

Основы тест-аналитики

Исследовательское тестирование



На этом уроке

1. Рассмотрим способы тестирования без требований
2. Научимся создавать отчет об исследовательском тестировании

Оглавление

[Как тестировать без требований?](#)

[Что такое исследовательское тестирование?](#)

[Исследовательское и сценарное тестирование](#)

[Преимущества сценарного тестирования](#)

[Преимущества исследовательского тестирования](#)

[Ограничения для исследовательского тестирования](#)

[Тур Дж. Уиттакера - подходы к исследовательскому тестированию](#)

[Тур по путеводителю.](#)

[Денежный тур.](#)

[Интеллектуальный тур.](#)

[Отчет об исследовательском тестировании](#)

[Свободное \(интуитивное, ad-hoc\) тестирование](#)

[Виды свободного тестирования](#)

[Глоссарий](#)

[Домашнее задание](#)

[Дополнительные материалы](#)

[Используемые источники](#)

Как тестировать без требований?

В идеальном мире тестировщика продуктовая документация тщательно проработана, доступна, непротиворечива, поддается тестированию, т.е. соответствует атрибутам требований. В жизни бывают ситуации, когда документации нет, она недоступна или описана поверхностно, а тестировать надо здесь и сейчас. В этом случае можно воспользоваться следующими приемами.

1. Работать с приложением, как будто вы его пользователь
2. Опирается на бизнес-процессы
3. Декомпонировать функциональность
4. Уточнить информацию у других участников разработки
5. Использовать оракулы, например, по методике FEW HICCUPS
6. Проводить мозговой штурм
7. Применять исследовательское тестирование

Что такое исследовательское тестирование?

Исследовательское тестирование – это одновременное изучение программы, проектирование и выполнение тестов. Такой подход противоположен сценарному, когда список проверок составлен до прохождения тест-кейсов. Исследовательские тесты не определены заранее и не выполняются в точном соответствии с планом.

Исследовательское тестирование не является методикой тестирования. Это подход или образ мыслей, который можно применить к любой методике тестирования.

Исследовательское тестирование не должно выполняться небрежно, в спешке и без подготовки. Исследовательский подход требует тщательной подготовки, а знания и умения тестировщика - это важная форма подготовки. Исследовательское тестирование может проводиться вручную, а может осуществляться с широким применением средств автоматизации, т.е. любых вспомогательных инструментов тестирования.

Когда следует применять исследовательское тестирование?

- Нужна быстрая обратная связь о новом продукте.
- Нужно быстро изучить продукт.
- Сценарное тестирование не находит баги и требует разнообразия.
- Следует принять решение о необходимости покрытия области сценарными тестами.
- Требований нет, они не полны или устарели.
- Продукт маленький, и разработка тестовых сценариев займет больше времени, чем сам процесс тестирования.

Исследовательское и сценарное тестирование

Преимущества сценарного тестирования

1. Тестирование можно планировать: тест-кейсы можно легко поделить между различными тестировщиками или командами.
2. Важные кейсы гарантированно будут пройдены.
3. Возможно оценить процент покрытия требований тестами.
4. Тестовые сценарии можно использовать для обучения новых сотрудников.
5. Тестовые сценарии помогают проводить приемочные испытания и определять критерии готовности.

Преимущества исследовательского тестирования

1. Нестандартные ходы выявляют нестандартные дефекты.
2. Не тратится время на описание всех сценариев.
3. Не нужна поддержка тестовых сценариев.
4. Не наступает “эффект пестицида”.
5. Можно тестировать без требований
6. Тесты могут стать интереснее и креативнее.

Ограничения для исследовательского тестирования

Чистое исследовательское тестирование хорошо работает на небольших краткосрочных проектах или на начальных этапах жизни продукта. В первом случае объемная тестовая документация может не оправдать того времени, которое будет затрачено на ее написание. Во втором - требования могут быть очень размытые, т.к. нет определенности, в каком направлении будет двигаться разработка. Здесь написание тестовых сценариев также неоправданно, поскольку функциональность будет меняться, а это потребует постоянной актуализации тест-кейсов.

Есть ряд случаев, когда сценарное тестирование необходимо, а исследовательское используется в качестве дополнения.

Приложение стандартизованное

Приложения, работающие по стандартам и гостам, для которых малейшее отклонение может быть критичным. Это банковские продукты, сложные системы CRM, ERP, любые программы с высокими финансовыми или иными рисками.

Интеграционное тестирование

Обычно интеграционное тестирование проводится для проверки взаимодействия внутренних компонентов приложений. Эта работа хорошо покрыта документацией и часто автоматизируется.

Тестовые сценарии на аутсорсе

Контролировать поставленную задачу и процент ее выполнения проще по формализованным сценариям.

Длительный проект

Тестировщики могут быть подключены к проекту на время определенной фазы, а после, пока разработчики реализовывают новый функционал, заниматься другими проектами. Если долго не тестировать конкретную функциональность, то ее специфика забывается, поэтому все должно быть описано в тестах.

Туры Дж. Уиттакера - подходы к исследовательскому тестированию

Один из подходов к исследовательскому тестированию описывает Джеймс Уиттакер в книге “Exploratory software testing”. Он сравнивает тестировщика-исследователя с туристом, который оказался в незнакомом городе. В зависимости от того, какую цель преследует турист-тестировщик, приводятся туры:

- Туры в бизнес-районах
- Туры в исторических районах
- Развлекательные туры
- Туристические достопримечательности
- Тур по отелям
- Туры по значным местам

и так далее.

