

## Раздел 9. Гибкое управление дефектами

Лайза Криспин: Ответственность за качество продукта в agile несет вся команда

**Добрый день, Лайза. Спасибо, что смогли найти время и ответить на наши вопросы. Расскажите о себе.**

Добрый день. Свою карьеру я начинала в качестве разработчика/аналитика, а в начале 90-х годов заинтересовалась тестированием и с 2000 года работаю в agile-командах. Сегодня я занимаюсь тестированием баг-трекинговой системы в компании Pivotal Tracker. В мои обязанности входит тестирование UI и API в SaaS-продуктах, а также тестирование нашего продукта на iOS.

Я с удовольствием обмениваюсь профессиональным опытом с другими специалистами, часто выступаю на конференциях, публикую статьи, пишу книги. Совместно с Джанет Грегори, консультантом по внедрению гибких практик, мы выпустили две книги: «Гибкое тестирование. Практическое руководство для тестировщиков ПО и гибких команд» и «Еще больше agile-тестирования: исследуем всей командой» (*More Agile Testing: Learning Journeys for the Whole Team*).

**Сегодня во многих компаниях традиционный каскадный подход к разработке ПО сменяется практиками agile. Как этот процесс может повлиять на тестирование?**

Я считаю, что для разработки и долгосрочной поддержки высококлассного ПО вся команда должна чувствовать ответственность за тестирование. Нужно перестать отслеживать баги после написания кода и попытаться предотвратить их возникновение.

**Каковы главные особенности тестирования в agile-проектах?**

Идеальный agile-проект выглядит так. Вся команда осознает важность качества продукта. Разработчики принимают участие в тестировании. Разработка ведется через модульное и приемочное тестирование. Началу

тестирования и написания кода предшествует общее обсуждение продукта и ожидаемой функциональности.

Применяя методики TDD (разработка через тестирование), ATDD (разработка через приемочное тестирование), BDD (разработка на основе поведения), SBE (Specification by Example), agile-команды выигрывают от автоматизации регрессионного тестирования, высвобождая время для исследовательского тестирования и детального изучения продукта.

Для успешного применения этих методик в команде должно поощряться желание специалистов учиться и экспериментировать. Акцент должен ставиться на качество продукта, не на скорость работы.

### **Какими навыками должен обладать гибкий тестировщик?**

В моем представлении хороший тестировщик – это любознательный, эрудированный специалист с гибким мышлением, руководствующийся принципом «сделаю все, что может оказаться полезным». Безусловно, он должен обладать хорошей технической подготовкой и сильными коммуникативными навыками для плодотворного общения с заказчиками и технической командой.

### **Приведите примеры тестовых подходов в agile-проектах.**

В успешных agile-проектах разработка основана на тестах, ориентированных на заказчика. Тут оказываются полезными вышеупомянутые методики BDD, ATDD или SBE. На мой взгляд, у них есть нечто общее: с самого начала вы собираете команду и обсуждаете продукт и релизные требования, оговариваете ключевые моменты, строите walking skeleton и решаете, как будете его тестировать.

Советую пользоваться такими техниками как персоны (personas) и карты историй (story mapping) для выделения приоритетных требований. Каждая пользовательская история обсуждается с экспертами по продукту, тестировщиками, разработчиками, дизайнерами.

В процессе обсуждения оговариваются правила и пожелания поведения системы, записываются критерии приемки. Техника Мэтта Вайна example

mapping может оказаться весьма кстати на данном этапе. Затем разработчики создают приемочные и модульные тесты и по возможности выполняют исследовательское тестирование каждой истории.

Тестировщики, в свою очередь, исследуют функциональность системы и информируют команду о функциональных и нефункциональных атрибутах качества.

### **Какие инструменты применяются в гибком тестировании?**

Если вы имеете в виду фреймворки для автоматизации, библиотеки и драйвера, то сначала тестировщики должны решить, как будут выглядеть тесты, а лишь затем подобрать соответствующие инструменты.

