1. Тестирование: понятие, критерии. Требования к тестированию. Процесс проведения отладки.

Тестирование - процесс выполнения ПО системы или компонента в условиях анализа или записи получаемых результатов с целью проверки (оценки) некоторых свойств тестируемого объекта.

Отладка – процесс поиска, локализация и исправления ошибок в программе.

Критерии тестирования:

1. Функциональная корректность – проверка на выполнение всех функций и соответствие ожидаемому поведению;
2. Надёжность – оценка стабильности продукта в различных условиях, выявление и устранение сбоев.
3. Производительность – оценка скорости и эффективности работы продукта, проверка на высокие нагрузки и скорость отклика;
4. Совместимость – проверка работы продукта на разных платформах, браузерах, устройствах. Обеспечение работоспособности в различных окружениях;
5. Безопасность – проверка на уязвимости и угрозы безопасности, защита данных и конфиденциальности;
6. Удобство использования – оценка уровня удовлетворения пользователей, удобство навигации и интерфейса.
7. Цели и основные принципы тестирования.

Цель проектирования тестовых вариантов - систематическое обнаружение различных классов ошибок при минимальных затратах времени и стоимости.

Основные принципы тестирования:

1. Предполагаемые результаты должны быть известны до тестирования;
2. Следует избегать тестирования программы автором;
3. Доскональное изучение результатов конечного теста;
4. Необходимо проверять работу программы на неверных данных;
5. Вероятность наличия необнаруженных ошибок в части программы пропорциональна числу ошибок, уже обнаруженных в этой части.

Принципы тестирования:

1. Полное тестирование невозможно – ограничены ресурсы и время. Фокус на наиболее критических аспектах;
2. Раннее тестирование – тестирование начинается на ранних этапах разработки, что экономит ресурсы и время;
3. Исчерпывающее тестирование – важно покрыть все функциональные и нефункциональные области. Тестовые случаи должны покрывать их все.
4. Стадии разработки программного обеспечения: описание.
5. Подходы к формированию тестовых наборов: описание, примеры.
6. Среда тестирования: понятие, настройка. Документированность процесса тестирования: тестовый план и отчёт.

Среда тестирования – настройка программного и аппаратного обеспечения для групп тестирования для выполнения тестовых случаев.

Тестовый план – это документ, или набор документов, который содержит тестовые ресурсы, перечень функций и подсистем, подлежащих тестированию, тестовую стратегию, расписание тестовых циклов, фиксацию тестовой конфигурации (состава и конкретных параметров аппаратуры и программного окружения), определение списка тестовых метрик, которые на тестовом цикле необходимо собрать и проанализировать (например, метрик, оценивающих степень покрытия тестами набора требований)

1. Методология тестирования сложных систем. Понятие сложной системы.

Сложная система – система, состоящая их множества взаимодействующих составляющих, вследствие чего сложная система приобретает новые свойства, которые отсутствуют на подсистемном уровне и не могут быть сведены к свойствам подсистемного уровня.

Методология:

1. Эффективно начинать тестирование комплексных систем на ранних стадиях разработки ПО.
2. Тестирование организует цикл: тестирование – исправление – повторное тестирование.
3. Тестирование выполняется всегда для всех объектов системы вне зависимости от её критичности.
4. Понятие жизненного цикла программного обеспечения. Стандарты и их классификация.
5. Каскадная модель жизненного цикла. Последовательность этапов. Особенности разработки по каскадной модели.
6. Спиральная модель жизненного цикла. Особенности и проблемы разработки по спиральной модели.
7. Каскадная и спиральная модель: сходства и различия.
8. Характеристики программного продукта. Этапы решения задач. Понятие алгоритма и его свойства.
9. Классификация видов тестирования.

Функциональное (работоспособность)/нефункциональное (удобство, качество).

Статическое (проверка документации)/динамическое (практическая проверка).

1. Функциональное тестирование. Понятие, основные функции, условия ошибки, режим выполнения. Примеры функционального тестирования.

Функциональное тестирование – проверка ПО на соответствие требованиям и спецификациям. Выполняется с использованием функциональной спецификации, предоставленной клиентом, и проверяет систему на соответствие функциональным требованиям.

Бизнес-требования – исходные данные для функционального тестирования.

Можно использовать ручное и автоматическое тестирование.

