

6

Исследовательское тестирование

6.1. Введение

До сих пор основное внимание уделялось автоматизированным тестам. Но в этой главе мы будем рассматривать ручное тестирование и в особенности тестирование, которое называют исследовательским. В разделе 6.2 рассказывается, где и с какой целью используются исследовательские тесты. В разделе 6.3 демонстрируются конкретные процедуры выполнения таких тестов. Исчерпывающее введение в эту тему также можно найти в книге «Explore It!» [1], где концепции этого вида тестирования объясняются более подробно.

6.1.1. Исследовательское тестирование: пример

В компании «Big Money Online Commerce Inc.», с которой мы познакомились в разделе П.2, накоплен большой опыт ручного тестирования — каждая новая версия подвергалась всестороннему тестированию вручную. Однако из-за автоматизации приемочных испытаний фокус ручных тестов сместился: теперь главной их целью является тщательная проверка новых функциональных возможностей. В этом контексте наиболее важным является понимание работы программного обеспечения с предметной точки зрения. Это — характерная особенность исследовательского тестирования, осуществляемого вручную. Исследовательское тестирование проводится также с целью оценить и усовершенствовать отдельные характеристики программного обеспечения, такие как удобство использования, защищенность, реакция на крайние ситуации.

6.2. Цель исследовательского тестирования

Главной целью методологии непрерывного развертывания является автоматизация тестов. Ручные тесты обходятся недешево, особенно когда

тестировать приходится часто. Каждый тест должен выполняться и анализироваться человеком. Автоматизированные тесты, с другой стороны, должны быть автоматизированы только один раз. Последующие запуски автоматизированных тестов практически ничего не стоят. Благодаря методологии непрерывного развертывания приложения часто развертываются и, соответственно, часто тестируются. Поэтому тестирование приходится автоматизировать, иначе его стоимость будет слишком высока.

6.2.1. Иногда ручное тестирование оказывается предпочтительнее

Как бы то ни было, в некоторых случаях ручное тестирование может оказаться предпочтительнее: когда вводятся новые возможности, специалисты в прикладной области должны сначала проверить их вручную. Чтобы суметь оценить новые возможности, тестировщики должны знать и понимать предметный контекст. Основой таких тестов могут служить требования или пользовательские истории. Кроме того, ручные тесты дают возможность еще раз критически изучить требования и выявить возможные ошибки. Естественно, этого нельзя достичь с использованием автоматизированных тестов, потому что они должны основываться на определенных требованиях.

Исследовательское тестирование основывается на том факте, что тестировщики часто обнаруживают проблемы, только когда отклоняются от сценария и самостоятельно изучают и исследуют приложение. При этом они руководствуются собственным опытом и знанием типовых уязвимостей приложений. Очевидно, что тесты этого вида нельзя автоматизировать — их может выполнить только опытный тестировщик.

Исследовательское тестирование может оказаться превосходным инструментом детального изучения особенностей, реализованных с ошибками, или где высока вероятность появления ошибок. Когда, например, конкретная функция или модуль чаще других вызывают проблемы в рабочем окружении, в ходе исследовательского тестирования можно особенно тщательно проанализировать эту область, чтобы выявить дополнительные ошибки.

6.2.2. Тестирование заказчиком

Иногда заказчик может пожелать протестировать новую версию, прежде чем принять ее. И снова в этом случае главное внимание уделяется новым возможностям. Если заказчик также тестирует давно реализованные функ-

ции, разработчики должны использовать это как возможность для повышения доверия к автоматизации. Например, разработчик может показать, как функции, которые заказчик оценивает вручную, тестируются в автоматическом режиме. Кроме того, разработчик может реализовать для заказчика приемочные тесты новых особенностей для заказчика, выполняющиеся автоматически.

Регрессионные тесты никогда не должны выполняться вручную, потому что это ведет к неоправданному увеличению затрат, когда требуется тестировать каждый релиз. С другой стороны, не всякое требование должно приводить к созданию автоматизированного теста. Вполне достаточно, если автоматизированные тесты дают разработчикам определенную уверенность в нормальной работоспособности новой версии. Для этого приложение должно быть протестировано до такой степени, что все случаи использования в тестах выполняются без ошибок.

6.2.3. Ручное тестирование нефункциональных требований

Исследовательскому тестированию обычно подвергаются новые функциональные возможности. Однако ручные тесты могут быть предпочтительнее автоматизированных, когда дело касается нефункциональных требований.

- *Тестирование удобства в использовании* позволяет оценить простоту пользования приложением. Очень легко выполняется человеком, но с большим трудом поддается автоматизации.
- То же верно для *оценки внешнего вида* и соответствия требованиям к оформлению. Часто человек способен сразу обнаружить ошибки, но автоматизировать этот процесс чрезвычайно сложно.
- Наконец, выявление *проблем безопасности* зачастую проще осуществить вручную, а не в виде автоматизированного процесса. В этой области часто применяются приемы обзора программного кода или проверки возможности вторжения, которые тоже весьма сложно автоматизировать.

6.3. Как это сделать?

