1. Типы и методы тестирования
2. Принципы тестирования (7 шт)
3. Работа с требованиями
4. Понятия QA, QC, тестирование
5. Верификация и валидация
6. Техники функционального тестирования
7. Зачем нужно тестирование? Место тестирования в SDLC?
8. «Протестируй карандаш», ожидается использование методов тестирования, типов
9. Методологии разработки, эстимация задач (оценка трудозатрат, времени и тд), RCA (анализ первопричин) - это особая техника, которую можно использовать для устранения проблем. С помощью него вы анализируете проблему, используя определенный набор шагов для поиска основной причины. В основе RCA лежит принцип, согласно которому бесполезно устранять симптомы проблемы, игнорируя ее истоки.
10. Что такое тестирование
11. Цель тестирования (основная: получение информации о текущем состоянии ПО, трансляция этого в команду и как следствие способствующая повышению качества ПО)
12. **Техники тест-дизайна**
13. Классы эквивалентности (даты, имена, email)
14. Граничные значения, матрица traceability - Матрица соответствия требований используется QA-инженерами для валидации покрытия требований по продукту тестами. Цель «Traceability Matrix» состоит в том, чтобы выяснить: какие требования «покрыты» тестами, а какие нет. избыточность тестов (одно функциональное требование покрыто большим количеством тестов).
15. Что такое доменное тестирование? Примеры? **Доменное тестирование**— это техника тестирования метода чёрного ящика, направленная на уменьшение количества тестов путём одновременной проверки множества различных позитивных значений. Данная техника базируется на одновременном использовании техник эквивалентного разбиения и анализа граничных значений. Суть доменного тестирования не только в том, чтобы выбирать показательные значения при помощи 2 техник, но ещё и одновременной проверке множества позитивных условий одновременно. Однако, **негативные тесты ни в коем случае нельзя объединять друг с другом!**
16. Техники черного и белого ящика?Исследовательское тестирование
17. Понимать для чего вообще нужны техники тест-дизайна, а не просто их зазубрить!
18. **Тестовая документация**
19. Отчет о дефектах (bug-report)
20. Тест-план, тест-стратегия, отчеты о тестировании
21. Работа с баг-трекинговой системой (Jira, Redmine, youtrack)
22. Эстимация тестирования
23. Системы управления тестированием (testrail)
24. Когда и где лучше применят тест-кейсы и чек-листы, если кандидат работал в гугл-доке, это плюсик)
25. ВАЖНО! Сначала поискал баг, заводили или нет, затем посмотри его жизненный цикл, может он ревьюится или закрыт, Не заводите заново!
26. **Тестирование Web-приложений**
27. Понятие клиент-серверной архитектуры
28. Работа с devtools
29. HTTP-методы, протоколы (статус-коды)
30. Форматы XML и JSON, (синтаксис)
31. Работа с логами (инструменты, где хранятся, как посмотреть и тд)
32. Кэш, куки, storage
33. Работа с анализаторами
34. Web-sockets
35. **Тестирование API**
36. Общее понимание API
37. Работа (автотесты) в Postman, создание колекций
38. Инсомния, сваггер – опыт работы
39. SOAP (UI)
40. Что такое веб-сервис
41. **Базы данных**
42. Что такое БД/СУБД
43. SQL: select, join
44. Типы БД
45. Нормальные формы и формы отношений внутри таблиц
46. NoSQL: что такое, примеры
47. Транзакция, миграция. Транзакция представляет собой последовательность операторов языка SQL, которая рассматривается как некоторое неделимое действие над базой данных, осмысленное с точки зрения пользователя. В то же время, это логическая единица работы системы. Миграция базы данных позволяет перенести базу вместе с данными как для той же СУБД (например, с одного сервера Microsoft SQL Server на другой), так и между разными СУБД (из MSSQL в PostgreSQL и обратно).
48. DDL, DML, ТCL – операторы
49. **Тестирование мобильных приложений**
50. Особенности тестирования мобильных приложений
51. Типы мобильных приложений
52. Работа с логами
53. Работа в Android Studio
54. Работа в X-Code
55. Знание гайдлайнов
56. Принципы формирования тестовой фермы
57. Кроссплатформы (флаттер)
58. Снифферы
59. Доставка билдов, крашлитики
60. Нативные автотесты
61. **Дополнительные инструменты**
62. Git, GitHub
63. CI/CD pipeline
64. Bash
65. Jmeter
66. Виртуальные машины
67. Docker