

Эвристики функционального тестирования: системный взгляд

Автор: Джорис Меерц (Joris Meerts), Testing References

Оригинал статьи: http://www.testingreferences.com/docs/Functional_Testing_Heuristics.pdf

Перевод: Ольга Алифанова, <http://software-testing.ru>

Использование техник тестирования, основанных на спецификации, для покрытия путей через программу или функцию – это очень заманчивая для функционального тестирования идея. Не менее заманчиво предположить, что раз эти пути или комбинации покрыты – функциональное тестирование более или менее завершено. Моя цель – показать при помощи описанных ниже эвристик, что функциональное тестирование может – и, возможно, должно – смотреть на вещи шире, учитывая не только то, что явно прописано в требованиях или дизайн-макете. Я уверен, что при помощи этих эвристик и точек зрения можно выявить приличное количество функциональных аспектов системы.

Список основан на моем личном опыте тестирования программных продуктов. Я также благодарен Джеймсу Баху за эвристику SFDPOT, и Элизабет Хендриксон, Джеймсу Линдси и Дейлу Эмери, как создателям чит-листа эвристик тестирования.

Эвристика	Определение	Описание
Последовательность	Упорядоченный набор	Функции могут выполняться последовательно. Варьируйте последовательность их выполнения, в том числе в ситуациях, когда выполнять функции в этой последовательности вроде бы нелогично.
Совпадение	Совмещенность событий, явлений и т. д.	Сделайте так, чтобы что-то происходило одновременно. К примеру, попытайтесь купить товар в интернет-магазине, и в то же самое время выставить его статус "недоступным". Для этого понадобятся два пользователя, работающие с одними и теми же данными.
Слияние	Объединение двух различных объектов в одно целое	Думайте о системе, как о наборе потоков, которые расходятся и встречаются вновь. В точках схождения потоков происходят интересные вещи!
Синхронизация	Процесс приведения к одному значению одного или нескольких параметров разных объектов	В крупных системах информация может храниться в нескольких местах и должна синхронизироваться. Синхронизация может проводиться партиями, по запросу, или онлайн. Протестируйте, что происходит, когда информация синхронизирована/не синхронизирована.
Делиться	Передавать, сообщать кому-либо что-либо; взаимно обмениваться чем-либо.	Функции могут делиться данными или совместно использовать железо. Протестируйте, что происходит, когда функции запрашивают или манипулируют одним и тем же одновременно.

Эвристика	Определение	Описание
Взаимодействие	Воздействие различных предметов, явлений и т. п. друг на друга, обуславливающее их изменение.	Многие функции взаимодействуют – передают друг другу результаты, ждут друг друга, вызываются друг другом. Изучите, каким образом они взаимодействуют, и протестируйте это взаимодействие. Функции зачастую имеют интерфейсы и взаимодействуют еще и с окружающим миром.
Непрерывность	Неразрывность, постоянство.	Непрерывность часто считается самоочевидной для функционирования системы, но в реальной жизни непрерывная работа гарантируется не всегда. Прервите работу системы в интересный момент и посмотрите, что будет.
Иерархия	Расположение частей или элементов целого в порядке от высшего к низшему или от низшего к высшему.	Не все функции одинаковы – некоторые могут иметь преимущество перед прочими, а некоторые – выполняться в соответствии со строгой иерархией. Используйте эту иерархию или поломайте ее, чтобы выявить проблемы.
Приоритет	Преобладающее, первоочередное значение чего-либо.	Приоритеты в системах могут зависеть от множества вещей и постоянно меняться. Спросите себя, какие функции приоритетнее прочих в определенный момент, и используйте эту информацию для тестирования.
Зависимость	Связанность явлений, предопределяющая их существование или сосуществование; обусловленность.	В системах, как правило, множество зависимостей в той или иной форме. Некоторые из них упоминаются в этом документе. Зависимости можно использовать для поиска проблем – чем больше зависимостей, тем веселее!
Повторение	Совершение чего-либо еще раз	Запустите функцию или набор функций несколько раз – возможно, даже много раз подряд. Запустите одну и ту же функцию несколько раз, чуть варьируя запуск.
Цикл	Организация многократного исполнения набора инструкций	Системы и функции содержат циклы, а также – иногда – циклы внутри циклов. Попробуйте думать о функциях и системах в терминах циклов (обратной связи), как гласит теория системного мышления, и посмотрите, куда эти размышления вас заведут.
Параметр	Величина, характеризующая какие-либо свойства процесса, явления, системы	Поведение многих функций управляется внешними параметрами/установками. Выясните, как можно менять параметры функции, и как их меняет система.
Предпосылка	Предварительное условие.	Некоторые системы/функции требуют предпосылок для запуска, и эти предпосылки редко бывают явными. К примеру, веб-приложение может требовать определенного браузера, а система – определенных настроек, соединений или версий других установленных приложений. Попытка поиграть с предпосылками может дать интересные результаты.

