Это вопрос на рассуждение, чем на конкретный ответ. По крайней мере я бы сам отвечал примерно так: тестировать основываясь на собственном опыте (т.е. проводить эксплоратори тестирование - видишь приложение, придумываешь проверку и записываешь ее, проводишь и фиксируешь результат) работы приложений в общем и полагаясь на бизнес идею твоего конкретного приложения (что-то же оно должно делать: форма должна сохранять юзера, корзина должна позволять делать оплату), использовать в качестве образцов для поведения похожие приложения или конкурентов, задавать вопросы (стейкхолдерам, разрабам, всем кто обладает какой-то информацией, иначе вообще не узнать поведение которое ты фиксируешь баг или фича). Тут могут быть проблемы, когда начинают на ходу придумывать дичь т па - на проекте не у кого спросить, заказчики недоступны и т.д. То есть у собесера есть или конкретный ответ,который тебе надо угадать, хоть это и неправильный подход, но кого это волнует, или правильного ответа не существует вообще и что бы ты не ответил его это не устроит. Я бы и отвечал сам и ждал ответа так, как я описал, потому что в реальной жизни не может быть, чтобы не было у кого спросить, что ожидается например от корзины в случае если покупатель пытается ввести номер мастеркард (должны мы поддерживать этот тип карты или нет), в тоже время ты и сам прекрасно понимаешь, что нужно проверить и без требований поведение корзины если клиент вводит просроченную карту и скорее всего ожидаемым результатом будет ошибка. Но мы не можем предполагать, что какое-то поведение правильное или неправильное, потому что мы не предполагаем, а проверяем соответствует приложение требованиям заказчика и ожиданиям пользователя. Ожидания мы ещё можем предусмотреть основываясь на опыте или работе аналогичных приложений или так называемой общей практике, а требования заказчика надо уточнять.

Короче как повезет, если нарвешься на идиота собесера, то это вопрос из разряда не берущихся. Некоторые даже на вопрос кем вы себя видите через 5 лет могут ожидать "правильного" ответа

Когда-то давно я встретилась с потенциальным клиентом, чтобы обсудить тестирование его проекта. Вскоре после начала нашей встречи он, однако, осознал, что тестировать его проект нельзя, потому что полностью отсутствуют формальные требования. Короче говоря, они ничего не записывали. Он скептично отнесся к моему заявлению, что письменно зафиксированные требования необязательны для тестирования. «Но как же вы будете тестировать без требований? Как вы узнаете, баг это или не баг?» - «Спрошу».

Я пояснила свой ответ, объяснив, что зачастую я открываю для себя требования, не читая документацию по ним. Я, помнится, сказала, что он прав – я вполне могу не понять, баг это или не баг – поэтому я буду задавать кучу вопросов и обсуждать все несостыковки с проектной командой.

По пути домой я размышляла над нашим разговором и тем, как мне, как тестировщику, справиться с ситуацией, когда требования неясны или не записаны формально. Эта статья описывает процесс, который я использовала, чтобы выкопать и организовать информацию, нужную мне для тестирования, вне зависимости от доступности ее источников.

Что же такое «требование», и почему тестировщикам оно необходимо?

Оксфордский словарь определяет требование как «нечто, что необходимо или желательно». Отметим, что здесь не описано, как высказывается эта нужда или желательность. В разработке ПО «требования» традиционно считаются статичной письменной документацией. Эта документация описывает нужды заказчика и превращает их в описание того, как должна работать программа. Однако ПО по своей натуре штука сложная, а нужды заказчиков – вещь переменчивая. Поэтому заранее предсказать требования и полностью их зафиксировать в Microsoft Office очень сложно! Более динамичные способы хранения требований – в головах разработчиков, бизнес-аналитиков и других заинтересованных в продукте лиц.

Как тестировщик, я использую любые типы требований, которые помогают мне понимать, что делает продукт, что он должен делать, а чего делать не должен. Я очищаю информацию, строя ментальную модель продукта. Ментальная модель – это сеть идей, формирующихся исходя из опыта, осознанного или нет. Мы используем их, чтобы объяснить себе события, причины, следствия, и предсказывать будущие события. Эти модели не идеальны, неполны, и очень личные по своей сути (у каждого будет своя собственная).

**Построение ментальной модели**

Построение ментальной модели продукта – сам по себе очень беспорядочный процесс. Однако я попыталась упростить его, сведя к следующим пяти видам деятельности:

1. Исследование имеющихся источников.
2. Охота на другие источники.
3. Парная работа и мозговой штурм.
4. Прояснения и фильтрация.
5. Сопоставление.

Несмотря на то, что я их пронумеровала, эти вещи необязательно происходят последовательно. Я начинаю процесс в различных местах, множество вещей происходит параллельно, и иногда я делаю не все вышеперечисленное!

**1. Исследование имеющихся источников.**

Эта деятельность включает использование любой легкодоступной информации. В большинстве случаев вам доступен хотя бы один из следующих источников: продукт, документ с требованиями, живой человек, с которым можно пообщаться.

Как только мне достается продукт, я ныряю в него и смотрю, а что он вообще делает. Зачастую, если продукт сложен или перегружен информацией, разобраться в нем сложно, и это очень раздражает. Джеймс Бах рекомендует стратегию «быстрого нырка» - используйте продукт в течение получаса, позвольте себе почувствовать непонимание или раздражение, затем оставьте продукт в покое и займитесь чем-нибудь еще. Проделав это несколько раз, вы получите общее представление о продукте.

