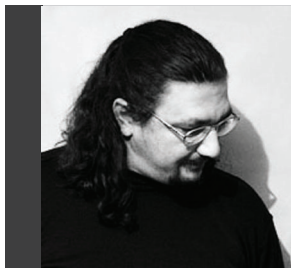


Как работать в условиях: «Вот тебе приложение, тестируй, как работает – сам поймешь»



Алексей Лупан

QA Trainer в компании 'SysIQ - Ukraine'.
Киев, Украина, Восточная Европа.

О себе: бородатый тестировщик.

Отлично, требований нет. Неужели так можно тестировать? Стрррашно!

Не, это не страшно. Это просто ещё один вызов тысячелетия одному-единственному тестировщику, который получил очень простое и, на взгляд окружающих, однозначное задание: «Тестер, найди мне тут все баги!». Ну дык! Ща найдём!

Не дрейфь, ты не шхуна без руля и ветрил

Проверь свой паспорт. Если ты не Баранцев, не Болтон, не Бах, и никогда не учился целенаправленно тестированию методом свободного поиска, то не рассчитывай на результативность своего «поиска».

Прежде всего, нужна системность. Помни, что «Упорство и прилежание лучше, нежели беспутство и гений» (поверь хотя бы писателю Эриху Марии Ремарку, не зря же у него двойное имя).

Сходу всегда воображается, что если нет ни плана, ни тест-кейсов, то надо сделать «исследовательское тестирование» — «эта када просто тыкаем мышкой, без плана и без тест-кейсов». Нуууу, брат, если после этого «када» план и тест-кейсы всё-таки будут написаны, тогда да — ты близко. Но поскольку обычно после «тыканья мышкой» ничего подобного не появляется, то рекомендую не пятнать светлое имя ad-hoc testing — именно так правильно называется это «када».

Поэтому соберись в узду — мы будем писать хоть и не каноничный, но все-таки тест-план, а тест-кейсы нам заменят птицы, камни и ветер. И идеи.

Ты же профессионал, едрить твой корешок!

Запроси на работу десять дней

Тут такое дело — если запросишь пять дней, то тебе дадут два дня. А запросишь десять — дадут неделю. А запросишь больше десяти — ничего не дадут.

Замечено, что если запросить

количество дней, которое не делится на два — будут вопросы про обоснование. А если оно делится — вопросов не будет.

Стань героем блокнота!

Открывай, ага, обычный блокнот «Notepad».

В нем запиши ПОСЛЕДОВАТЕЛЬНО все функциональные возможности приложения.

Что тут можно сделать? Залогиниться? Ок. Посмотреть каталог? Ок. Зохавать Вселенную? Пиши. Всё пиши, в ряд, столбиком.

Да, это долго. Да, это скучно. Да, это ой как дол го и скучно, мы уже знаем.

Зато потом всё это превратится в «документацию по проекту». Потом все будут ценить тебя именно за то, что «хотя бы что-то записано!». Потом тебе дадут пряник. Это всё будет потом. А сейчас пиши, не отвлекаясь.

Настоящие тестировщики никогда не работают успешно в отрыве от производственного коллектива.

Второй шаг...

Эй, не перепрыгивай! Закончи с записыванием функциональных возможностей приложения!

Таки перепрыгнул...

Ну, теперь надо все эти возможности организовать в стройную систему: это UserAccount, это WriteReview, это SetupProduct.

Фишка в том, что если всё УЖЕ в принципе расписано, то раскидывать строки по категориям тебе будет намного проще, чем если ты сперва продумаешь категории, а потом будешь их «заполнять». Поверь Ремарку с двойным именем, не изобретай уже изобретённое.

Если понял, то вернись на шаг назад и заверши расписывание функциональных возможностей.

Подобный список, как правило, растягивается на много-много-много страниц во всех этих наших Вордах. Это нормально. Пробежись

по программистам и прочим товарищам, которых заметишь, попроси растолковать те или иные возможности.

Если не знаешь, как разговаривать с незнакомыми людьми, то сразу иди в дверь и домой, ровным и смелым шагом. Настоящие тестировщики никогда не работают успешно в отрыве от производственного коллектива.

Третий шаг — расположить эти категории по важности. Самые важные — первые. Не верь, что «важные — все». Тут тебе поможет менеджер тестируемого ПО. Клянусь, некоторые функции, которые ты распишешь, ему будут внове... Особенно если речь идет о тестировании целого ееее-шопа интернет-магазина.

Четвёртый шаг семенит рядом с третьим — уже отсортированные функции надо расположить в два списка Critical features и Common features.

Поимей идеи!

Верю, что с каталогом покончено. Велком ту зз второй этап большой работы: теперь для каждого пунктика в твоём огромном чек-листе следует записать ВСЕ идеи о том, что и как можно протестировать в каждом из пунктиков.

Да, для каждого :)

Да, это долго. Да, это скучно.

Да, это ой как долго и скучно, мы уже знаем.

Но если у тебя не будет расписано, что можно тестировать, то как ты будешь соображать, что именно нужно протестировать?

Ведь ты уже знаешь, что тестировать следует в первую очередь воображением, а потом уже «тыкать мышкой в хладный труп приложения», ага? Или тебе всё ещё надо рассказывать, чем чревато «тестирование по-живому»?

Да, можно придумывать на ходу. А регрессионное тестирование как потом будешь проводить? Тоже «на ходу» будешь придумывать да вспоминать?

Итак, кратко пишем простые идеи. Форма логина? Ввести правильные значения, залогиниться. Ввести неправильные значения,

убедиться в том, что залогиниться невозможно. Что-то ещё тут можно проверить? Пиши.

В итоге ты получишь отличный чек-лист для тестирования.

Чек-лист отличается от тест-кейсов много чем. У него есть плюсы и минусы, но пока что он самый плюсовой вариант.

Перестань ныть и думать про «а вот если автоматизировать»

Робот за тебя ничего не сделает, и человека он никогда не заменит. Робот туп, хитёр только человек. Если тебя можно заменить роботом, то у меня для тебя плохие новости, брат...

Автоматизировать будешь позже, когда появится свободное время.

Сходу рекомендую проверить того, что находится в списке Common features, ведь Critical features лучше проверять руками и глазами.

Составь тест-план

Ну, как тест-план... Это будет план действий вообще. Помнишь про «Четвёртый

шаг семенит рядом с третьим»? Твой простой и изящный план тестирования будет таким:

- New features.
- Bug-fixes.
- Critical features.
- Common features.

Логика такая: сперва проверяем все новые функции или изменения в релизе. Знаешь ведь, что такое релиз? Это когда в офисе скрытая паника, и кофе уже нет.

Последовательно воспроизведи все тестовые ситуации

Последовательно и въедливо проверяем все Critical features.

Если останется время — проверим и Common features. Если не останется — ну, и черт с ними...

Пусть у тебя и не будет тест-кейсов — можно ориентироваться уже только по идеям.

Заноси все баги в баг-трекер сразу при обнаружении

Как, ты ещё не поставил сам себе рулезный баг-трекер Mantis?????????

Поэтому хватай всё, что дают, и убегай делать мир лучше.

Хватай деньги и убегай

Мы пришли сделать мир лучше, или заработать все возможные деньги?

Поэтому хватай всё, что дают, и убегай делать мир лучше.

Если что непонятно — обращайся в любое время, я тут тусуюсь, поблизости...

У хорошего тестера – сердце разработчика... в баночке на століке
/Автор не известен/

Требования не были плохими; их просто не так поняли
/Автор не известен/

Не сломал – значит, не старался! /Автор не известен/