75 вопросов на собеседовании QA

1. Что такое тестирование программного обеспечения?

Это процесс анализа ПО и его документации для определения того, удовлетворяет ли оно потребностям конечного потребителя. Также это процесс выявления дефектов и повышения качества продукта путем проверок его производительности, возможностей, полезности и удобства использования.

2. Что такое контроль качества и обеспечение качества?

Контроль качества — это тщательное тестирование программы на наличие дефектов, а также проверка того, что программное обеспечение соответствует всем требованиям, выдвинутым заказчиком.

Обеспечение качества – это подход, который помогает убедиться, что методы, технологии и процессы, используемые для создания качественных результатов, применяются правильно.

3. Ручное и автоматизированное тестирование ПО

Ручное тестирование программного обеспечения – это процесс, в котором тестировщики вручную прогоняют тест-кейсы, после чего пишут отчеты о результатах тестирования. При [автоматизированном тестировании](https://qarocks.ru/automated-testing/) ПО эти функции выполняются средствами автоматизации, которые включают в себя код и тестовые сценарии.

4. Каковы преимущества ручного тестирования?

Плюсы ручного тестирования:

* Оно дешевле автоматизированного.
* Точная и оперативная визуальная обратная связь.
* Идеально подходит для тестирования ПО при внесении незначительных изменений.
* Удобно для Ad-hoc тестирования.
* Тестировщикам не обязательно знать автоматизацию и языки программирования.
* Отлично подходит для тестирования пользовательского интерфейса.

5. Каковы недостатки ручного тестирования?

Минусы ручного тестирования:

* Подверженность человеческому фактору.
* Некоторые задачи отнимают больше времени, если выполнять их вручную.
* Расходы на ручное тестирование со временем возрастают, поэтому в долгосрочной перспективе оно может обойтись дороже автоматизированного.
* Невозможно записать прохождение всех тестовых сценариев, поэтому их бывает трудно воспроизвести.

6. Какие нужны навыки, чтобы стать тестировщиком?

Тестировщикам программного обеспечения необходимо:

* Уметь решать проблемы.
* Владеть навыками письменного и устного общения.
* Быть внимательным к деталям.
* Уметь работать в условиях давления.
* Одинаково хорошо работать как в одиночку, так и в команде.
* Иметь организаторские способности.
* Владеть соответствующими техническими знаниями.

7. Что такое SDLC?

Эта аббревиатура расшифровывается как Software development lifecycle, в переводе – жизненный цикл разработки программного обеспечения. SDLC включает в себя все этапы разработки ПО, в том числе сбор и анализ требований, проектирование, написание кода, тестирование, развертывание и сопровождение.



8. Что такое тест-кейс?

Тест-кейс используется для проверки соответствия приложения его требованиям. Это документированный набор действий, включающий предварительные условия, входные значения, ожидаемые и фактические результаты проверки.

9. Что такое сценарий тестирования?

Тестовый сценарий используется для тестирования функции приложения от начала до конца. В одном тестовом сценарии можно разместить несколько тест-кейсов. Применяется, если время на тестирование ограничено.

10. Что такое план тестирования?

План тестирования – это официальный документ, определяющий объем тестирования, используемый метод, необходимые ресурсы и расчетное время для завершения процесса. Он составляется на основе спецификаций (требований к программному обеспечению).

11. Что такое тестовые данные?

Тестовые данные – это информация, которая используется для тестирования программного обеспечения с различными входными данными и определения, соответствует ли полученный результат запланированному. Тестовые данные генерируются в зависимости от потребностей компании.

12. Что такое тестовый скрипт?

Автоматизированный тест-кейс, написанный на любом языке программирования, называется тестовым скриптом. По сути, это набор инструкций для оценки функциональности приложения.

13. Какие существуют виды ручного тестирования?

Ручное тестирование подразделяется на:

* Черный ящик
* Белый ящик
* Интеграционное
* Модульное
* Системное
* Приемочное

14. Что такое тестирование методом “черного ящика”?

QA инженеры используют тестирование методом “черного ящика”, когда они не знают внутренней архитектуры или у них нет доступа к структуре кода. К его техникам относятся:

* Разбиение на классы эквивалентности
* Анализ граничных значений
* Построение графа причинно-следственных связей

15. Что такое тестирование методом “белого ящика”?