Эвристика	Определение	Описание
Конфигурация	Взаимное расположение каких-либо предметов или элементов.	Многое в системах можно конфигурировать – куда больше, чем можно подумать на первый взгляд. Тестировщики обычно не лезут глубоко в конфигурации, но конфигурация может значить очень много в определенных случаях. Изучение конфигурации системы может дать вам полезную информацию.
Правило	Норма поведения, принятый кем-либо образ мыслей, действий.	В ряде систем бизнес-правила и функциональность до некоторой степени отделены друг от друга, и область между тем, что управляется правилами и тем, что управляется функцией, несколько размыта. Подумайте о приоритетах, предпосылках и параметрах, исследуйте эту область.
Кастомизация	Индивидуализация продукции под заказы конкретных потребителей.	Кастомизация функций начинается там, где пользователям присваиваются различные роли. Это всегда интересно! Пользователи к тому же частенько могут настраивать систему под собственные нужды, отходя довольно далеко от настроек по умолчанию.
Ограничение	Правило, установление, норма, определяющие какие-то границы, рамки, условия.	Ограничения – к примеру, границы значений – встречаются в системах и функциях повсеместно. Интересно понаблюдать, что происходит на границах на разных уровнях системы. Границы также рассказывают нам о возможностях системы и о том, что произойдет, если мы превысим эти возможности. Можно подумать об ограничениях и в другом ключе: тест-стенд – это само по себе ограничение.
Ресурс	Запасы, средства, которые используются при необходимости.	Ресурс – это все то, что используется системой. Как правило, системы используют процессор и память. Что будет, если этот ресурс изменится или станет менее доступным?
Доступ	Возможность пользоваться чем-либо, получать что-либо	Функции могут иметь или требовать доступ к другим функциям, источникам данных, оборудованию и т. п. Возможности доступа функций могут изменяться.
Блокировка	Закрытие с целью обезопасить или сделать недоступным	Данные часто блокируются, чтобы запретить изменение их другими функциями. Как правило, блокировка через какое-то время снимается. Выясните, что именно делает эта блокировка.
Состояние	Положение, в котором кто-либо или что-либо находится.	Система всегда находится в определенном состоянии. Попробуйте выявить условия перехода в это состояние и узнать, как разные состояния влияют на функциональность.
История	Последовательный ход развития, изменения чего-либо	У систем, как правило, есть история использования. Тестировщики любят начинать с чистого листа и систем, у которых эта история нулевая. Пользователи всегда работают с уже повидавшей виды системой. Попробуйте разобраться, как наличие истории использования влияет на поведение системы.
Откат	Движение назад к предыдущему состоянию	Откаты, как правило, применяются, если что-то пошло не так. Они предназначены для "перезагрузки" системы в валидном, функциональном состоянии. Но происходит ли это на самом деле? Как система ведет себя после отката?
Восстановление	Возобновление, воссоздание в предыдущем или изначальном состоянии.	Восстановление похоже на откат, но не забывайте про функции и данные, которые подлежат восстановлению. Восстанавливается ли система целиком, или только ее части? Как насчет функций, которые не должны восстанавливаться?

