

## Вопросы по теории тестирования:

### 1. Что подразумевается под тестовой средой?

**Среда тестирования** — это настройка программного и аппаратного обеспечения для групп тестирования для выполнения тестовых случаев. Другими словами, он поддерживает выполнение теста с настроенным оборудованием, программным обеспечением и сетью. Тестовая среда настраивается в соответствии с требованиями тестируемого приложения.

Вообще существует несколько сред:

**Среда разработки (Development Env)** — в ней разработчики пишут код, проводят отладку, исправляют ошибки, выполняют Unit-тестирование. За эту среду отвечают также разработчики.

**Среда тестирования (Test Env)** — в этой среде работают тестировщики. Тут тестируются новые билды. Здесь тестировщики проверяют функционал, проводят регрессионные проверки, воспроизводят ошибки. Эта среда появляется во время начала динамического тестирования;

**Интеграционная среда (Integration Env)** — иногда реализована в рамках среды тестирования, а иногда в рамках превью среды. В этой среде собрана необходимая для end-to-end тестирования схема взаимодействующих друг с другом модулей, систем, продуктов. Собственно, необходима она для интеграционного тестирования. Поддержка среды – также, как и в случае со средой тестирования

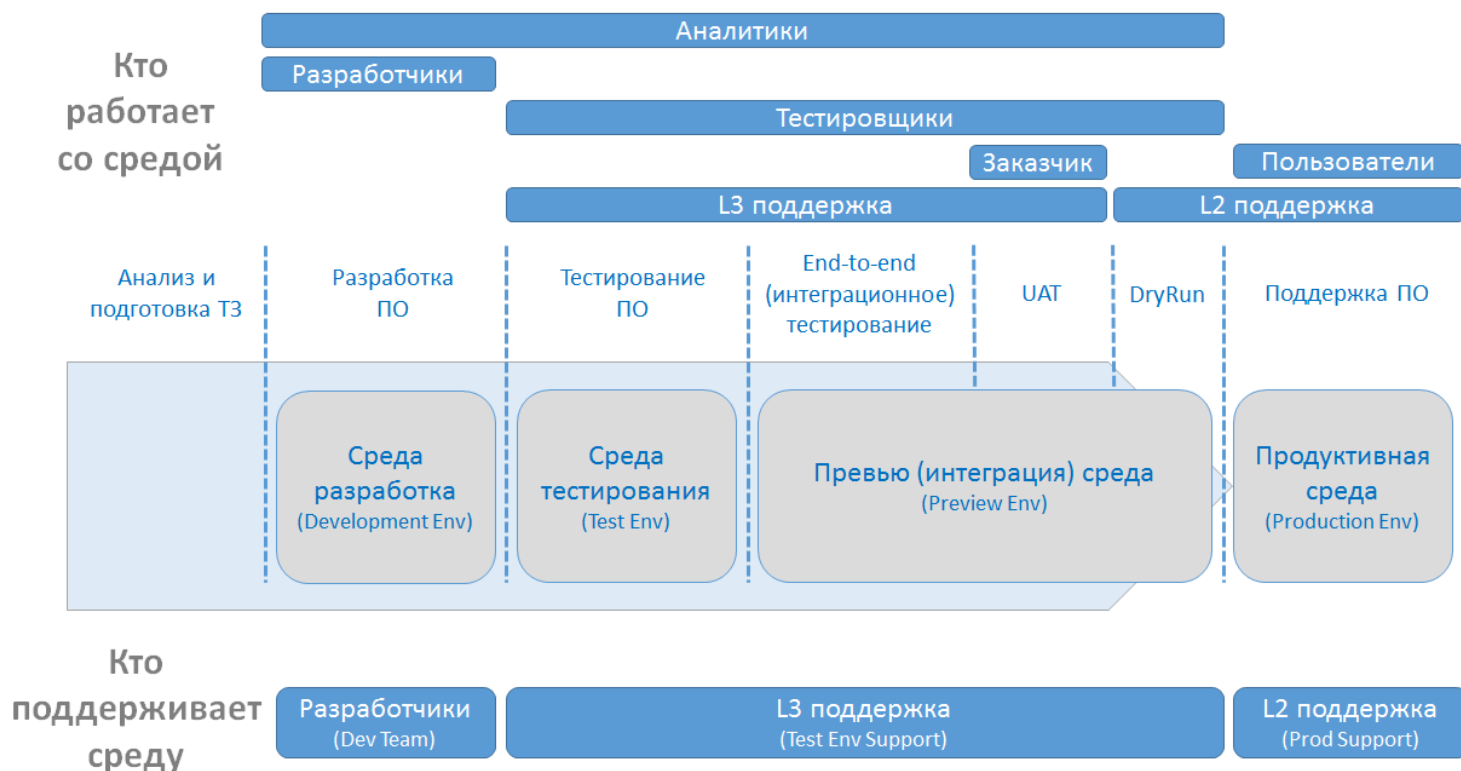
**Превью среда (Preview, Preprod Env)** — в идеале, это среда идентичная или максимально приближенная к продуктивной: те же данные, то же аппаратно-программное окружение, та же производительность. Она используется, чтобы сделать финальную проверку ПО в условиях максимально приближенным к «боевым». Здесь тестировщики проводят заключительное end-to-end тестирование функционала, бизнес и/или пользователи проводят UAT, а команды поддержки L3 и L2 выполняют DryRun (пробную установку релиза). Как правило за эту среду отвечает группа L3 поддержки.

**Продакшн среда (Production Env)** — среда, в которой работают пользователи. С этой средой работает команда L2 поддержки устанавливая поставки ПО или патчи с исправлениями, выполняя настройки, отвечая за работоспособность всех систем. Инциденты и проблемы требующие исправления ПО передаются в работу команде на L3.