В отличие от тестирования “черного ящика”, метод “белый ящик” включает в себя анализ внутренней архитектуры системы и/или ее реализации, а также доступ к программному коду. К его техникам относятся:

* Покрытие операторов в коде
* Покрытие решений

16. В чем разница между альфа- и бета-тестированием?

Альфа-тестирование проводят разработчики перед релизом. Бета-тестирование проводится конечными пользователями на практически финальной версии продукта перед его выпуском на широкую аудиторию.

17. В чем разница между верификацией и валидацией?

Верификация оценивает программное обеспечение на этапе разработки, выясняя, соответствует ли продукт ожидаемым требованиям. Валидация оценивает готовое ПО на соответствие требованиям заказчика и конечного пользователя.

18. Что такое тестовая среда?

Это среда, используемая для тестирования приложения. В нее входит аппаратное обеспечение, а также любое программное обеспечение, необходимое для запуска тестируемой программы.

19. Что такое Sanity тестирование?

Sanity тестирование проводится на уровне релиза и проверяет, работают ли основные функции в соответствии с заявленными в спецификации требованиям. Оно также считается одним из этапов регрессионного тестирования.

20. Когда должен быть внедрен менеджмент конфигураций?

Это должно быть сделано на этапе планирования тестирования.

21. Перечислите четыре различных уровня тестирования

Четыре уровня:

* Модульное
* Интеграционное
* Системное
* Приемочное

22. В чем разница между багом и дефектом?

Баг – это ошибка в программном обеспечении, которая обнаруживается во время тестирования. Дефект – это расхождение между ожидаемыми и фактическими результатами, которое обнаруживается разработчиком после того, как продукт введен в эксплуатацию.

23. В чем разница между ошибкой и сбоем?

Если программа не может быть запущена или скомпилирована во время разработки, это ошибка. Если конечный пользователь сталкивается с неправильной работой ПО, это сбой.

24. Что такое тестирование графического интерфейса?

Это тестирование интерфейса между программным обеспечением и конечным пользователем. GUI – сокращение от Graphics User Interface (графический интерфейс пользователя).

25. Когда должно заканчиваться тестирование?

Существует несколько критериев для прекращения тестирования:

* Количество ошибок стало меньше согласованного
* Закончились сроки тестирования или пришло время релиза
* Исчерпан бюджет, выделенный на тестирование
* Пройден определенный процент тест-кейсов
* Закончился период альфа- или бета-тестирования
* Код, функциональность или покрытие требований были выполнены в заявленной точке

26. Для чего нужно тестирование программного обеспечения?

Тестирование программного обеспечения необходимо для обеспечения качества и надежности ПО. Оно:

* Помогает выявить баги, ошибки и другие проблемы в программном обеспечении, чтобы их можно было устранить до выпуска продукта.
* Гарантирует, что программное обеспечение отвечает всем требованиям, установленным заказчиком, и работает так, как ожидается.
* Позволяет убедиться в том, что программное обеспечение безопасно и может противостоять вредоносным атакам.

27. Назовите уровни ручного тестирования

К уровням ручного тестирования относятся:

* Модульное
* Интеграционное
* Системное
* Приемочное
* Тестирование производительности
* Тестирование безопасности
* Тестирование совместимости
* Тестирование удобства использования
* Тестирование установки
* Smoke тестирование
* Sanity тестирование
* Регрессионное тестирование

28. Как проводят ручное тестирование?

Ручное тестирование – это процесс выявления ошибок в программном продукте без использования автоматизированных инструментов.

Первым шагом ручного тестирования является определение объема тестирования. Это может быть конкретный модуль, функциональность, функция или вся система в целом.

Следующим шагом является разработка тестовых примеров на основе установленного объема тестирования. Они должны включать сценарии тестирования, предусловия, ожидаемые результаты и все другие детали, необходимые для выполнения тестов.

После разработки тестовых примеров тестировщики выполняют их, чтобы найти любые расхождения между ожидаемыми и фактическими результатами.

При выполнении тестов тестировщики фиксируют результаты для дальнейшего анализа.

29. Что такое тест-кейс?

Тест-кейс – это набор условий или переменных, при которых тестировщик определяет, работает ли программная система или одна из ее функций так, как это было изначально запланировано. Тест-кейс включает предусловия, этапы тестирования, описание окружения, ожидаемые и фактические результаты.