Исследовательское тестирование в первую очередь направлено на проверку прикладной функциональности. То есть тестировщики должны оценить

соответствие приложения выдвинутым требованиям. Поддержку автоматизации рутинных операций при проведении тестирования могут оказать такие инструменты, как Selenium (см. главу 4, «Приемочные тесты»). Они записывают последовательность взаимодействий с веб-интерфейсом приложения и воспроизводят их снова и снова.

Вслед за этим тестировщики выполняют операции с приложением вручную. Такие сценарии автоматизации могут также послужить основой для более поздней, полной автоматизации тестирования.

6.3.1. Руководство по проведению тестирования

Планы тестирования, описывающие, что и как тестировать, безусловно, полезны сами по себе. Однако они также помогают определить момент, когда тесты достигнут определенного уровня зрелости и сложности и их автоматизация сможет дать определенные выгоды. Поэтому в исследовательском тестировании вместо планов используются уставы (см. раздел 6.3.6).

6.3.2. Автоматизированное окружение

Окружение, созданное автоматически, послужит хорошей основой для исследовательского тестирования. В этом окружении необходимо установить программное обеспечение и соответствующие тесты.

Исследовательскому тестированию должны подвергаться только версии, успешно преодолевшие все другие стадии — то есть тестирование, выполняющееся на этапе сохранения изменений, автоматизированные приемочные испытания и тестирование пропускной способности. Иначе усилия, потраченные на ручное тестирование, окажутся затраченными зря на проверку версии, имеющей сомнительное качество.

6.3.3. Демонстрационные примеры как основа

Демонстрационные примеры служат надежной основой для исследовательских тестов. В проектах, развиваемых с применением методик гибкой разработки, в конце итерации клиенту часто демонстрируются примеры, показывающие, как работают вновь добавленные возможности и какие выгоды они дают. Именно эти возможности также должны подвергаться исследовательскому тестированию. Поэтому одним из результатов иссле-

довательских испытаний может быть успешный прогон через такой демонстрационный пример.

6.3.4. Пример: приложение электронной коммерции

Предположим, что в приложении электронной коммерции была реализована возможность создания срочных заказов. В ходе исследовательского тестирования эксперт создает такой срочный заказ и проверяет, насколько правильно он обрабатывается. В дополнение к этому очевидному случаю тестировщик выясняет, что произойдет, если клиент попытается оформить срочный заказ на товар, недоступный в данный момент, или срок доставки которого превышает время выполнения срочного заказа. Аналогично может быть проверена функция отмены срочного заказа. При этом особое внимание уделяется процедурам, подверженным ошибкам, например обновлению состояния заказа. Благодаря знаниям бизнес-процессов, особенностей работы текущей версии системы и областей, где возможны ошибки, ручное тестирование может быть весьма эффективным. При этом, конечно, создается хорошая основа для автоматизации тестирования, которое впоследствии защитит приложение от регрессий.

Кроме того, в ходе исследовательского тестирования проверяется удобство использования и внешнее оформление с применением методов, устойчивых в этой области. Здесь пригодятся опросы пользователей, а также запись взаимодействий с программным обеспечением с последующим анализом и оценкой.

Кроме перечисленного во время ручного тестирования может проверяться защищенность приложения — например, проверка возможности вторжения или обзор кода в поисках проблем с безопасностью. Аналогично вручную проверяются пропускная способность или производительность. Однако в последнем случае обычно требуется использовать генератор нагрузки, поэтому такие тесты трудно реализовать без автоматизации.

6.3.5. Бета-тестирование

Еще одна разновидность исследовательских тестов — бета-тесты. В этом случае программное обеспечение передается ограниченному кругу пользователей, которые, как предполагается, опробуют его и сообщат об обнаруженных ошибках и предложениях по улучшению. В некоторых случаях это

позволяет одновременно проверить, позволяет ли новая версия увеличить объем продаж.

6.3.6. Сеансовые тесты

Для структурирования исследовательского тестирования удобно использовать метод сеансовых тестов [2]. В этом случае тесты делятся на сеансы. Для каждого сеанса определяется его цель — виды ошибок или проблем, которые должны быть проверены. Сеанс можно сравнить с экспедицией в приложение. Цель экспедиции определяется уставом. Устав географической экспедиции может, например, определить такую цель, как исследование определенного ландшафта. Устав исследовательского тестирования аналогично определяет цели каждого сеанса. Устав может иметь следующую форму:

Исследовать ... (цель)
с помощью ... (инструмент),
чтобы определить ... (информация).

Эти три части определяют следующие аспекты.

- *Цель* — часть приложения, которая подвергается тестированию, — то есть определенная функция, требование или модуль.
- *Инструмент* определяет средства, используемые для тестирования. К инструментам можно отнести определенные наборы данных или программные средства.
- Наконец, *информация* определяет результат тестирования. Результатами могут быть выводы о защищенности, быстродействии или надежности. То есть во время исследовательского тестирования обязательно должны проверяться конкретные предметные требования и ошибки.

В контексте приложения регистрации клиента можно было бы определить такой устав исследовательского тестирования:

Исследовать приложение регистрации клиента
с помощью соответствующего набора данных,
чтобы убедиться в правильной интернационализации.

