**1. Назвіть тестові активності.**

1. Test Planning

2. Test monitoring and control

3. Test analysis

4. Test design

5. Test implementation

6. Test execution

7. Test completion

**2. Які є принципи тестування?**

1. Тестування демонструє наявність дефектів, але не гарантує їх відсутності

2. Вичерпне тестування неможливе

3. Раннє тестування економить час і гроші

4. Кластеризація\скупчення дефектів

5. Парадокс пестициду

6. Тестування залежить від контексту

7. Омана про відсутність помилок

**3. Хто такий хороший тестувальник і якими якостями він має володіти?**

1. Вивчення області, на яку орієнтоване ПЗ

2. Деструктивний напрямок мислення

3. Розвиток навичок пошуку різних видів помилок

4. Впевненість у тому, що продукт містить помилки

5. Удосконалення англійської мови

6. Вміння розподіляти свій час між багатьма задачами одночасно

**4. Яка різниця між верифікацією та валідацією?**

Верифікація - це процес підтвердження відповідності кінцевого продуктувимогам замовників. Відповідає на питання Чи правильно ми робимо продукт?

Валідація - це процес підтвердження відповідності кінцевого продукту потребам користувача. Відповідає на питання Чи правильний продукт ми робимо? Чи потрібен він користувачам?

**5. Опишіть будь-які 3 принципи тестування.**

1. Раннє тестування економить час і гроші. Тестування слід починати на ранніх стадіях життєвого циклу розробки ПЗ, щоб знайти дефекти якомога раніше. Чим пізніше знайдений баг, тим дорожче він коштує.

2. Парадокс пестициду. Якщо повторювати ті самі тести знову і знову, то в певний момент цей набір тестів перестане виявляти нові дефекти. Тому тест кейси постійно треба додавати і оновлювати.

3. Вичерпне тестування неможливе. Повне тестування з використанням всіх вхідних комбінацій даних, результатів і передумов фізично неможливе. Тому при створенні тестів потрібно враховувати ризики і використовувати техніки тестування.

**6. Які основні етапи SDLC?**

1. Requirement gathering and analysis (Збір та аналіз вимог)

2. Design (Проєктування)

3. Implementation or coding (Реалізація або кодування)

4. Testing (Тестування)

5. Deployment (Розгортання)

6. Maintenance (Обслуговування)

**7. Чи може тестування на Acceptance testing level виконуватись тестувальниками?**

До acceptance testing залучаються тестувальники зі сторони замовників

**8. Яка різниця між Black Box та White box?**

Black  Box - це метод тестування програмного забезпечення, за якого функціональні можливості ПЗ перевіряються без знання внутрішньої структури коду та деталей реалізації системи. Воно також відоме як зовнішнє тестування, закрите тестування або ж поведінкове тестування.

White box - це метод тестування, який перевіряє внутрішнє функціонування системи. У цьому методі тестування перевіряються внутрішня структура, будова і код програмного забезпечення. У тестуванні білого ящика код доступний тестувальникам, тому його також називають тестуванням чистого ящика, тестування відкритого ящика, тестування прозорого ящика, тестування на основі коду та тестування скляного ящика.

**9. Які відрізняють види Integration Testing Level?**

Component integration testing level - описує процес взаємодії між компонентами ПЗ.

System integration testing level - описує процес взаємодії між різними системами.

**10. Що таке проект**

Проєкт - це обмежена часовими рамками діяльність, яка має початок і кінець. У більшості випадків проєкт може бути обмежений фінансами або досягненням певних результатів.

Головна ціль проєкту - це реалізація унікальних цілей та завдань.

**11. Що являється більш пріорітетним функціональне чи нефункціональне тестування? Аргументуйте.**

Більш пріоритетним є функціональне тестування. Воно дає змогу виявити помилки, дефекти та недоліки в програмному продукті і перевірити основні його функції, правильність роботи, взаємодію різних компонентів. А нефункціональне тестування  оцінює якість продукту з точки зору аспектів, не пов'язаних із функціональністю, таких як продуктивність, безпека, зручність у використанні.

**12. Яка різниця між каскадними та ітеративно-інкрементними моделями розробки ПЗ?**

У каскадній моделі перехід на наступну фазу можливий лише після повного успішного завершення попередньої без можливості повернення.