Автоматизация тестирования важна тем, что позволяет освободить время для исследовательских тестов. Но внутрикомандное общение важнее любых инструментов.

Что касается автоматизации, то существует мнение, что автоматизация регрессионных тестов – обязательное условие. Но нужно мыслить шире. Прежде чем начать работу над продуктом, необходимо удостовериться, что команда сходится во мнении касательно продукта, релизных требований и пользовательских историй.

Составление вышеупомянутой карты историй может оказаться весьма удобной техникой. Для этого вам потребуются специалисты с навыками тестирования и бизнес-анализа. Важно проверять код до и после его запуска, чтобы быть уверенными в том, что он действительно решает какую-либо проблему заказчика.

### **Как можно завершить тестирование короткими итерациями?**

Повторюсь, что ответственность за тестирование должна быть возложена на всю команду. Ни одна пользовательская история не может быть закрыта без тестирования. Применение методик TDD и BDD/ATDD/SBE позволяет достичь этой цели. Что касается сроков, то тестирование не будет отставать от разработки, если вы сможете обеспечить тесное сотрудничество тестировщиков и разработчиков.

Зачастую agile-команды чрезмерно стараются угодить заказчику. Это проявляется в планировании слишком большого числа пользовательских историй в спринте и пренебрежении этапами тестирования. Чтобы избежать подобной проблемы, команды должны планировать меньший объем работы, но выполнять ее на максимально высоком уровне. Хороший код и прочная тестовая база позволят вам в будущем поставлять качественное ПО быстрее. Попробуйте ограничить число историй и сфокусируйтесь на завершении одной истории, прежде чем переходить к следующей.

### **Какими принципами должны руководствоваться agile-тестировщики?**

Мы с Джанет выделили 10 основных принципов agile-тестировщика:

- Предоставляйте регулярную обратную связь.
- Удовлетворяйте потребности заказчика.
- Общайтесь вживую.
- Не бойтесь пробовать.
- Ищите простой способ решения.
- Анализируйте возможные способы улучшения.
- Реагируйте на изменения.
- Научитесь управлять своим временем.
- В первую очередь, думайте о людях.

### **Без какой документации не обойтись в agile-проектах?**

Документация в agile должна создаваться, когда она действительно необходима. Манифест Agile гласит, что рабочее ПО превыше документации. Значит ли это, что документировать не нужно совсем? Нет. В процессе работы я пришла к выводу, что автоматизированные регрессионные тесты, созданные с использованием методик BDD/ATDD/SBE, являются хорошим источником информации о поведении системы.

Вам также может потребоваться дополнительная документация для пользователей, заказчика, контролирующих органов. Каждая команда вправе

сама решать, какая документация нужна, и обсуждать это с представителями бизнеса и аудиторами.

### **Что нужно для подготовки соответствующей документации?**

Рабочие приемочные тесты – отличный способ создания актуальной документации. Кроме того, вы можете генерировать документацию. К примеру, наша команда создает API документацию на основе примеров из реальных тестов.

Из своего опыта могу отметить, что большинство команд предпочитают нанять профессионального технического писателя для создания грамотной, исчерпывающей документации.

### **Что вы можете посоветовать agile-тестировщикам?**

Не переставайте учиться. Сегодня в вашем распоряжении множество онлайн-ресурсов и сообществ, где вы можете делиться своими мыслями, идеями, задавать вопросы и общаться на актуальные темы. При возможности посещайте конференции тестировщиков. Читайте, пробуйте, экспериментируйте!

### **Каким вы видите будущее agile?**

Я не стараюсь предвидеть будущее, но сейчас я вижу, что довольно трудно найти тестировщика с правильным отношением и подходом, который мог бы внести свой вклад в agile-проект.

