

Струков Максим

Путеводитель в мир ручного тестирования

Открытие двери в тестирование
программного обеспечения

Максим Струков

Путеводитель в мир ручного тестирования: Открытие двери в тестирование программного обеспечения

*http://www.litres.ru/pages/biblio_book/?art=69489553
SelfPub; 2023*

Аннотация

Добро пожаловать в захватывающий мир тестирования программного обеспечения! "Путеводитель в мир ручного тестирования: Открытие двери в тестирование программного обеспечения" – это увлекательное путешествие в профессию ручного тестировщика, предназначенное для всех, кто хочет разобраться в сущности этой важной и захватывающей области. В этой книге вы найдете все необходимые знания и навыки, чтобы стать успешным тестировщиком и принять активное участие в обеспечении качества программного обеспечения. Мы начнем с основ, рассмотрим терминологию и типы тестирования, познакомимся с жизненным циклом тестирования и ролью ручного тестирования в процессе разработки ПО. В дальнейшем вы узнаете о том, как подготовиться к тестированию, создавать тестовую документацию, определять стратегию тестирования и

выбирать подходящие методы. Мы разберемся с тестированием функциональности, пользовательского интерфейса, функций и возможностей ПО, а также обсудим валидацию данных и обработку ошибок.

Содержание

Глава 1: Введение в ручное тестирование	6
Глава 2: Основы ручного тестирования	11
Глава 3: Подготовка к тестированию	18
Конец ознакомительного фрагмента.	20

Максим Струков
Путеводитель в мир
ручного тестирования:
Открытие двери
в тестирование
программного обеспечения

Эта книга предоставляет комплексное введение в мир ручного тестирования и поможет читателю понять, что такое профессия тестировщика, какие умения и знания необходимы для успешной работы в этой области, а также как развиваться профессионально. Она поможет решить, подходит ли данная профессия для человека и является ли она интересной сферой для дальнейшего обучения и развития карьеры.

Глава 1: Введение в ручное тестирование

В современном мире программного обеспечения, где каждый аспект нашей жизни становится все более цифровым и автоматизированным, тестирование программного обеспечения (ПО) играет важнейшую роль в обеспечении качества и надежности различных продуктов и сервисов. Тестирование ПО – это процесс проверки программного продукта на соответствие определенным требованиям и стандартам, а также выявление ошибок, уязвимостей и дефектов.

1.1 Что такое тестирование ПО?

Тестирование программного обеспечения – это систематический процесс, в рамках которого выполняются различные действия с целью обнаружения ошибок и проверки работоспособности программного продукта. Целью тестирования является убедиться, что ПО функционирует правильно, соответствует заявленным требованиям и ожиданиям пользователей, а также не допускает ошибок и уязвимостей, которые могут нанести ущерб его функциональности и безопасности.

Тестирование ПО включает в себя выполнение различ-

ных тестовых сценариев, проверку функциональности, интерфейсов, производительности, безопасности и других характеристик программного продукта. Тестирование может быть проведено как вручную, так и с использованием автоматизированных инструментов и скриптов.

1.2 Роль ручного тестирования в процессе разработки ПО

Ручное тестирование является неотъемлемой частью процесса разработки программного обеспечения. При его использовании тестировщики вручную выполняют тестовые сценарии, проверяют работоспособность и корректность функционирования ПО. Ручное тестирование позволяет тестировщикам более глубоко понять продукт, его особенности и потенциальные уязвимости.

Одной из основных преимуществ ручного тестирования является его адаптивность. Тестировщик может быстро адаптироваться к изменениям в продукте или требованиям, вносить корректировки в тестовые сценарии, что позволяет более эффективно обнаруживать дефекты и ошибки.

Ручное тестирование также играет важную роль в ранней стадии разработки ПО, когда автоматизированные средства тестирования могут быть еще недоступны или неэффективны. Тестировщики могут выполнять тестирование уже на прототипах или MVP (минимально жизнеспособный про-

дукт), что позволяет рано выявлять и устранять дефекты и снижать риски для проекта.