Ітеративно-інкрементний підхід передбачає виконання робіт паралельно з неперервним аналізом отриманих результатів і можливістю коректування попередніх етапів. В кожній ітерації додаються нові вимоги та розширюється функціонал ПЗ.

**13, Основна відмінність Waterfall та V моделі?**

У V-моделі тестування починається ще зі стадії написання вимог і для кожного етапу передбачений Test Design відповідного Testing Levels. А Waterfall модель - це каскадна методологія, в якій процес розробки виглядає як потік, послідовно проходячи всі фази ПЗ.

**14, Які є види тестування пов'язаного зі змінами?**

1. Re-testing/Confirmation testing

2. Regression testing

3. Sanity testing

4. Smoke testing

**15. Яка основна ціль ретроспективи?**

Retrospective - це обговорення того, як команда спрацювала протягом спринту, можливість поділитися своїм досвідом та виділити успішні моменти. Основна ціль - визначення позитивних аспектів та пошук підвищення якості роботи в майбутньому.

**16, Що таке трасованість вимоги?**

Трасованість - це зв'язок з вимогою вище та вимогою нижче (того ж або іншого рівня). Вимоги посилаються на джерела у ранніх вимогах. Кожна вимога повинна мати унікальне ім'я та номер.

**17, Кнопки на калькуляторі мають виконувати множення,ділення,додавання. Яка характеристика хороших вимог порушена?**

Відсутня кнопка віднімання, а це порушення вимоги повноти, що впливає на функціональність калькулятора.

**18. Чим відрізняється недводумність від несуперечливості?**

Недводумність означає, що вимога має бути описана чітко, зрозуміло і коротко, а всі спеціальні терміни мають бути виписані та визначені. Тому вимоги, які інтерпретуються по-різному мають бути перероблені або видалені.

А несуперечливість - це харарактеристика якості вимог, яка передбачає, що реалізація однієї вимоги не повинна перешкоджати реалізації іншої вимоги. Вимога не повинна конфліктувати з вимогами такого ж типу або з вимогами інших рівнів.

**19. Веб сторінка має завантажуватись швидко. Яка характеристика хороших вимог порушена?**

Порушено перевірність, тому що не вказано чітко часове значення, протягом якого має завантажитися веб сторінка, тобто вимога описана дуже розмито, а відповідно її не можливо перевірити.

**20. Що таке недводумність вимог?**

Недводумність означає, що вимога має бути описана чітко, зрозуміло і коротко, а всі спеціальні терміни мають бути виписані та визначені. Тобто, вимоги, які інтерпретуються по-різному мають бути або перероблені, або видалені. Тестується найчастіше перевіркою специфікації, яка перевіряється на те, чи всі вимоги з предметної області описані та чи кожен із термінів, які зустрічаються у специфікації, відповідають своєму визначенню.

**21. Які є види тестового покриття?**

1. Покриття коду - оцінка покриття виконуваного коду тестами шляхом відстеження неперевірених у процесі тестування частин програмного забезпечення.

2.Покриття вимог - оцінка покриття тестами функціональних та нефункціональних вимог до продукту шляхом побудови матриць трасування або сучасних підходів Mind Map.

**22. Які можуть бути Suspension criteria and resumption requirements ?**

Suspension criteria можуть бути знайдені критичні дефекти, зміна вимог, блокування тестування стороннім сервісом, до якого підв'язаний функціонал нашого ПЗ, спрацювання якогось з ризиків.

Resumption criteria - виправлення критичних дефектів, опрацювання і коригування планування згідно нових вимог, вирішення проблеми блокування тестування стороннім сервісом, усунення якогось з ризиків.

**23. Різниця між тест планом та тест стратегією?**

Тест План створює Test Manager або Test Lead, а Тест Стратегію створює Project Manager.

Тест План в деталях описує об'єм тестування і дії, необхідні для проведення тестування, а Тест Стратегія - це високорівневий (не детальний) документ з інструкціями про те, як буде проходити тестування.

Тест План описує всі активності тестування: які використовувати техніки, графік, ресурси, тощо, а Тест Стратегія поверхнево описує техніки, що будуть використовуватись, систему для тестування тощо.

Тест План може бути змінений і оновлений, а Тест Стратегію зазвичай не змінюють.