30. Какова роль документации в ручном тестировании?

Документация является неотъемлемой частью ручного тестирования. Очень важно записывать все шаги, предпринятые в процессе тестирования, чтобы обеспечить качественное тестовое покрытие и точные результаты. Документация помогает вести контрольный журнал, который может быть использован для оценки результатов прошлых тестов и выявления областей для улучшения. Кроме того, она является справочным материалом для других тестировщиков, которые могут быть незнакомы с тестируемой системой.

Вопросы продвинутого уровня

31. Каковы типы тестирования программного обеспечения?

Тестирование программного обеспечения подразделяется на две основные категории:

* [Функциональное](https://qarocks.ru/functional-testing/)
* [Нефункциональное](https://qarocks.ru/non-functional-testing/)

32. Что такое функциональное тестирование?

Функциональное тестирование – это тип тестирования методом “черного ящика”. Оно фокусируется на функциональных требованиях к программному обеспечению, а не на его внутренней реализации.

Функциональное тестирование проверяет программное обеспечение на соответствие функциональным требованиям или спецификациям, игнорируя нефункциональные характеристики, такие как удобство использования и надежность.

Цель функционального тестирования – убедиться, что программное обеспечение соответствует функциональным требованиям и решает проблемы целевых пользователей.

Типы функционального тестирования:

* Модульное тестирование
* Интеграционное тестирование
* Регрессионное тестирование
* Системное тестирование
* Smoke-тестирование
* Тестирование производительности
* Стресс-тестирование

33. Что такое нефункциональное тестирование?

Нефункциональное тестирование проверяет характеристики или качества системы, специально запрошенные заказчиком. Среди них – безопасность, масштабируемость и удобство использования.

Нефункциональное тестирование проводят после функционального. Оно изучает аспекты, которые не связаны с функциональными требованиями к программному обеспечению, и гарантирует, что программа безопасна, адаптивна, быстро работает и способна выдерживать перегрузки.

34. Каковы преимущества автоматизации?

* Автоматизированное выполнение тестов экономит значительное количество времени.
* При тщательной подготовке тестовых сценариев исключаются человеческие ошибки во время тестирования.
* Такие авто-инструменты непрерывной интеграции, как Jenkins, которые также могут быть настроены на рассылку результатов тестов заинтересованным сторонам, можно использовать для планирования выполнения тестов на ночное время.
* Автоматизированное тестирование использует намного меньше ресурсов. Выполнение тестов после их автоматизации почти не занимает времени у QA инженеров.

35. Что такое регрессионное тестирование?

Регрессионное тестирование – это повтор выполненных ранее тест-кейсов с целью убедиться, что существующие функции работают нормально.

Действия при регрессионном тестировании:

1. Повторное тестирование: все тест-кейсы из текущего набора выполняются заново. Это достаточно дорого и отнимает много времени.
2. Регрессионные тесты делятся на три категории: функциональные, интеграционные и сквозные. Некоторые из тестов выбираются на этом этапе.
3. Определение приоритетов тест-кейсов: они ранжируются в соответствии с их влиянием на бизнес и важными функциональными возможностями.

36. Что такое тестовая среда?

Тестовая среда – это набор программного обеспечения и данных, используемых для тестирования программного модуля путем его запуска в различных условиях и мониторинга его поведения.

37. Отличия позитивного и негативного тестирования.

|  |  |
| --- | --- |
|  |  |
| **Позитивное тестирование** | **Негативное тестирование** |
| Позитивное тестирование гарантирует, что программное обеспечение работает так, как ожидается. Тест завершается неудачей, если во время позитивного тестирования была выявлена ошибка. | Негативное тестирование гарантирует, что ваше приложение может с легкостью справиться с неожиданным поведением пользователя или неправильным вводом данных. |
| При этом тестировании QA инженер всегда использует только достоверные данные | При проверке приложения на ввод ошибочных данные тестировщики стараются быть максимально изобретательными. |

38. Что такое критическая ошибка?