В ходе тестирования в соответствии с этим уставом проверяется возможность использования букв национальных алфавитов и других наборов сим-

волов, например корейских, японских или китайских. Возможно ли, что деление на имя и фамилию подходит не для всех стран? И какие символы фактически допустимы в адресах электронной почты?

Ниже приводится пример совершенно другого устава:

Исследовать приложение регистрации клиента
с помощью атак OWASP,
чтобы выявить пробелы в системе защиты.

OWASP¹ [3] — это коллекция наиболее известных уязвимостей в веб-приложениях. В данном случае целью исследовательского тестирования является оценка защищенности — соответствие нефункциональному требованию.

Оба устава имеют общее свойство, формулируемое универсальным способом. Ручные тесты необязательно должны быть полностью предопределены; допустимо сформулировать только цель и используемые методы. Тестировщики сами определяют конкретную реализацию сеанса.

Уставы могут определяться на основе требований или преследовать цель более тщательного исследования определенных рисков — например, выявление пробелов в системе защиты на ранних этапах.

Формальным результатом исследовательского тестирования является отчет о проведенном сеансе, содержащий, кроме устава, сведения о том, как выполнялось тестирование, какие ошибки были обнаружены и как долго длился сеанс. В заключение дается ретроспективный обзор сеанса. Сеанс длится всего несколько часов. Соответственно, ручное тестирование можно разбить на несколько разных сеансов с определенными целями и таким способом структурировать его.

Тестировщики имеют определенную свободу в выборе инструментов. В дополнение к использованию инструментов автоматизации они могут исследовать файлы журналов или параметры работы системы. Для этого могут очень пригодиться инструменты для мониторинга и анализа файлов журналов, представленные в этой книге (глава 8, «Эксплуатация»). Кроме того, исследовательские тесты могут опираться на пользовательские и программные интерфейсы или на веб-службы. Естественно, все это требует наличия технических знаний у тестировщиков. Такие исследования могут

¹ Open Web Application Security Project — открытый проект обеспечения безопасности веб-приложений. — *Примеч. пер.*

проводиться только специалистами, знакомыми с необходимыми инструментами. В особых случаях от тестировщиков может даже потребоваться умение писать собственное программное обеспечение.

В течение сеанса тестировщики исследуют программное обеспечение с применением разных приемов.

- Приложения имеют *переменные*. В зависимости от типа они могут хранить разные значения. Поэтому имеет смысл проверить разные вариации — например, использование китайских символов в имени пользователя.
- Порядок *взаимодействий и процедур* может изменяться. Это позволяет исследовать, что случится, если некоторая процедура выполняется пользователем не так, как предполагалось изначально.
- Кроме того, тестировщики могут исследовать *сущности и отношения между сущностями*. В ходе этого тестирования можно попробовать создать недопустимую сущность.
- Многие приложения имеют определенный набор состояний и определяют порядок переходов между ними. Здесь тестировщики проверяют, насколько разумно реализованы состояния или возникновение неожиданных состояний.

Эти приемы позволяют тестировщикам различать уставы и исследовать потенциальные слабости программного обеспечения.

ПРОБУЙТЕ И ЭКСПЕРИМЕНТИРУЙТЕ

- ▲ Реализуйте один из уставов, упомянутых выше, для тестирования приложения регистрации клиента.
- ▲ Какие другие уставы вы могли бы предложить для тестирования приложения регистрации клиента?

Взгляните на любой из своих проектов.

- ▲ Выберите определенную функцию. Какой устав для ее тестирования вы могли бы предложить?
- ▲ Какие нефункциональные требования (например, защищенность или производительность) являются особенно важными для приложения? Определите устав для их тестирования.

6.4. В заключение

Для большей ясности отметим еще раз: в проекте, который развивается с применением методологии непрерывного развертывания, не должно быть слишком много тестов, выполняемых вручную. Непрерывное развертывание предполагает частое развертывание и тестирование; на самом деле — очень частое. Поэтому тесты почти всегда имеет смысл автоматизировать. Однако не бывает правил без исключений: исследовательское тестирование новых возможностей в действительности возможно только вручную. Такое тестирование позволяет получить более полное представление об уязвимых аспектах приложения и реализовать автоматизированное тестирование для защиты от регрессий. То же верно для тестирования удобства в использовании и оформления, а также других нефункциональных требований, таких как проверка возможности вторжения. Исследовательское тестирование новых возможностей должно в конечном итоге приводить к созданию автоматизированных тестов. Кроме того, исследовательское тестирование может послужить хорошим способом более тщательного изучения некоторых аспектов приложения. В конце концов, автоматизированное тестирование не способно выявить абсолютно все ошибки.

Ссылки

1. Elisabeth Hendrickson: *Explore It! — Reduce Risk and Increase Confidence with Exploratory Testing*, Pragmatic Bookshelf, 2013, ISBN 978-1-93778-502-4.
2. <http://www.satisfice.com/sbtm/>
3. https://www.owasp.org/index.php/Main_Page