильным ПО, но и тестировать взаимодействия внутри различных конфигураций оборудования, ПО и интеграций.

Обычно тестировщик занимается проверкой самой программы, её работоспособности и внешнего вида. Но порой к этим задачам прибавляется проверка и технических инструкций, и баз данных, и бизнес-процессов, и нужд заказчика, и даже работы команды разработки.

Плюсы профессии тестировщика

1. До определённого момента работа тестировщика проще остальных технических специальностей и доступна многим, кому интересна ИТ-сфера.
2. Потребность в тестировщиках не исчезнет до тех пор, пока есть информационные технологии, связь, интернет, роботы, автопилоты и т.д.
3. Тестирование — не столь изученная область, как программирование. Если у вас есть талант и трудолюбие, вы сможете сказать своё слово миру (написать книгу, создать методологию, преподавать и т.д.).
4. Карьера тестировщика довольно свободная: можно сидеть в офисе с гибким графиком, можно работать удалённо, а можно стать фрилансером, набрать проектов и тестировать их по сдельной оплате.
5. Тестировщику легче вернуться на работу после долгого перерыва, например, из опыта создания своего стартапа, фриланса, декрета, иных обстоятельств.
6. Работу в тестировании можно сочетать с учебой без вреда для обоих видов деятельности.
7. Вы научитесь «видеть продукт» со всех сторон, узнаете все функциональные возможности, посмотрите на софт глазами инженера и глазами потребителя. Это прямой путь в менеджеры продукта. Общая картина продукта поможет вам в любом случае — например, если вы решитесь уйти в разработку.

Минусы профессии тестировщика

1. Команда недолюбливает тестировщиков
2. На первом этапе вы работаете с повторяющимися задачами, иногда работать становится невыносимо скучно.
3. Тестировщики ищут ошибки разработчиков, искать ошибки тестировщиков никому. Поэтому вы будете крайними в некоторых неприятных ситуациях.
4. Работа тестировщиков часто бывает незаметна руководству — придётся привыкнуть быть серым кардиналом, невидимым героем.
5. Сверхурочная работа — бич тестировщиков. Рано или поздно вам будет нужно срочно оттестировать релиз или сборку, которую нужно выкатить завтра или «вот прямо сейчас» или же остаться и проверить внесённые программистами исправления. И вы останетесь, а вот оплачивается такой героизм далеко не всегда.
6. На тестировщиках лежит огромный груз ответственности за полноту и охват тест-планов — если что-то упустить, отвечать уже придётся за пропущенные баги.



Спект задач тестировщика включает:

1

Разработку плана тестирования

2

Написание тест-кейсов

3

Поиск дефектов и их описание в баг-трекинговой системе

4

Общение с заказчиками и командой разработки

5

Контроль за устранением ошибок

6

Мониторинг жизненного цикла дефектов

7

Анализ и оптимизацию процессов тестирования

8

Составление отчётности и ведение тестовой документации

10 заповедей тестировщика:

1

Ходите разными путями

выполнять один и тот же сценарий меняя действия

2

Не думайте за пользователя

невозможно продумать все тестовые сценарии всех пользователей

3

Спрашивайте очевидное

каждый человек воспринимает одно и тоже по-разному

4

Инструменты под задачу

для разной задачи может быть актуален различный инструмент

5

Локализуйте баги

правило хорошего тона –указывать, где, в какой части приложения найден баг

10 заповедей тестировщика от Андрея Мясникова:

6

Не придумывайте, а спрашивайте

читайте документацию,
задавайте вопросы ВА/РМ

7

Отслеживайте баг-трекер

изучайте баги, узнавайте в каких
модулях их больше

8

Выходите за рамки компетенции

не стоит бояться сложных
задач – это возможность,
это развитие

9

Смотрите документацию

документация – главный друг и
источник пробелов

10

Будьте ответственными

Вы – часть большого механизма. От вашей
работы зависит, отчасти, итоговый результат

Тестирование программного обеспечения (testing)*** —

процесс исследования, испытания программного обеспечения (ПО) с целью получения информации о качестве продукта.

процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта. (по Куликову)

процесс, содержащий в себе все активности жизненного цикла, как **динамические**, так и **статические**, касающиеся планирования, подготовки и оценки программного продукта и связанных с этим результатов работ с целью **определить, что они соответствуют описанным требованиям, показать, что они подходят для заявленных целей и для определения дефектов.** (по ISTQB)

Цели тестирования*** —

- проверка соответствия ПО предъявляемым требованиям,
- обеспечение уверенности в качестве ПО,
- поиск ошибок в программном обеспечении, которые должны быть выявлены - до того, как их обнаружат пользователи программы,
- предоставление информации о качестве ПО конечному заказчику.

Для чего же проводится тестирование ПО?

- Для проверки соответствия требованиям.
- Для обнаружение проблем на более ранних этапах разработки и предотвращение повышения стоимости продукта.
- Обнаружение вариантов использования, которые не были предусмотрены при разработке.

Причины возникновения ошибок

- ✓ Проблемы в коммуникациях между членами команды
- ✓ Сложность программного обеспечения
- ✓ Изменение требований
- ✓ Некачественное документирование кода и временные рамки
- ✓ Ошибки программистов – человеческий фактор
- ✓ Баги в инструментах для разработки программного обеспечения
- ✓ Ошибки тестировщиков
- ✓ Некачественный контроль версий кода
- ✓ Архитектура программного обеспечения
- ✓ Недостаток финансирования

Баг* (bug)** — это любое отклонение фактического поведения системы от ожидаемого.

Качество программного обеспечения*** (software quality)

— это степень, в которой система, компонент или процесс удовлетворяют потребностям или ожиданиям заказчика или пользователя.



Обеспечение качества*** (QA = Quality Assurance)

это процесс или результат формирования требуемых свойств и характеристик продукции по мере её создания.

Чем же формируются эти характеристики сами по себе?

До промышленная эпоха:

законами
обществом
стандартами

Промышленная эпоха:

Стандарты (ISO, госты)
закон (отдельные отрасли)
общество (конкуренция)

Настоящее время

конкуренция
стандарты
закон (отдельные отрасли)

Контроль качества*** (QC = Quality Control)

это совокупность **действий**, проводимых над ПО в процессе разработки, для получения информации о его актуальном состоянии в аспектах **ГОТОВНОСТИ К выпуску, соответствия зафиксированным требованиям и соответствия заявленному уровню качества этого ПО.**

Так в чем же разница между **QA - тестированием** – и **QC**?

Quality Assurance обеспечивает правильность и предсказуемость процесса.

QA — планирование того, как сделать продукт качественным.

Quality Control предполагает контроль соблюдения требований.

QC — контроль за данными действиями.

Тестирование обеспечивает сбор статистических данных и внесение их в документы, созданные в рамках QC-процесса.

Testing — конкретные действия по проверке качества продукта.

Quality Assurance

Quality Control

Тестирование

Testing (тестирование) - это самый низкий уровень - прохождение тест-кейсов и локализация дефектов.

QC - следующий уровень - контроль качества продукта - анализ результатов тестирования и качества “билдов” в процессе разработки.

QA - решает более глобальные задачи. Анализируя работу тестировщиков и QC, в случае возникновения проблем, вовремя находит пути ее решения и не дает ей развиться и повлиять на качество продукта.