Документированные требования: с ними стратегия быстрого нырка тоже работает. Я читаю требования краткими урывками, узнаю что-то о продукте, откладываю их, а затем начинаю читать их снова. Я исследую их, чтобы выяснить, о чем авторы требований забыли подумать, и ищу информацию, которая неполна или не имеет смысла. Я задаю вопросы, представляя граничные ситуации и неожиданные сценарии и спрашивая себя, что может или должно произойти.

Люди зачастую наиболее важный и актуальный источник информации о продукте. Собирая требования, я поступаю, как бизнес-аналитик, очищая информацию, полученную от разных значимых лиц, о работе продукта.

**2. Охота на другие источники.**

Иногда люди, документированные требования и сам продукт неясны или просто недоступны. В этой ситуации я активно охочусь на другие источники или оракулы. Креативность тут большое подспорье, и я стараюсь выходить из своей зоны комфорта. К примеру, в проекте могут быть заинтересованные лица, с которыми можно пообщаться, но на первый взгляд это неочевидно. Иногда я могу сравнить наш продукт с похожими. Майкл Болтон предлагает полезную мнемонику для поиска подходящих оракулов – HICCUPPS:

H – History (история): как работали предыдущие версии?

I – Image (имидж) – какой имидж компания старается донести до пользователей?

C – Comparable Products (сравнимые продукты) – как работают похожие продукты?

C – Claims (заявления) – что говорят о продукте важные лица?

U – Users’ expectations (ожидания пользователей) – хотят ли пользователи такого поведения от этой фичи?

P – The Product itself (сам продукт) – соответствует ли поведение продукта другим его функциям?

P – Purpose (цель) – соответствует ли поведение фичи ее предназначению?

S – Statutes (нормы) – соответствует ли продукт законодательству и другим источникам права?

Охотясь на другие источники, я смотрю на всю картину целиком и стараюсь заполнить пустоты в своей ментальной модели.

Я смотрю на продукт, выбирая максимум возможных углов зрения. Расширение масштабов работы помогает мне держать себя в тонусе и пытаться оспорить допущения, которые существовали в моей голове.

**3. Парная работа и мозговой штурм.**

Другой способ оспорить свои допущения – это совместная работа и мозговой штурм с другими тестировщиками и разработчиками. К примеру, разработчик и я можем нарисовать диаграмму того, как что-то работает, а потом сравнить результаты. Мы также можем устроить штурм для генерации идей для тестирования в группе, или совместно заняться исследовательским тестированием. Зачастую мы собираемся, вооруженные, как нам кажется, одной и той же информацией. Теоретически мы знаем одно и то же. Однако как часто бывает так, что мы абсолютно по-разному представляем себе то, что происходит! Дело в том, что мы приходим на эти встречи, принося свой личный, разный для всех опыт.

Очень важно создать безопасное окружение, в котором не страшно ошибиться. В этом случае люди будут делиться идеями, здраво спорить, выявлять слабости в своем понимании вопроса, не боясь быть осмеянными. Здесь следует признать, что иногда это раздражает, так как никто из нас не знает, чья ментальная модель больше похожа на «правду».

**4. Прояснения и фильтрация.**

Перед этим я пыталась собрать воедино максимум возможных различныхточек зрения и источников информации. В течение этого процесса моя ментальная модель росла. Я старалась быть гибкой и не выбивать свое понимание продукта в камне чересчур рано. 1

1. Если информация конфликтует друг с другом, я спрашиваю мнения владельца продукта или кого-либо другого с похожим уровнем ответственности.
2. Если продукт ведет себя странно, я глубоко тестирую его путем специфических контролируемых сценариев. Это помогает мне получить доказательства того, что происходит.

Моя цель – избавиться от неверных предположений, ложной информации или ложных оракулов, которые попали в мою модель.

**5. Сопоставление**

Сопоставление информации – неотъемлемая часть формирования, развития и цементирования ментальной модели. Обычно я могу удержать в голове всякие несложные мелочи. Однако иногда то, что на первый взгляд кажется простенькой мелочью, оказывается куда более сложной штукой (просто я пока об этом не знаю).

В какой-то момент информация должна где-то формализоваться. Сбор информации – сам по себе катарсис. Маркированные списки, диаграммы и прочие инструменты помогут вам обработать и организовать информацию так, чтобы она имела смысл. Аарон Ходдер пишет про майнд-карты, помогающие передавать сложную информацию простым методом. Мне они также кажутся полезными, так как позволяют взглянуть на проект с высоты птичьего полета. Иногда в таком формат проще увидеть пробелы в своей модели. Они также легко позволяют записать информацию, которая постоянно меняется в процессе тестирования.

**Заключение**

Итак, отсутствие письменных требований не остановит мое тестирование. Я исследую источники, которые есть под рукой, и поищу другие. Я поработаю в паре и устрою мозговые штурмы. Когда я буду переполнена информацией, я сделаю шаг назад, проясню все необходимое, и зафиксирую информацию письменно.

Роль тестировщика имеет много общего с ролью бизнес-аналитика. Мы занимаемся сбором информации, хотя то, где, когда и как мы это делаем, может различаться. Лучших практик тут нет. То, что сработало в одном проекте, может не выстрелить в другом.

В своей статье я разбила свой процесс построения ментальной модели на пять различных видов деятельности. В реальности грань между ними размыта, и я упростила свое описание как могла. Эти виды деятельности пересекаются, могут существовать параллельно, весь этот процесс – сплошной бардак, но и у сумасшествия есть методология!

Когда мне удалось убедить скептично настроенного клиента, что ему не требуется создавать кучу формальных документов, чтобы я начала тестиро