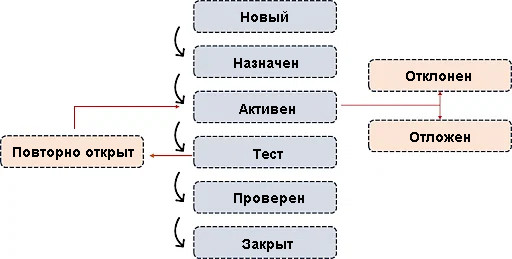
Критическая ошибка – это ошибка, которая потенциально может повлиять на главные функциональности приложения. Эта ошибка указывает на то, что работа важных функций или критических компонентов системы полностью нарушена. Приложение не передается конечным пользователям до тех пор, пока критическая ошибка не будет исправлена.

39. Что такое завершение тестирования?

Завершение тестирования – это финальный этап жизненного цикла тестирования ПО, на основании которого создается итоговый документ. В нем обобщаются все тесты, проведенные на протяжении разработки программного обеспечения, а также делается полный анализ исправленных дефектов и обнаруженных ошибок. Общее количество запланированных проверок, общее количество выполненных проверок, общее количество обнаруженных дефектов, общее количество устраненных дефектов, общее количество не устраненных ошибок, общее количество отклоненных ошибок и так далее – все это включается в этот документ.

40. Что такое жизненный цикл дефекта?

Жизненный цикл дефекта – это процесс, в ходе которого дефект проходит множество стадий за все время своего существования. Цикл начинается с момента обнаружения дефекта и завершается, когда дефект закрывается после проверки его устранения.



41. Что такое парадокс пестицида? Как его преодолеть?

Согласно парадоксу пестицида, если одни и те же тесты повторять многократно, то они в конечном итоге перестанут находить новые ошибки. Разработчики будут особенно осторожны в тех областях, где тестировщики обнаружили больше недостатков, и могут пропустить ошибки в других.

Методы, позволяющие избежать “парадокса пестицида”:

* Создание совершенно нового набора тестовых примеров для проверки различных аспектов программного обеспечения.
* Создание новых проверок и включение их в существующие тест-кейсы.

С помощью этих методов можно обнаружить больше недостатков в областях с низким уровнем дефектов.

42. Что такое тестирование API?

Это оценка интерфейсов прикладного программирования (API) на предмет их соответствия требованиям функциональности, надежности, производительности и безопасности. Проще говоря, тестирование API предназначено для обнаружения дефектов, несоответствий или отклонений от ожидаемого поведения. Как правило, приложения делятся на три слоя:

* Пользовательский интерфейс также известен как уровень представления.
* Для логической обработки бизнес-данных используется бизнес-слой или пользовательский интерфейс приложения.
* Тестирование API проводится на самом главном и важном уровне архитектуры программного обеспечения – бизнес-слое, предназначенном для моделирования и управления данными.

43. Что такое системное тестирование?

Системное тестирование – это вид тестирования, при котором проверяется все программное обеспечение. Системное тестирование проверяет соответствие приложения бизнес-требованиям.

44. Что такое приемочное тестирование?

Приемочное тестирование – это вид тестирования, проводимый возможным конечным пользователем или заказчиком, чтобы убедиться, что оно соответствует бизнес-требованиям и может быть выпущено для широкой аудитории.

45. В чем различие между утечкой дефекта и релизом бага?

Когда протестированное программное обеспечение выводится на рынок, а конечный пользователь обнаруживает ошибки, это называется утечкой дефекта. Это ошибки, которые команда тестирования упустила из виду на этапе тестирования.

Когда определенная версия программного обеспечения выпускается на рынок с ошибками, о которых известно и которые, как ожидается, будут исправлены в последующих версиях, это называется релизом бага. Чаще всего это низкоприоритетные проблемы, которые указываются в примечаниях к релизу при обмене информацией с конечными пользователями.

46. Что такое сортировка дефектов?

Сортировка дефектов – это процедура, в ходе которой дефектам присваивается приоритет в зависимости от различных характеристик, таких как серьезность, риск и количество времени, которое потребуется для устранения неисправности.

Для оценки дефектов собираются несколько заинтересованных сторон – команда разработчиков, команда тестировщиков, менеджер проекта, бизнес-аналитики и так далее, – чтобы определить порядок, в котором дефекты должны быть устранены.

47. Что такое интеграционное тестирование?

Интеграционное тестирование выполняется после модульного. При интеграционном тестировании проверяют группу связанных между собой модулей ПО. Его цель – выявить неисправности, связанные с их взаимодействием.