Если провести аналогию с процессом конструирования велосипеда, получим такую картину:

QA

проверяет соответствие всех этапов конструирования нашего велосипеда определенным стандартам качества, начиная с планирования и создания чертежей и заканчивая сборкой уже готового велосипеда

QC

направлен на проверку из правильных ли материалов он сделан, с применением нужных методик и инструментов или нет

Тестирование

определяет работают ли все детали и сам велосипед в целом так, как мы ожидаем



Quality Assurance

Комплекс мероприятий, который охватывает все технологические аспекты на всех этапах разработки, выпуска и введения в эксплуатацию программных систем для обеспечения необходимого уровня качества программного продукта

Фокус в большей степени на процессы и средства, чем на непосредственно исполнение тестирования системы

Процессно-ориентированный подход

Превентивные меры

Подмножество процессов Software Test Life Cycle — цикла тестирования ПО

Quality Control

Процесс контроля соответствия разрабатываемой системы предъявляемым к ней требованиям

Фокус на исполнение тестирования путём выполнения программы с целью определения дефектов с использованием утвержденных процессов и средств

Продуктно-ориентированный подход

Корректирующий процесс

Подмножество процессов QA

Тестирование

Процесс, отвечающий непосредственно за составление и прохождение тест-кейсов, нахождение и локализацию дефектов и т.д.

Фокус на исполнение тестирование как такового

Продуктно-ориентированный подход

Превентивный процесс

Подмножество процессов QC



РЕЛИЗ

release, RTM

выпуск программного продукта на рынок, например, размещение мобильного приложения в App Store или Google Play.



БИЛД

от англ. to build — строить

финальный вариант программного продукта или его элемента, который готов к тестированию.

СРЕДА ТЕСТИРОВАНИЯ

Test Env

среда, в которой работают тестировщики (проверяют функционал, проводят smoke и регрессионные тесты, воспроизводят



Верификация (verification)

Верификация (verification)
— оценка ПО или его компонентов с точки зрения соответствия всем заявленным к нему требованиям.

VS

Валидация (validation) — это проверка работоспособности функциональности приложения.

Валидация (validation)

Что является валидацией, а что - верификацией на скриншоте?

Вход тут

Имя

Фамилия

Город

Дом

Где жить

Ключ от сейфа

Верификация - наличие полей. Все поля должны соответствовать спецификации. Их наличие определено дизайнерами, создающими макеты. Необходимая информация вносится в ТЗ, а в случае отсутствия такового – желательно иметь доступ к мокапам.

Валидация - вводимая в поля информация и ее соответствие спецификации.

Какая разница между валидацией и верификацией?

Верификация	Валидация
Делаем ли мы продукт правильно?	Делаем ли мы правильный продукт
Реализована ли вся функциональность	Правильно ли реализована
Верификация происходит раньше и включает проверку правильности написания документации, кода и т.д.	Валидация происходит после верификации, и, как правило, отвечает за оценку продукта в целом.
Производится разработчиками	Производится тестировщиками
Включает статический анализ - инспектирование кода, сравнение требований и т.п.	Включает динамический анализ - выполнение программы для сравнения её реальной работы с установленными требованиями.
Проверяется наличие чего-нибудь	Проверяется работоспособность этого «чего-нибудь»
проверка того, что продукт отвечает ожиданиям и потребностям пользователей	подтверждение того, что определенные требования были выполнены

Артефакты — это документы, которые используют в процессе тестирования.

Какими они бывают?

Спецификация

(*specification, спека*) — детализированное описание работы приложения, которое включает технические свойства.



Баг-репорт

(*bug report, отчёт об ошибке*) — описание действий или условий, которые привели к выявлению дефекта. Подобные отчёты создают в баг-трекинговой системе (*bug tracking system, система отслеживания ошибок*). Это программа для описания и контроля дефектов.

План тестирования

(*test plan*) — в этом документе содержатся все данные о проводимых проверках.

(описание программного продукта, стратегия тестирования, сроки выполнения поставленных задач, используемые в процессе инструменты и оборудование, оценка потенциальных рисков и прочее).

Артефакты — это документы, которые используют в процессе тестирования.

Какими они бывают?

Чек-лист

(*checklist*, контрольный список) — перечень параметров, которые нуждаются в проверке.