Вместе с тем, все больше разработчиков начинают осознавать важность исследовательских тестов и методик BDD/ATDD/SBE. Некоторые команды предпочитают нанимать высококвалифицированных тестировщиков со стороны, чтобы те передали свои навыки разработчикам и задействовали в тестировании всю команду.

Это ни в коем случае не означает, что в будущем у нас не будет потребности в тестировщиках. Нам будут нужны такие тестировщики, которые смогут поделиться навыками тестирования со всей командой.

**Спасибо, Лайза, за то, что нашли время и поделились с нами своими идеями.**

## Как использовать Agile Manifesto в мобильном тестировании

Начало 21 века было ознаменовано рождением документа, который оказал большое влияние на разработку программного обеспечения – **Agile Manifesto** – Манифест гибкой методологии. Этот документ со списком идей подписали 17 разработчиков ПО в феврале 2001, однако лишь немногие решились использовать его на практике.

За 14 лет уровень использования гибких методологий превзошел уровень использования классических моделей жизненного цикла. Принципы Agile-манифеста повлияли на появление новых подходов к разработке ПО.

Гибкие методологии оказали влияние и на сферу обеспечения качества. Тестировщики стали универсальными солдатами, способными выполнять поставленные задачи, даже в условиях нехватки времени и информации.

Специфика мобильных разработок требует максимальной гибкости для создания по-настоящему качественного программного продукта.

**Две ключевые идеи Agile Manifesto, которые характерны для мобильных разработок:**

- работающий продукт важнее исчерпывающей документации;
- готовность к изменениям важнее следования первоначальному плану.

Почему Agile – это лучший подход к мобильному тестированию?

1. **Продукт постоянно изменяется, чтобы лучше отвечать потребностям пользователей**  
Ваше приложение должно превосходить конкурентов. Создатели мобильных приложений сражаются за пользователей, хотят, чтобы их продукты возглавляли рейтинги. Очевидно, что быстрая реакция на запросы рынка – один из ключей к успеху в мобильном мире.

Это приводит к постоянному изменению требований, необходимости быстрого тестирования и отсутствию качественной документации. Agile отлично подходит для таких случаев.

**В данной ситуации работают следующие принципы Agile:**

- удовлетворение потребностей клиента благодаря быстрой поставке ПО;

- возможность изменения требований даже на поздних стадиях разработки;
- частая поставка рабочего ПО.

**Эти принципы можно реализовать при помощи следующих инструментов:**

- наличие доступа к сборкам, отлаженный процесс поставки новых сборок;
- активное уточнение требований на ежедневных scrum-совещаниях.

Одна из особенностей мобильных разработок — возможность предоставления версии продукта для тестирования практически в любое время. В большинстве случаев, сборка приложения занимает не более 20 минут.

Отлаженный процесс поставки сборок на тестирование позволяет быстро получить проверенный продукт по требованию.

Возможность быстрого изменения и тестирования новой функциональности могут сыграть решающую роль в повышении конкурентоспособности продукта.

## **2. Способность тестировать на новых версиях операционных систем и новых устройствах**

Мобильные технологии развиваются значительно быстрее, чем остальные отрасли ИТ. Руководитель QA команды должен быть осведомлен о последних тенденциях, выбирать правильное тестовое окружение и быть готовым следовать инновациям в тестировании.

В этом случае применим принцип постоянной адаптации к изменяющимся обстоятельствам.

**Этот принцип можно реализовать следующим образом:**

- отслеживать технологические новинки;
- изучать custom-прошивки и использовать их в тестировании;
- планировать тесты новых версий OS/моделей устройств заранее.

В заключение стоит отметить, что наличие дефектов делает мобильные приложения неконкурентоспособными, ведь, как правило, у пользователя всегда есть несколько альтернатив.

Потребность в постоянных изменениях требует максимальной гибкости в разработке и тестировании мобильных приложений. Именно поэтому Agile подходит для мобильного тестирования лучше других существующих методологий.