Типы интеграционного тестирования:

* “Большой взрыв” – проводится после объединения всех модулей.
* Нисходящее – тестирование и интеграция начинаются сверху и идут вниз.
* Восходящее – модули более низкого уровня тестируются перед перемещением вверх по иерархии к модулям более высокого уровня.
* Гибридное – сочетает в себе нисходящее и восходящее интеграционное тестирование. Интеграция при таком подходе начинается со среднего уровня, а тестирование проводится в обоих направлениях.

48. Что такое заглушка?

Во многих случаях, когда проводится интеграционное тестирование сверху вниз, модули нижнего уровня не создаются до тех пор, пока модули верхнего уровня не будут протестированы и интегрированы. В этих обстоятельствах используются заглушки или фиктивные модули для имитации поведения отсутствующего модуля путем предоставления жестко запрограммированного или прогнозируемого результата на основе входных переменных.

49. Что такое покрытие кода?

Количество кода, охваченного сценариями тестирования, называется покрытием кода. Оно отражает масштаб охвата приложения тестовыми наборами.

50. Что такое граф причинно-следственных связей?

Метод тестирования с использованием графа причинно-следственных связей – это способ разработки тестов методом “черного ящика”, который использует графическое представление входа (причины) и выхода (следствия) для построения теста. В этом методе используются различные обозначения для описания функций и других взаимосвязей между входными и выходными условиями.

51. Объясните технику разбиения на классы эквивалентности

Разбиение на классы эквивалентности – это метод тестирования “черного ящика”, основанный на спецификациях. Набор входных данных, определяющий условия тестирования, разбивается на логически сопоставимые группы таким образом, что использование даже одних тестовых данных из группы для тестирования можно считать аналогичным использованию всех остальных данных в этой группе.

52. Что такое анализ граничных значений?

В этом методе для разработки тест-кейсов в качестве входных данных используются верхние, нижние и средние значения границ классов, образованных при эквивалентном разбиении.

53. Что вы будете делать, если обнаружите серьезные ошибки в ПО?

В таких случаях лучший способ действий для тестировщиков — немедленно сообщить обо всех ошибках со статусом “критикал” или “блокер”. Поскольку подобные ситуации могут привести к серьезным проблемам, таким как недостаточное модульное или интеграционное тестирование, плохой дизайн, неправильные методы сборки или выпуска и так далее, следует обязательно связаться с руководством и предоставить документацию в качестве доказательства наличия проблемы.

54. Иногда рост организации настолько стремителен, что стандартные процедуры тестирования уже невозможны. Что делать в такой ситуации?

Это очень распространенная проблема в индустрии разработки ПО, которая обостряется с развитием новых технологий. В этом случае нет однозначного ответа, но кое-что сделать можно:

* Нанимать людей, которые действительно хороши в своем деле.
* Вопросы качества должны быть “жестко приоритизированы” руководством, с постоянным фокусом на клиенте.
* Все в компании должны понимать, что означает термин “качество” для конечного пользователя.

55. Когда можно с уверенностью сказать, что код соответствует своим спецификациям?

В большинстве компаний существуют стандарты написания кода, которым должны следовать все разработчики, но у каждого есть свое мнение о том, как делать лучше. Существует множество методов, таких как матрица трассируемости, чтобы гарантировать соответствие тест-кейсов требованиям. И когда все тест-кейсы пройдены, это означает, что код удовлетворяет требованию.

56. В чем разница между ручным и автоматизированным тестированием?

Ручное тестирование – это процесс проверки программного обеспечения на наличие дефектов. Оно требует, чтобы тестировщик вручную выполнял шаги тестирования и сравнивал фактические и ожидаемые результаты.

Автоматизированное тестирование использует специальное программное обеспечение для контроля выполнения тестов и сравнения фактических результатов с ожидаемыми. Автоматизированное тестирование выполняется быстрее, чем ручное, и существенно сокращает время, необходимое для завершения цикла тестирования.

57. Когда следует предпочесть ручное тестирование автоматизированному?

Ручное тестирование следует использовать, когда тесты являются специфическими или требуют человеческой интерпретации. Ручное тестирование также лучше подходит для исследовательского тестирования, тестирования удобства использования, тестирования на нескольких операционных системах или уникальном оборудовании.

58. Какие фазы включает в себя жизненный цикл тестирования ПО?