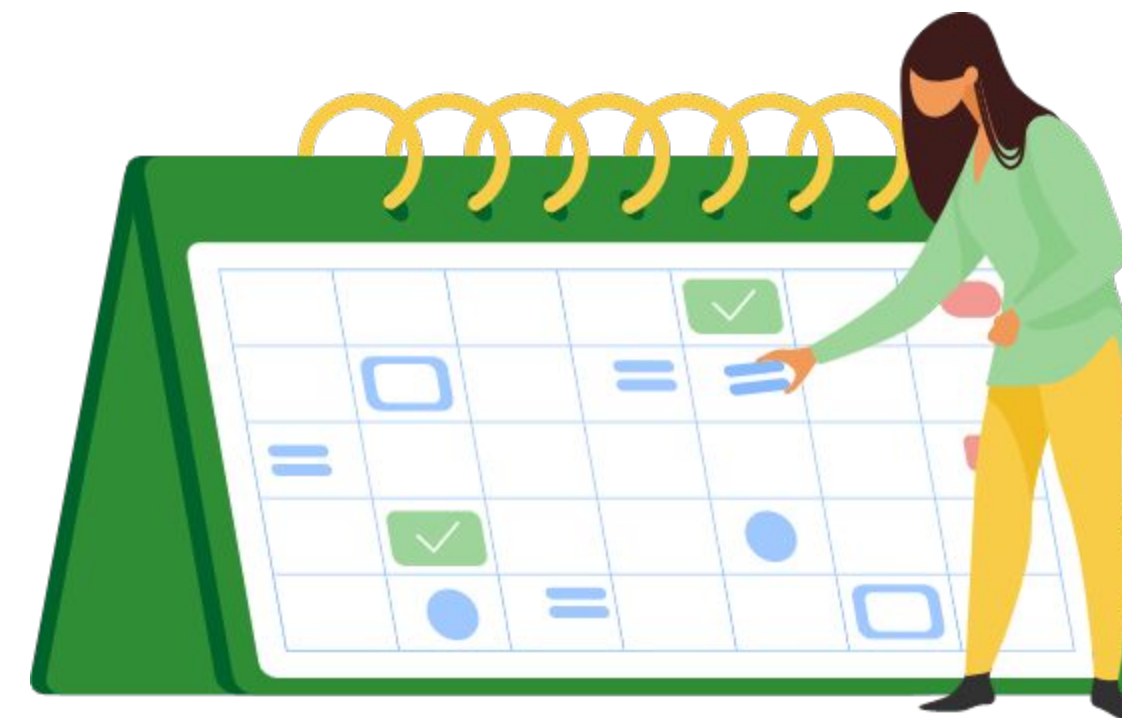
Тест-кейс

(*test case*, тестовый случай) — своего рода сценарий или описание последовательности шагов при проведении тестирования.

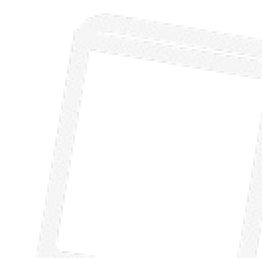
Тестовый набор

(*test suite*) — несколько тест-кейсов, которые объединены по типу тестирования или другим признакам.

Планирование (planning) — непрерывный процесс принятия управленческих решений и методической организации усилий по их реализации с целью обеспечения качества некоторого процесса на протяжении длительного периода времени.



Отчётность (reporting) — это сбор и распространение информации о результатах работы (включая текущий статус, оценку прогресса и прогноз развития ситуации).





Принципов тестирования

1
Тестирование демонстрирует наличие дефектов, а не их отсутствие

2
Исчерпывающее тестирование недостижимо

3
Раннее тестирование сохраняет время и деньги

4
Кластеризация дефектов

5
Парадокс пестицида

6
Тестирование зависит от контекста

7
Заблуждение об отсутствии ошибок

Спасибо за внимание!)