Испытательный стенд (Test Bed) — более глобальная сущность и включает в себя operating system, configuration management for the products, hardware, network topology и т. д. В некоторых случаях испытательный стенд может представлять собой комбинацию тестовой среды и тестовых данных, которые он использует. Настройка правильной

среды тестирования гарантирует успех тестирования ПО. Любые недостатки в этом процессе могут привести к дополнительным затратам и времени для клиента.

В настройке тестовой среды участвуют: Системные администраторы, Разработчики, Тестировщики.



## 2. В чем отличие build от release?

Билд это номер, даваемый ПО при передаче от разработчиков тестирующим. Релиз — это номер, даваемый ПО при передаче конечному пользователю.

**Build** — это сборка, которую разработчики передают тестирующим для проверки наличия ошибок и соответствия требованиям.

**Release** — проверенная сборка, которую после тестирующих передается в продакшн (окончательная версия).

## 3. Основные фазы тестирования?

**Pre-Alpha** — ПО является прототипом. Пользовательский интерфейс завершен, но функционал не полный. Необходим для ознакомления с будущими возможностями ПО. На данной фазе ПО еще не публикуется.

**Alpha** — является ранней версией программного продукта, тестирование которой проводится внутри фирмы-разработчика. В некоторых случаях тестирование может быть передано на аутсорс.

**Beta** — практически готовый продукт, который разработан в первую очередь для тестирования конечными пользователями. Цель состоит в том, чтобы получить отзывы клиентов о продукте и внести соответствующие изменения в ПО.

**Release Candidate (RC)** — основываясь на отзывах Beta Test, возможные ошибки в каждой из фичей уже устранены и разработчики выпускают версию на которой проводится регрессионное тестирование.

**Release** — финальная версия программы, которая готова к использованию.

#### 4. **Что такое импакт анализ?**

**Impact Analysis (импакт анализ)** - это исследование, которое позволяет указать затронутые места (affected areas) в проекте при разработке новой или изменении старой функциональности, а также определить, насколько значительно они были затронуты.

Затронутые области требуют большего внимания во время проведения регрессионного тестирования.

Импакт анализ **может быть полезным в следующих случаях:**

- есть изменения в требованиях;
- получен запрос на внесение изменений в продукт;
- ожидается внедрение нового модуля или функциональности в существующий продукт;
- каждый раз, когда есть изменения в существующих модулях или функциональностях продукта.

В настоящее время продукты становятся все более большими и комплексными, а компоненты все чаще зависят друг от друга. Изменение строчки кода в таком проекте может "сломать" абсолютно все.

**Информация о взаимосвязи** и взаимном влиянии изменений **могут помочь QA:**

- сфокусироваться на тестировании функциональности, где изменения были представлены;
- принять во внимание части проекта, которые были затронуты изменениями и, возможно, пострадали;
- не тратить время на тестирование тех частей проекта, которые не были затронуты изменениями.

## 5. Критерии выбора тестов?

6. Принципы тестирования?
7. Что такое независимое тестирование?
8. Что такое тестирование со сдвигом влево?
9. Почему тестирование делится на отдельные этапы?
10. Какие есть этапы тестирования?
11. Как вы тестируете продукт, если требования еще не зафиксированы?
12. Как вы узнаете, было ли создано достаточно тестов для тестирования продукта?
13. Что подразумевается под тестовыми данными?
14. Что такое бета-тестирование?
15. Что означает пилотное тестирование? (Pilot)
16. Что такое бизнес – логика?
17. Какие существуют основные виды тестирования ПО?
18. Методы тестирования?
19. Что такое пирамида / уровни тестирования?
20. Что такое деструктивное/разрушающее/негативное тестирование? (DT — Destructive testing)
21. Что такое недеструктивное/неразрушающее/позитивное тестирование? (NDT – Non Destructive testing)
22. Что подразумевается под компонентным/модульным/юнит тестированием? (Component/Module/Unit testing)
23. Что подразумевается под интеграционным тестированием?
24. Разница между Unit testing и Integration testing?
25. Что такое системное интеграционное тестирование? (SIT — System Integration testing)
26. В чем разница между тест-драйвером и тест-заглушкой? (Test Driver and Test Stub)
27. Что подразумевается под системным тестированием?
28. Что такое функциональное тестирование?
29. Что такое тестирование совместимости/взаимодействия? (Compatibility/Interoperability testing)
30. Что такое тестирование на соответствие? (Conformance/Compilance testing)
31. Что такое нефункциональное тестирование?
32. Что вы знаете о Тестировании удобства пользования?

33. Отличия тестирование на удобство пользования и тестирования доступности? (Usability Vs. Accessibility testing)
34. Что такое тестирование интерфейса?
35. Что такое тестирование рабочего процесса/воркфлоу? (Workflow testing)
36. Что вы знаете о пользовательском приемочном тестировании? (UAT – User Acceptance testing)
37. Что такое эксплуатационное приемочное тестирование? (OAT — Operational Acceptance testing)
38. Что такое инсталляционное тестирование?
39. Что вы знаете о тестировании безопасности?
40. Что такое конфигурационное тестирование?
41. Объясните, что такое тестирование N+1
42. Что вы знаете о тестировании сборки? (Build Verification Test)
43. Что такое тестирование потоков? (Thread testing)
44. Что вы знаете о мутационном тестировании? (Mutation testing)
45. В чем разница между Baseline и Benchmark testing?
46. Что такое A/B тестирование?
47. Что означает сквозное тестирование? (E2E — End-to-End)
48. В чем разница между E2E и системным тестированием?
49. Что такое параллельное тестирование? (Parallel testing)
50. Чем AdHock тестирование отличается от Исследовательского тестирования?