1.3 Основные задачи и цели ручного тестирования

Основной задачей ручного тестирования является проверка функциональности, стабильности и безопасности программного обеспечения. В ходе ручного тестирования тестировщики проверяют, соответствует ли продукт заявленным требованиям и ожиданиям пользователей, выполняет ли он свои функции правильно и надежно.

Цели ручного тестирования:

1. Проверка функциональности: Ручное тестирование помогает выявить ошибки в функциональности программного продукта. Тестировщики выполняют тестовые сценарии, проверяют, что все функции работают правильно и в соответствии с заявленными требованиями.

2. Проверка стабильности: Тестирование проводится с целью выявить возможные сбои, ошибки или неожиданное поведение ПО. Тестировщики ищут проблемы, которые могут привести к зависаниям, падениям или некорректной работе продукта.

3. Проверка безопасности: Ручное тестирование позволяет выявить уязвимости в безопасности программного обеспечения. Тестировщики проверяют, что система защищена

от различных атак, обеспечивает аутентификацию и авторизацию пользователей, а также надежное шифрование данных.

4. Проверка пользовательского опыта: Тестировщики оценивают удобство и интуитивность пользовательского интерфейса, обнаруживают возможные проблемы в навигации и взаимодействии с продуктом.

1.4 Важность ручного тестирования в контексте автоматизации

Автоматизация тестирования ПО становится все более популярной и широко используется в современных проектах разработки. Автоматизированные тесты позволяют значительно ускорить процесс тестирования, повысить его эффективность и точность, а также сократить рутинные ручные действия.

Однако, несмотря на все преимущества автоматизации, ручное тестирование остается необходимым и ценным элементом процесса тестирования ПО. Во-первых, автоматизация не может заменить человеческий фактор – интуицию, креативность и аналитические способности тестировщиков. Тестировщик может обнаружить нестандартные ситуации и неочевидные проблемы, которые автоматизированные тесты могут упустить.

Во-вторых, ручное тестирование необходимо для провер-

ки пользовательского опыта. Тестировщик может эмулировать действия и реакции реального пользователя, проверить, насколько продукт удобен и интуитивен для пользователя.

Наконец, ручное тестирование остается актуальным на ранних стадиях разработки, когда продукт может быть не до конца готов или его функциональность может меняться. Тестировщик может быстро адаптироваться к изменениям, проводить тестирование "на лету" и выявлять ошибки до того, как они перейдут в автоматизированные тесты.

В заключение, ручное тестирование является неотъемлемой и важной частью процесса тестирования ПО. Оно позволяет более глубоко понять продукт, выявить уязвимости, проверить пользовательский опыт и адаптироваться к изменениям. Вместе с тем, автоматизация тестирования также важна и эффективна в определенных случаях. Комбинация ручного и автоматизированного тестирования обеспечивает высокий уровень качества и надежности программного обеспечения, что является ключевым фактором для успеха проекта и удовлетворенности пользователей.

Глава 2: Основы ручного тестирования

Тестирование программного обеспечения – это сложный и многогранный процесс, который требует от специалистов глубокого понимания различных аспектов и методов тестирования. В этой главе мы поговорим о основах ручного тестирования, которые являются фундаментом для успешной работы тестировщика.

2.1 Терминология тестирования, которую необходимо знать

Перед тем как мы начнем рассматривать основы ручного тестирования, давайте ознакомимся с ключевой терминологией, которая используется в этой области.

Тестирование ПО – это процесс проверки программного продукта на соответствие определенным требованиям и выявление дефектов.

Тестовый случай – это набор шагов и данных, которые тестировщик выполняет для проверки определенной функциональности или характеристики ПО.

Дефект – это неполадка или ошибка в программном обеспечении, которая приводит к некорректной работе продукта.

Репорт (отчет) о дефекте – это документ, в котором описывается найденный дефект, его характеристики, шаги для его воспроизведения и другая информация.