Эвристика	Определение	Описание
Обновление	Оживление, замена на новое	Наиболее очевидная функция обновления – это кнопка "Обновить" или "Перезагрузить" в любом браузере. Тут возможны интересные наблюдения: в некоторых приложениях (не обязательно веб) есть функции, которые автоматически или неявно обновляют отображаемое содержимое. Как это обновление влияет на другие функции?
Клон	Копия оригинала	Клонирование – это концепция разработки ПО. Системы, функции и наборы данных могут клонироваться по разным причинам. Отдельный тест-стенд может считаться клоном всей системы или каких-либо ее частей. Копирование таблиц базы данных тоже может считаться клонированием. Тестируя клонов, держите в уме возможные расхождения с исходной системой или функцией.
Временное существование	Длжащееся, существующее только в течение ограниченного времени	Многое в системе не вечно, и не собирается вечно жить. К примеру, это временные таблицы, поддерживающие копирование информации, а также файлы и объекты. Тут можно проверить, скажем, то, очищаются ли временные объекты после использования.
След	Характерный отпечаток, возникший в результате чего-либо; признак, свидетельствующий о чём-либо.	Следы очень важны в функциональном тестировании. Их можно найти в логах и контрольных журналах. Используйте эти следы (трейсы) для анализа первопричин событий. Их также можно использовать для лучшего понимания внутренней кухни системы.
Партия	Определённое количество каких-л. предметов, товаров.	Многие системы обрабатывают данные сразу партиями. Тестирование этих партий и их функциональность – это отдельное искусство. Как функциональный тестировщик, вы должны понимать, что функциональность такой обработки может приводить к удивительным вещам. Зачастую бывает, что пакетная обработка началась, а тестировщик и не подозревает об этом. Также ее могут отключить при системном тестировании, что приводит к сюрпризам на более поздних этапах контроля качества.
Пустота	Ничем не заполненное пространство.	К примеру, в базе данных пустое поле и NULL – это не одно и то же. Функции могут работать с пустыми значениями совершенно иначе, нежели с NULL. Выясните, как функции обращаются с пустыми значениями.
Отсутствие	Неимение, недостаток, лишение	Данные, части данных, функции, приложения, ресурсы и что угодно другое могут временно отсутствовать в доступе. Выясните, как функции поведут себя, если что-то нужное отсутствует.
Обратная связь	Процесс получения информации о результатах действий, позволяющий регулировать поведение	Функции тем или иным образом предоставляют обратную связь, и она очень важна в тестировании – она позволяет верифицировать результаты теста. Будьте осторожны: обратная связь может врать! Когда вы используете ее, чтобы убедиться, что функциональность работает, используйте и другие источники информации.

Эвристика	Определение	Описание
Насыщение	Повышение концентрации чего-либо до момента, когда оно более не поглощается объектом.	Насыщение может вызывать функциональные ошибки на различных уровнях системы. Попробуйте оценить функциональность при больших объемах данных, открытии нескольких экранов интерфейса, использовании более чем одного пользователя, или любых других ситуациях, которые могут варьировать от нуля до множества.
Сорт	Разряд, категория предметов, объединённых общностью каких-либо внешних или внутренних черт	Алгоритмы сортировки присутствуют в системах повсеместно. Они могут быть описаны в спецификациях, а могут быть и упущены. Начните с понимания, что данные всегда предоставляются функции или операции в отсортированном виде. Попробуйте сломать порядок сортировки, перемешать его, сортировать по-разному, и посмотрите, что произойдет.
Масштаб	Степень значения, влияния и т.п.	Размер имеет значение. Масштаб обычно важен в нефункциональном тестировании – например, тестировании производительности, но изменение масштаба некоторых аспектов системы может помочь выявить и функциональные проблемы. К примеру, масштабирование количества записей в базе или установка системы на медленное/быстрое оборудование может выявить ограничения некоторых функций.
Повреждение	Нарушенная целостность/исправность состояния, испорченность	В системе после определенного периода использования появятся поврежденные данные или настройки. Как тестировщики, мы обычно тестируем с чистыми данными, и определить, при каких условиях они повреждаются, довольно тяжело. Однако мы можем протестировать, как будет работать система при поврежденных данных.
Целостность	Полнота, завершенность, отсутствие повреждений	Система – это модель, описывающая какой-то аспект реального мира. Исследуйте как объекты или сбор данных в системе развиваются вследствие выполненных над ними операций. Попробуйте выяснить, описывают ли эти объекты аспекты реального мира, и что может их повредить.
Запуск	Приведение в действие	Функция может быть вызвана множеством способов. Наиболее очевидные – при помощи интерфейса – это через нажатие клавиш клавиатуры или мыши. Функции могут вести себя по-разному в зависимости от способа запуска.
Время осуществления	Расположение на временной шкале	Время – это часть системы. В функциональном тестировании мы склонны забывать о том, что время выполнения операции может значительно повлиять на функционирование операции, а также на ее результат. Найдите функции, которые могут в той или иной степени зависеть от (к примеру) системного времени или внутренних часов. Попробуйте попутешествовать во времени.
Задержка	Приостановка, промедление	Задержки встречаются повсеместно, и в некоторых системах задержка ответа, к примеру, может быть организована специально. Незапланированная задержка, с другой стороны, тоже может быть результатом операций, выполняющихся системой. Используйте задержки для выяснения, что произойдет, если вы активируете другие функции во время этой задержки. Также попробуйте исследовать задержки с системной точки зрения, чтобы выявить, как меняется равновесность системы.