**Этапы тестирования ПО**

**Всего принято выделять 7 этапов тестирования:**

1. **Работа с требованиями.** Знакомство с требованиями заказчика, что должен из себя представлять итоговый продукт, обсуждение.

Команда тестирования знакомится с требованиями заказчика, функционалом продукта. На данном этапе важно непрерывное общение команд, необходимо задавать вопросы разработчикам.

**Тщательное изучение требований должно:**

* + выявить противоречия в требованиях;
  + помочь определить потенциальные дефекты в функционале.

1. **Разработка стратегии тестирования.** Оценка сроков тестирования, выявление среды тестирования, объединение всей информации, полученной при работе с требованиями.

Данный этап важен для лидов или менеджеров, поскольку от понимания полученной на предыдущем этапе информации зависит качество тестирования.

**Тест — лид должен:**

* резюмировать полученную информацию,
* оценить сроки тестирования,
* разработать стратегию тестирования: определить виды тестирования, которые можно применить к проекту, проанализировать имеющиеся среды и ресурсы, что имеется для проведения тестирования, описать приоритеты для непредвиденных ситуаций, как и где будет вестись тестовая документация;
* определение среды тестирования: какое оборудование необходимо для тестирования,
* составить план, который содержит описание, с чего начинается и чем заканчивается тестирование, и что будет тестироваться.

1. **Создание тестовой документации.** Написание сценариев, которые позволят проверить функционал.

Цель данного этапа – создать документацию, объем которой будет охватывать детализацию, ход работ, а также вносить ясность для заказчика.

Общение с другими командами, понимание желаний заказчика напрямую влияют на качество тестовой документации. После проведенного тестирования можно проанализировать его успешность.

**Тестовая документация может состоять из:**

* тестовых сценариев: что и как будет проверяться при регресс-, дымовом и приемочном тестированиях;
* отчетности: результаты тестирования, списка багов и их серьезность;
* методологий тестирования.

Детализация тестовой документации зависит от проекта, поэтому она может отличаться и по охвату, и по формату, и по объему. Для тестировщика важно поддерживать документацию в актуальном виде, вносить любые изменения, связанные с изменением итогового продукта.

1. **Тестирование прототипа.** Тестирование основного функционала продукта, корректировка целей, добавление фичей.

При создании и тестировании прототипа продукта необходимо выявить основные отклонения от ожидаемого результата и соответствие с бизнес-стратегией. Здесь же выявляются ошибки в работе логики основного функционала, устраняются найденные уязвимости и дефекты, допущенные на этапе разработки. Заказчик может сам участвовать в процессе тестирования прототипа, чтобы он мог оценить, на каком этапе находится разработка продукта. После тестирования выдвигаются пожелания со стороны заказчика. Новые пожелания необходимо задокументировать, оценить сроки, внедрить в проект и передать на осмотр заказчику.

Самый подходящий метод тестирования прототипа – проведение закрытого бета-тестирования, когда продукт тестирует продукт малое количество людей, которые в итоге будут использовать его после релиза. Это помогает учесть пожелания конечных пользователей.

Очень важно лиду или менеджеру проекта передавать информацию тестировщикам и разработчикам о пожеланиях заказчика, на какие частые сбои в продукте натыкались пользователи для того, чтобы сделать его более понятным.

1. **Основное тестирование**. Выполнение общей проверки продукта.

Тестирование программного обеспечения является самым длительным и объемным процессом. Здесь формируются репорты о найденных дефектах, выполняется набор тестовых сценариев, создается тестовая среда, выполняется тестирование, виды которого были задокументированы на этапе создания тестовой документации. Смоук- и регресс-тестирования являются одними из основных видов тестирования, которые проводятся на данном этапе.

Важно понимать: невозможно найти все ошибки в продукте. Но и не найти ошибки при тестировании можно считать провалом. Главная цель — не сделать идеальный продукт без ошибок, а найти максимальное количество дефектов, которые потенциально могут сломать систему.

1. **Стабилизация.** На данном этапе происходит работа над устранением багов.

Наверное, самый непонятный с точки зрения формулировки этап. На нем заканчивается работа с пожеланиями заказчика и фиксируются найденные баги. Он является связующим звеном между командой тестирования и командой разработчиков. На протяжении создания онлайн-ресурса команда разработчиков занималась своими делами, реализуя «хотелки» заказчика, а тестировщики репортили о новых дефектах. А на этапе стабилизации разработчики начинают слушать тестировщиков, устраняя то, что уже работает, но некорректно.

Если продукт существует в какой-то большой системе, то на данном этапе также проверяется коммуникация системы и продукта, то есть проводится интеграционное тестирование.

1. **Эксплуатация.** Проводится регресс-тестирование, устранение ошибок, которые нашел конечный пользователь.

Пожалуй, самый волнительный период. После устранения дефектов команда разработчиков переходит в этап тестирования продукта в продакшн-среде. Так как многие ставят точку на проекте после релиза, очень важно подметить, что здесь происходит не только релиз продукта, но и пост-релизовая поддержка.  На данном этапе необходимо сделать акцент на том, что говорят пользователи, важно прислушиваться к их мнению, поскольку они участвуют не только в использовании продукта, но и тестировании, натыкаясь на ненайденные ошибки.