QA (Quality Assurance) – отдел или команда, которая отвечает за обеспечение качества продукта, включая тестирование.

Тестировщик – специалист, который занимается выполнением тестовых сценариев и выявлением дефектов в программном обеспечении.

Regression Testing (регрессионное тестирование) – повторное тестирование продукта после внесения изменений с целью убедиться, что новые функции не повлияли на работу существующих.

Smoke Testing (смоук-тестирование) – быстрая проверка основных функций продукта для выявления критических ошибок перед основным тестированием.

Load Testing (тестирование нагрузки) – проверка работоспособности продукта при максимальных нагрузках и условиях высокой нагрузки.

Security Testing (тестирование безопасности) – проверка продукта на уязвимости и защищенность от атак.

2.2 Основные типы тестирования и их отличия

Существует множество различных типов тестирования, каждый из которых направлен на проверку определенных ас-

пектов программного обеспечения. Рассмотрим основные из них и их отличия.

1. Функциональное тестирование – проверка основных функций продукта на соответствие требованиям. Здесь тестируются такие аспекты, как работа кнопок, ввод и вывод данных, обработка запросов и другие функции.

2. Нагрузочное тестирование – проверка работоспособности продукта при максимальных нагрузках и условиях высокой нагрузки. Этот вид тестирования позволяет определить, как продукт будет вести себя при большом количестве одновременных пользователей или транзакций.

3. Интеграционное тестирование – проверка взаимодействия различных компонентов программного обеспечения и их корректной работы вместе. В ходе этого тестирования выявляются возможные конфликты между компонентами системы и проблемы с их взаимодействием.

4. Приемочное тестирование – проверка продукта на соответствие заявленным требованиям и его готовности к выпуску. Приемочное тестирование проводится с целью удостовериться, что ПО полностью соответствует требованиям заказчика или конечного пользователя.

5. Регрессионное тестирование – проверка продукта после внесения изменений для убеждения в отсутствии новых дефектов и ошибок. Это необходимо для того, чтобы убедиться, что изменения не повлияли на стабильность и работоспособность остальных частей программы.

6. Тестирование совместимости – проверка работы продукта на различных платформах, браузерах и устройствах. В ходе этого тестирования убеждаются в том, что ПО корректно функционирует на различных конфигурациях и не зависит от конкретных настроек системы.

7. Тестирование безопасности – проверка продукта на уязвимости и защищенность от атак. Цель этого тестирования – обнаружить и исправить потенциальные уязвимости, которые могут быть использованы злоумышленниками.

2.3 Жизненный цикл тестирования: от планирования до отчетности

Тестирование программного обеспечения проходит через определенные этапы, известные как жизненный цикл тестирования. Рассмотрим каждый из них подробнее.

1. Планирование: В этом этапе определяются цели и задачи тестирования, составляется тестовая документация и план тестирования. Планирование – это важный этап, на котором определяются основные аспекты тестирования, а также распределяется время и ресурсы для его выполнения.

2. Проектирование тестов: Здесь разрабатываются тестовые сценарии, тестовые случаи и создается тестовая среда. В ходе этого этапа определяется, какие конкретные тесты будут проводиться и какие данные будут использоваться для их выполнения.

3. Выполнение тестов: На этом этапе тестировщики выполняют тестовые сценарии и проверяют работоспособность продукта. Тестировщики должны быть внимательны и ответственны, чтобы обнаружить все возможные дефекты и ошибки.

4. Анализ результатов: После выполнения тестов проводится анализ результатов, выявляются тенденции и проблемы. Тестировщики анализируют полученные данные, сравнивают их с заранее определенными ожиданиями и выявляют несоответствия.

5. Исправление дефектов: После обнаружения дефектов разработчики исправляют их, а затем проводится повторное тестирование. Важно, чтобы исправления дефектов были тщательно протестированы, чтобы избежать появления новых проблем.

6. Повторное тестирование: После внесения изменений проводится повторное тестирование для проверки исправленных дефектов. В ходе повторного тестирования убеждаются, что все дефекты были успешно исправлены и продукт работает стабильно.