В Тест Плані багато деталей та уточнень, а в Тест Стратегії загальні підходи та методології.

**24. Хто створює тест план?**

Тест План створює менеджер з тестування, QA Lead, а редагуватися він може тестувальниками.

**25. Що таке тест план?**

Тест План - це документ, який описує весь обсяг робіт з тестування, починаючи з опису об'єкта, стратегії, розкладу, критеріїв початку та закінчення тестування, використовуваного обладнання та інструментальних засобів, спеціальних знань, а також оцінка ризиків з варіантами їх вирішення. Може створюватися один на проєкт або на кожну версію.

**26. Які є підходи до Risk based testing?**

1. Швидкі тести - підхід, коли ми шукаємо типові місця для типових помилок. Наприклад, помилки граничних значень.

2. Дослідницькі посібники. Керівні слова - це список понять, які ви хочете застосувати до системи, що тестується. Спочатку ви розбиваєте свою програму на компоненти, а потім розглядаєте компоненти один за одним з точки зору керівних слів користувача.

3. Ризики на рівні проєкту. Виявлення проблем, які можуть призвести до невдачі проєкту або відставання від графіка або занадто великих витрат або відчуження ключових зацікавлених осіб. Аналіз потенційних витрат, постійна оцінка або моніторинг ризиків.

**27. Що таке Risk based testing?**

Тестування засноване на ризиках - це тестування, що використовує ризики як метод для визначення пріоритетності та виокремлення певних тестів у процесі тест дизайну.

**28. Що таке Exploratory testing?**

Дослідницьке тестування - тестування ПЗ, яке підкреслює особисту свободу та відповідальність окремого тестувальника за постійну оптимізацію цінності своєї роботи (навчання, пов'язане з тестуванням, розробка тестів, виконання тестів) протягом усього проєкту.

**29. Які переваги Exploratory testing?**

1. Проєктування тестів за потреби.

2. Виконання тестів під час проєктування або повторне використання їх пізніше.

3. Зміна тесту за потреби.

**30. Exploratory testing це повноцінний підхід чи техніка. Аргументуйте**

Це повноцінний підхід до тестування, що дозволяє виконати повне тестування систем і має свою власну структуру. Тестувальник активно контролює проєктування тестів в той час, як ці тести виконуються, і використовує отриману під час тестування інформацію для проєктування нових тестів. Цей підхід можна застосувати до будь-якого виду тестування. Дослідницьке тестування – це не тільки виконання тестів. Тестувальники можуть застосовувати дослідницький підхід і при розробці нових тестів, і при аналізі вже завершених тестів.

**31. Які є види тест кейсів (різні класифікації)?**

По очікуваному результату (expected result):

1. Позитивні (positive)

2. Негативні (negative)

За рівнем деталізації:

1. Високорівневі

2. Деталізовані

**32. Що таке тест кейс?**

Тест кейс - це тестовий документ, артефакт, що описує сукупність кроків, конкретних умов і параметрів, необхідних для перевірки реалізації функції або її частини, що тестується. Це мінімальний елемент тестування.

**33. Які є основні атрибути тест кейсу?**

Основними атрибутами тест кейсу є:

1. Номер тест кейсу (ID)

2. Назва тест кейсу (Test case summary)

3. Передумови (Pre-Conditions)

4. Кроки або послідовність дій (Steps)

5. Очікуваний результат (Expected result)

6. Пріоритет (Priority)

7. Компонент (Component)

8. Посилання на вимоги (Reference Document)

**34. Що таке чек лист?**

Чек лист - це тестовий документ, який зазвичай описує функції, які мають бути перевірені. Складається з переліку блоків, секцій, сторінок, інших елементів, які потрібно протестувати.

**35. Що таке ризик лог?**

Ризик Лог - тестовий артефакт, який відповідає за перевірку певної функціональності, може мати різний рівень деталізації та заснований на ризиках.