В жизненный цикл тестирования программного обеспечения входят следующие фазы:

* Планирование тестирования
* Анализ тестирования
* Разработка тестов
* Реализация процесса тестирования
* Выполнение тестов
* Анализ результатов тестирования
* Закрытие тестирования

59. Какие качества нужны хорошему тестировщику?

Хороший инженер по тестированию ориентирован на детали и организован, обладает отличными навыками решения проблем и может быстро и эффективно выполнять свою работу. Кроме того, он должен обладать навыками общения и быть отличным командным игроком, оставаться в курсе последних технологий и тенденций в области программного обеспечения и уметь применять их в процессе тестирования.

60. В чем разница между системным и интеграционным тестированием?

Системное тестирование – это вид тестирования программного обеспечения, при котором оценивается полностью собранный программный продукт. Оно проверяет, соответствует ли программное обеспечение требованиям, указанным в проекте и технических спецификациях на уровне системы. Системное тестирование также выявляет любые слабые места, ошибки или баги.

Интеграционное тестирование проверяет взаимодействие между двумя или более компонентами системы. Оно проводится после модульного тестирования и перед системным, и проверяет, как компоненты взаимодействуют друг с другом. Интеграционное тестирование необходимо для того, чтобы убедиться, что компоненты системы работают вместе именно так, как было запланировано.

61. Что такое каскадирование дефектов?

Каскадирование дефектов – это тип проблем при тестировании программного обеспечения, при котором последствия дефекта в одном модуле вызывает появление дефектов или проблем в других модулях. Это может произойти по разным причинам, но чаще всего это случается из-за того, что первоначальный дефект не был устранен должным образом.

Каскадирование дефектов может привести ко множеству проблем, от незначительного замедления производительности до значительных сбоев системы, что делает его серьезным риском для разработчиков и тестировщиков программного обеспечения.

62. Что означает термин “качество” в тестировании?

Под качеством при тестировании понимается степень, в которой продукт соответствует установленным требованиям, а также степень, в которой он удовлетворяет потребности и ожидания клиентов. Оно включает в себя как функциональные, так и нефункциональные аспекты продукта.

63. Какие существуют методы тестирования на основе опыта?

К методам тестирования на основе опыта относятся:

* Исследовательское тестирование
* Метод угадывания ошибок
* Ad-hoc тестирование
* Тестирование по чек-листам
* Альфа-тестирование
* Бета-тестирование
* Приемочное тестирование
* Тестирование удобства использования

64. Что такое нисходящий и восходящий подход в тестировании?

Подход к тестированию сверху вниз и снизу вверх относится к порядку проведения тестирования.

* Нисходящее тестирование начинается с самого высокого уровня и работает по нисходящей. Таким образом, каждый компонент более высокого уровня тестируется изолированно от компонентов более низкого уровня.
* Тестирование снизу вверх начинается с самого низкого уровня и работает по нарастающей. Таким образом, каждый компонент нижнего уровня тестируется изолированно от компонентов верхнего уровня.

65. В чем разница между Smoke и Sanity тестированием?

Smoke тестирование – высокоуровневое тестирование, используемое для обеспечения правильной работы наиболее важных функций программной системы. Это быстрое тестирование, которое можно использовать для определения того, стоит ли тратить время и энергию на дальнейшее, более обширное тестирование.

Sanity тестирование – это более специфический тест, используемый для проверки того, что недавние изменения в системе не вызвали никаких новых, нежелательных проявлений. Оно гарантирует, что после внесения незначительных изменений основные функции продолжают работать так, как ожидалось.

66. В чем разница между статическим и динамическим тестированием?

Статическое тестирование – это тип тестирования, проводимый без выполнения кода программного приложения. Вместо этого оно включает в себя обзоры, инспекции и прохождение тестов.

Динамическое тестирование предполагает выполнение кода программного приложения для определения результатов определенных функций и операций. Оно включает модульное, интеграционное и приемочное тестирование.

Вопросы, основанные на реальном опыте

Вопросы для собеседования по ручному тестированию в реальных условиях основаны на настоящих проектах и сценариях, над которыми работал кандидат, и могут дать представление о его возможностях и опыте. В результате менеджеры по подбору персонала могут лучше понять навыки и знания кандидата и определить, подходит ли он на эту должность.

67. Как вы определите, когда следует прекратить тестирование?