7. Отчетность: Формируются отчеты о тестировании, в которых фиксируются результаты и обнаруженные дефекты. Отчетность является важной частью тестирования, так как она предоставляет информацию о состоянии продукта и оценку его качества.

2.4 Роли и обязанности тестировщика

Тестировщик – это ключевой игрок в процессе обеспечения качества программного обеспечения. У него есть свои роли и обязанности, которые необходимо выполнять для эффективного тестирования.

1. Подготовка тестовой среды: Тестировщик должен создать подходящую тестовую среду, в которой будет проходить тестирование. Тестовая среда может включать в себя определенные версии операционных систем, баз данных, браузеров и другого программного обеспечения.

2. Выполнение тестовых сценариев: Основная задача тестировщика – выполнение тестовых сценариев и проверка работоспособности продукта. Тестировщик должен следовать заданным шагам и использовать определенные данные для проверки функциональности продукта.

3. Регистрация дефектов: Если тестировщик находит дефекты или ошибки, он должен тщательно их документировать и зарегистрировать. Дефекты должны быть описаны таким образом, чтобы разработчики могли легко воспроизвести и исправить их.

4. Сотрудничество с разработчиками: Тестировщик должен активно сотрудничать с разработчиками, чтобы обсудить найденные дефекты и следить за их исправлением. Взаимодействие между тестировщиками и разработчиками по-

могает снизить время исправления дефектов и повысить качество продукта.

5. **Отчетность:** Тестировщик должен составлять отчеты о тестировании и предоставлять информацию о результатах. Отчеты могут включать информацию о количестве протестированных функций, найденных дефектах, степени покрытия тестами и другие метрики.

6. **Анализ данных:** Тестировщик должен уметь анализировать данные и выявлять тенденции и проблемы в продукте. Анализ данных позволяет определить области, которые требуют дополнительного тестирования или исправления.

7. **Улучшение процесса:** Тестировщик может предлагать улучшения в процессе тестирования и оптимизировать его. Важно постоянно стремиться к улучшению процесса тестирования и применять новые методы и технологии для повышения эффективности и качества тестирования.

Тестирование ПО – это сложный процесс, который требует от тестировщика высокой квалификации, внимательности и ответственности. Основы ручного тестирования, которые мы рассмотрели в этой главе, являются фундаментом для дальнейшего развития и успеха в этой профессиональной области.

Глава 3: Подготовка к тестированию

Подготовка к тестированию является важным этапом в процессе обеспечения качества программного обеспечения. Это время, когда тестировщики знакомятся с проектом, определяют особенности и требования, разрабатывают тестовую документацию и выбирают стратегию тестирования. В этой главе мы рассмотрим все этапы подготовки к тестированию и узнаем, как эффективно подготовиться к тестированию ПО.

3.1 Знакомство с проектом и его особенностями

Первый шаг в подготовке к тестированию – это полное знакомство с проектом и его особенностями. Тестировщики должны иметь ясное представление о том, что именно должно быть протестировано и какие ожидаются результаты. Ниже перечислены основные этапы знакомства с проектом:

1. Изучение требований: Тестировщики должны внимательно изучить требования к проекту. Требования определяют функциональность и характеристики продукта, которые должны быть протестированы.
2. Проведение собеседования с заказчиком: Собеседова-

ние с заказчиком поможет лучше понять цели и ожидания проекта, а также уточнить детали, которые могут повлиять на тестирование.

3. Изучение документации проекта: Тестировщики должны ознакомиться с доступной документацией проекта, такой как спецификации, архитектурные диаграммы и документы с описанием функциональности.

4. Понимание архитектуры: Важно понимать архитектуру проекта, чтобы определить критические компоненты и их взаимодействие. Это поможет лучше спланировать тестирование.

5. Идентификация рисков: Тестировщики должны выявить потенциальные риски и проблемы, которые могут возникнуть во время тестирования и влиять на результаты.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.