**36. Яка техніка тестування доповнює техніку Класів Еквівалентності?**

Техніку Класів еквівалентності доповнює техніка Аналіз граничних значень, яка допомагає перевірити помилки меж класів еквівалентності

**37. Що таке Ad-hoc тестування?**

Ad-hoc тестування або Вільне тестування - це вид тестування, який виконується без підготовки до тестування продукту, без визначення очікуваних результатів, без проєктування тестових сценаріїв. Це неформальне, імпровізаційне тестування, яке не вимагає ніякої документації, планування процесів, яких слід дотримуватися

**38. Яка різниця між статичним та динамічним тестуванням?**

Статичне тестування застосовують для верифікації практично будь-якого артефакту розробки і асоціюють з аналізом програмного забезпечення. БЕЗ ЗАПУСКУ ПРОГРАМНОГО КОДУ!

Динамічне тестування - це тестування, що виробляється над працюючою системою або підсистемою. З ЗАПУСКОМ ПРОГРАМНОГО КОДУ!

**39. Різниця між ad-hoc і exploratory testing.**

Різниця між ad-hoc і exploratory testing полягає в тому, що ad-hoc може провести будь-хто, а для проведення exploratory testing необхідна майстерність і володіння певними техніками

**40. Різниця формального та не формального рецензування?**

Формалізм залежить від кількості документації, кількості учасників рецензії, серйозності планування, кількості мітингів.

Неформальне може бути суто спілкування з колегою за чашкою кави і обговорення якогось документа чи артефакта.

Формальне - це планування рецензування завчасно, визначення ролей учасників, створення різних мітингів.

**41. Назвіть основні поля/атрибути баг репорту.**

Основними атрибутами баг репорту є:

1. ID

2. Summary (назва, короткий опис, проблеми)

3. Project - назва проєкту

4. Component - назва частини або функції, що тестується

5. Version (номер версії)

6. Severity (серйозність)

7. Priority (пріоритет)

8. Status (статус)

9. Author/Reporter (автор)

10. Assigned to (назначено на)

11. Environment (середовище)

12. Steps to Reproduce (кроки для відтворення)

13. Actual result (фактичний результат)

14. Expected result (очікуваний результат)

15. Additional (додатково)

16. Attachment (прикріплений файл)

**42. Що таке баг трекінгова система?**

Баг трекінгова система - це інструмент, який забезпечує фіксування дефектів та їх змін і навіть підтримку їх станів. Наприклад, Jira, Bugzilla, Trello.

**43. Як скласти хороше самері багу?**

Summary - це короткий опис проблеми, чітко вказуючи на причину і тип помилкової ситуації.

Для того, щоб скласти хороше summary потрібно відповісти на три питання:

ЩО? відбувається або не відбувається відповідно до специфікацій або подання тестувальника про нормальну роботу програмного продукту

ДЕ? в інтерфейсі користувача чи архітектурі ПЗ перебуває проблема

КОЛИ? в який момент роботи ПЗ, настання якої події або за яких умов проблема проявляється

**44. Хто виставляє Priority дефекту?**

Priority дефекту встановлюється менеджером, тімлідом або замовником

**45. Яка існує градації Priority дефекту?**

Priority (Пріоритет) - це атрибут, що вказує на послідовність виконання завдання або усунення дефекту. Чим вищий пріоритет, тим швидше потрібно виправити дефект.

Градації Priority:

Р1 Високий (High) - помилку необхідно виправити якнайшвидше, оскільки її наявність є критичною для проєкту

Р2 Середній (Medium) - помилку необхідно виправити, її наявність не є критичною, але вимагає обов'язкового вирішення

Р3 Низький (Low) - помилку необхідно виправити, її наявність не є критичною і не вимагає негайного вирішення

**46. Що таке BugManet?**

BugManet - розширення з відкритим вихідним кодом для Браузера, яке забезпечує легкий доступ до поширених проблемних значень тестування програмного забезпечення та граничних випадків. Можна легко отримати до них доступ під час дослідницьких сесій тестування, щоб можна було заповнити перевірені поля вибраним значенням.

**47. Що таке TestRail?**

TestRail - джерело для масштабованого веб-керування тест кейсами, тест сьютами, що настроюється.

**48. Навіщо тестувальнику інструменти тестування?**

Інструменти допомагають QA-інженерам ефективно організовувати свою роботу на кожному етапі тестування програмного забезпечення, які обирають залежно від певного виду тестування, особистих переваг і місця роботи тестувальника.