Примеры:

1. Ручное – основный вид тестирования, проверяет все требования;
2. Регрессивное – проверка что изменения исправляющие ошибки не породили новые
3. Нефункциональное тестирование. Понятие, основные функции, условия ошибки, режим выполнения. Примеры нефункционального тестирования.
4. Модульное тестирование. Понятие, основные методы, алгоритм выполнения, преимущества и недостатки.
5. Дымовое тестирование. Понятие, основные методы, алгоритм выполнения, преимущества, примеры дымового тестирования, отличия от санитарного тестирования.
6. Санитарное тестирование. Понятие, основные методы, алгоритм выполнения, преимущества, примеры санитарного тестирования, отличия от дымового тестирования.
7. Интеграционное тестирование. Понятие, необходимость интеграционного тестирования, основные методы, критерии входа и выхода для интеграционного тестирования.
8. Тестирование методом «черного ящика». Типы и область тестирования, примеры.
9. Тестирование методом «белого ящика». Типы и область тестирования, примеры. Преимущества и недостатки метода.
10. Тестирование методом «серого ящика». Стратегия, преимущества и проблемы тестирования. Область тестирования. Этапы выполнения тестирования.
11. Регрессивное тестирование. Понятие, необходимость, методы проведения регрессивного тестирования. Выбор регрессивного теста и тестовых случаев. Различия между повторным тестированием и регрессивным. Проблемы регрессивного тестирования.
12. Определение и цель ручного тестирования. Преимущества и недостатки ручного тестирования. Типы тестирования.
13. Интеграционное тестирование. Понятие, цель и объект тестирования. Методы и инструменты тестирования.
14. Юзабилити тестирование: понятие, основные подходы планирования и разработки программных продуктов.
15. Экспертный подход к юзабилити-тестированию: критерии. Преимущества и недостатки подхода.
16. Пользовательский подход к юзабилити-тестированию: критерии, методы измерения. Преимущества и недостатки подхода.
17. Автоматизация тестирования: понятие, типы, принципы. Требования к тестировщику. Преимущества и недостатки автоматизации тестирования.
18. Понятие утверждений и их параметры. Модели и основные виды утверждений. Понятие и категории директив.
19. Определение, цель и этапы тестирования производительности. Критерии для измеряемых метрик.
20. Структура и виды CASE-средств. Критерии при выборе.
21. Структуры среды разработки. Факторы при выборе.
22. Характеристика основных возможности среды разработки.
23. Основные инструменты среды для создания, исполнения и управления информационной системой.
24. Средства обработки информации: определение, виды.
25. Организация работы в команде разработчиков. Виды моделей и роли.
26. Система контроля версий: совместимость, установка, настройка.
27. Характеристика методов обеспечения кроссплатформенности информационной системы.
28. Особенности сервисно-ориентированной архитектуры, основные принципы.
29. Интегрированные среды разработки для создания независимых программ: назначение, сфера применения.
30. Особенности объектно-ориентированных языков программирования.
31. Диаграммы UML вариантов использования, последовательности, кооперации: цель построения и синтаксис.
32. Диаграммы UML развертывания, компонентов, потоков данных: цель построения и синтаксис.
33. Архитектура информационной системы: виды, типы групп, применение архитектурных описаний.
34. Определение конфигурации информационной системы. Этапы, методы и средства конфигурирования.
35. Критерии выбора технических средств проекта информационной системы.
36. Репозиторий проекта информационной системы: понятие и описание процесса создания.
37. Алгоритм определения уровня доступа к репозиторию в системе контроля версий.
38. Процесс настройки среды разработки информационной системы: описание установки предпочитаемых инструментов и языков кода.
39. Инструменты мониторинга разработки проекта.
40. Требования к пользовательскому интерфейсу информационной системы.
41. Принципы создания графического пользовательского интерфейса (GUI).
42. Понятие и формы спецификации языка программирования для разработки информационной системе.
43. Методы организации ввода-вывода информации в информационной системе.
44. Спецификация настроек типовой информационной системы.
45. Сообщения между модулями: назначения, вход и разделы модуля.
46. Приложения для моделирования процессов: виды моделей, требования.
47. Разработка приложений: виды, этапы разработки.
48. Отладки приложения: определение, существующие методы.
49. Интеграция модуля: определение, возможность, способы интеграции.