Рассмотрим примеры бизнес-туров. Название говорит само за себя: тестировщик исследуют основные бизнес-процессы, реализованные в приложении. Иными словами - то, ради чего клиент покупает программу. Также это функции, описанные в маркетинговой или иной документации по продукту.

Тур по путеводителю.

Цель - тестирование пользовательской документации. Тестировщик берет руководство пользователя и последовательно выполняет все, что там написано. Это открывает ошибки как в функциональности, так и в пользовательской документации.

Денежный тур.

Цель денежного тура - протестировать все то, за что клиенты готовы заплатить, т.е. платные возможности программы. С другой стороны, в денежном туре необходимо проверять маркетинговые артефакты: демо-режим, рекламные ролики, соответствие маркетинговой документации реальности.

Интеллектуальный тур.

Цель тура - “озадачить” приложение, чтобы заставить его работать по максимуму. Например, это может быть формирование очень сложного отчета, оформление очень большого заказа, допущение максимального количества ошибок при выполнении операции и т.д.

Дж. Уиттакер приводит очень большое количество различных туров, с которыми можно ознакомиться, прочитав книгу “Exploratory software testing”

Отчет об исследовательском тестировании

В отчет об исследовательском тестировании входит:

1. Список протестированных разделов или модулей продукта. С его помощью в дальнейшем можно оценить тестовое покрытие, и необходимость дополнительных исследований.
2. Список дефектов: найденных вообще или только самых критических – в зависимости от того, для кого и на какой стадии тестирования делается отчет.
3. Найденные проблемы, вопросы, наблюдения.
4. Риски. Важно рассказать о том, что не было протестировано и в связи чем это произошло – функциональность не входила в объем работ, не работал сервер, не было подходящих тестовых данных и так далее.
5. Краткий вывод по результатам тестирования в зависимости от изначальной цели тестирования. Например, можно ли передавать продукт заказчику для ознакомления.

Свободное (интуитивное, ad-hoc) тестирование

Часто исследовательское тестирование приравнивают к свободному тестированию (ad-hoc testing), но это неверно. Свободное тестирование выполняется без подготовки, без определения ожидаемых результатов, проектирования тестовых сценариев. Это неформальное, импровизационное тестирование. Оно не требует никакой документации, планирования, процессов, которых следует придерживаться при выполнении тестирования.

Чаще всего такое тестирование выполняется, когда владелец продукта не обладает конкретными целями, проектной документацией и ранее поставленными задачами. При этом тестировщик полагается на свое общее представление о продукте, сравнение с похожими продуктами, собственный опыт. При тестировании ad-hoc имеет смысл владеть общей информацией о продукте, особенно если проект сложный. Поэтому нужно хорошее представление о целях проекта, его назначении и основных функциях, и возможностях. А дальше уже можно приступать к ad-hoc тестированию.

Виды свободного тестирования

Buddy testing – процесс, когда 2 человека, как правило разработчик и тестировщик, работают параллельно и находят дефекты в одном и том же модуле тестируемого продукта. Такой вид тестирования помогает тестировщику выполнять необходимые проверки, а разработчику исправлять множество дефектов на ранних этапах.

Pair testing – процесс, когда 2 тестировщика проверяют один модуль и помогают друг другу. К примеру, один может искать дефекты, а второй их документировать. Таким образом, у одного тестировщика будет функция, скажем так, обнаружителя, у другого – описателя.

Monkey testing – произвольное тестирование продукта с целью как можно быстрее, используя различные вариации входных данных, нарушить работу программы или вызвать ее остановку (простыми словами – сломать).

Однако, свободное тестирование не позволяет адекватно оценить тестовое покрытие, не фиксирует результаты и зачастую может оказаться неэффективным. Поэтому лучше придерживаться подходов исследовательского или сценарного тестирования.

Глоссарий

Исследовательское тестирование – это одновременное изучение программы, проектирование и выполнение тестов. Такой подход противоположен сценарному, когда список проверок составлен до прохождения тест-кейсов

Свободное тестирование - тестирование без подготовки, без определения ожидаемых результатов, проектирования тестовых сценариев. Это неформальное, импровизационное тестирование

Buddy testing – процесс, когда 2 человека, как правило разработчик и тестировщик, работают параллельно и находят дефекты в одном и том же модуле тестируемого продукта.

Pair testing – процесс, когда 2 тестировщика проверяют один модуль и помогают друг другу.

Monkey testing – произвольное тестирование продукта с целью как можно быстрее, используя различные вариации входных данных, нарушить работу программы или вызвать ее остановку

Домашнее задание

1. Изучите информацию об исследовательских турах Дж. Уиттакера в дополнительных материалах. Выберите любой из исследовательских туров.

2. Выберите любой сайт, которым вы часто пользуетесь. Проведите тестирование в соответствии с правилами выбранного тура. Составьте отчет о тестировании, указав, какой тур вы использовали.
3. Тест для самопроверки <https://coreapp.ai/app/player/lesson/61cd9f1ef98ef592e30202ba>
(сдавать не нужно)

Дополнительные материалы

1. <https://www.software-testing.ru/library/testing/general-testing/2416-testing-without-requirements>
2. [Исследовательское тестирование: пустая трата времени или мощный инструмент?](#)
3. [Туры в исследовательском тестировании. Личный перевод из книги Д. Виттакера «Исследовательское тестирование ПО»](#)
4. [Переводы туров для исследовательского тестирования](#)
5. [Исследовательское тестирование и исследовательские туры Виттакера](#)

Используемые источники

1. <https://habr.com/ru/post/148479/>
2. <https://habr.com/ru/company/redmadrobot/blog/280618/>
3. J. Whittaker. Exploratory software testing