При тестировании крайне важно определить, когда следует остановиться, чтобы не тратить ресурсы впустую. Принимая решение о прекращении тестирования, следует учитывать следующие критерии:

* Желаемый уровень качества
* Соблюдение сроков и бюджета
* Количество найденных дефектов
* Количество пройденных тест-кейсов
* Факторы риска, связанные с проектом

Как только эти критерии будут выполнены, можно прекратить тестирование.

68. Как вы протестируете продукт, если требования еще не установлены?

Когда требования еще не определены, лучше всего использовать методологию гибкой разработки, например Scrum.

Первым шагом должно стать проведение совещаний по сбору требований со всеми заинтересованными сторонами, чтобы понять цель продукта и желаемые результаты. Затем следует разбивка проекта на отдельные, управляемые пользовательские истории.

После этого определяют приоритеты пользовательских историй и назначают спринт для их разработки.

По мере продвижения проекта продукт постоянно тестируют, используя такие виды тестирования, как модульное, интеграционное, приемочное и системное. Кроме того, по мере изменения требований тесты должны обновляться, чтобы гарантировать соответствие продукта желаемым результатам.

69. В каких случаях вы выберете автоматизированное тестирование ?

* **Когда тест повторяется снова и снова.** Автоматизированное тестирование идеально подходит для выполнения тестов, требующих нескольких итераций или многократного повторения одних и тех же действий.
* **Когда тест требует большого количества данных.** Автоматизированное тестирование позволяет быстро вводить большие объемы данных в тестируемую систему.
* **Когда тест требует использования нескольких сред.** Автоматизированное тестирование можно легко настроить для тестирования систем в различных средах, таких как несколько операционных систем, браузеров и устройств.
* **Когда тестирование запланировано на ночное время.** Автоматизированные тесты могут быть запрограммированы на точное выполнение, гарантируя, что каждый шаг теста будет выполнен в точное время.
* **Когда тест требует участия нескольких пользователей.**Автоматизированное тестирование может имитировать одновременный доступ к системе нескольких пользователей, что позволяет проводить более реалистичное тестирование.

70. Что такое “управление конфигурацией”?

Управление конфигурацией – это управление, отслеживание и контроль изменений в программном, аппаратном или сетевом обеспечении системы. Все это помогает поддерживать целостность системы и гарантировать, что она безопасна, стабильна и соответствует политике организации.

Основными целями управления конфигурацией являются обеспечение надежности, поддержание доступности и повышение производительности системы.

71. Системное тестирование можно проводить на любом этапе?

Нет, системное тестирование обычно проводится в конце процесса разработки, после интеграционного и приемочного.

72. Каких практик следует придерживаться при написании тест-кейсов?

Лучшие советы для написания тест-кейсов:

* Разрабатывайте тест-кейсы, которые будут ясными, краткими и по существу.
* Убедитесь, что они проверяют функциональность программного обеспечения во всех измерениях.
* Убедитесь, что они охватывают все требования.
* Разрабатывайте повторяющиеся тест-кейсы, которые при необходимости можно автоматизировать.
* Разрабатывайте независимые друг от друга тест-кейсы.
* Используйте четкие, описательные заголовки.
* Записывайте результаты проверок для дальнейшего использования.
* Проводите их анализ для обеспечения точности и полноты проверок.
* Документируйте тест-кейсы в стандартном формате.

73. Почему анализ граничных значений удачно вписывается в тест-кейсы?

Анализ граничных значений помогает составить хорошие тест-кейсы, поскольку он гарантирует, что границы входных и выходных значений будут проверены. Такие проверки помогают выявлять ошибки, связанные с обработкой граничных условий, и гарантируют, что ваша система надежна и может обрабатывать любые неожиданные входные или выходные значения.

74. Почему нельзя протестировать программу или исключить ошибки на 100%?

Невозможно полностью проверить всю программу или избавить ее от всех возможных ошибок, потому что нельзя предусмотреть и протестировать все возможные комбинации входов, окружений и состояний, с которыми может столкнуться ПО.

75. Может ли автоматизированное тестирование заменить ручное?

Нет. Мы можем автоматизировать повторяющиеся, утомительные тест-кейсы и сделать процесс тестирования более эффективным. Однако автоматизированное тестирование не может полностью заменить ручное, поскольку некоторые тесты можно выполнить только вручную.

Виды тестирования, которые нельзя автоматизировать, – исследовательское, пользовательское, удобства использования.