**49. Під час test design як інструменти допомагають нам?**

Під час test design інструменти допомагають нам:

- створювати тестові дані (інструменти, як електронні таблиці; генератори моделі стану; генератори випадкових чисел);

- змінювати або очищати виробничі дані з міркувань конфіденційності (перемішувачі даних, замінники імен);

- генерувати цікаві комбінації параметрів (усі пари або комбінаторні генератори даних).

**50. Під час звітності тестування як інструменти допомагають нам?**

Під час звітності тестування інструменти допомагають нам:

- записувати нашу діяльність та документувати наші процедури (інструменти для створення нотаток; інструменти відеозапису; вбудований журнал; інструменти обробки текстів; інструменти запису взаємодії з користувачем);

- готувати звіти для наших клієнтів (ментальні карти; текстові процесори; електронні таблиці; програмне забезпечення для презентацій).

**51. Яка різниця між one-tier та two-tier клієнт-серверною архітектурою?**

 One-tier клієнт-серверна архітектура - це коли бізнес-логіка, логіка даних та інтерфейс користувача знаходяться в одній системі, а two-tier клієнт-серверна архітектура - це коли база даних знаходиться на сервері, а інтерфейс користувача - на стороні клієнта.

**52. В чому відмінність між тонким та товстим клієнтом? Поясніть**

Тонкий клієнт - це легкий комп'ютер, створений для підключення до сервера з віддаленого розташування, а більшу частину роботи виконує сервер, а товстий клієнт - це комп'ютер, який зазвичай забезпечує багату функціональність незалежно від центрального сервера і характеризується здатністю виконувати багато функцій без підключення до сервера.

**53. Що таке тонкий клієнт?**

Тонкий клієнт - це легкий комп'ютер, створений для підключення до сервера з віддаленого розташування, а більшу частину роботи виконує сервер.

**54. Що таке клієнт-серверна архітектура?**

Клієнт-серверна архітектура - це мережева архітектура, в якій мережеве навантаження розподілене між постачальниками послуг, серверами, і замовниками послуг, клієнтами. Модель такої системи полягає в тому, що клієнт відправляє запит на сервер, де він обробляється і готовий результат відправляється клієнтові.

**55. Що таке мережевий протокол?**

Мережевий протокол - це набір правил, за якими відбувається взаємодія між комп'ютерами в мережі.

**56. Яке речення визначає спосіб сортування результатів.**

Для того, щоб відсортувати результати під час виконання запиту, використовується речення ORDER BY, яке розташовується в кінці конструкції SQL. Речення ORDER BY містить список полів, які потрібно використовувати для сортування у порядку, який потрібно застосовувати до операцій сортування.

**57. Що видасть цей запит? SELECT \* FROM Customers;**

Запит SELECT \* FROM Customers повертає всі дані - стовпці та рядки  з таблиці Customers.

**58. Яке речення перелічує таблиці, у яких містяться поля, зазначені в реченні SELECT.**

Речення FROM перелічує таблиці, у яких містяться поля, зазначені в реченні SELECT.

**59. Що видасть цей запит? SELECT City FROM Customers;**

Цей SQL-запит вибирає всі значення стовпця City з таблиці  Customers.

**60. Яке речення перелічує поля, які містять потрібні дані.**

Речення SELECT перелічує поля, які містять потрібні дані.

**61. Що таке complexity та velocity?**

Complexity (складність) - це оцінка обсягу роботи, яку потрібно виконати для завершення завдання.

Для оцінки складності можна використовувати одиниці часу (години) або Story Points.

Velocity (швидкість) - це міра того, скільки роботи команда може виконати за певний проміжок часу.

Зазвичай швидкість вимірюється в Story Points.

**62. Що таке master в системі контролю версій?**

Master в системі контролю версій - це назва гілки за замовчуванням, яка представляє основну версію коду, що розробляється.

**63. Цілі конфігураційного менеджмента?**

Цілі конфігураційного менеджмента:

1. Контроль - дозволяє відстежувати зміни в об'єктах, що контролюються, забезпечує дотримання процесу розробки.

2. Управління - диктує процес автоматичної ідентифікації в ході всього життєвого циклу програмного забезпечення, забезпечує простоту модифікації та супроводу ПЗ.

3. Економія засобів - знижується ризик втрат від ротації кадрів в організації, надати можливість змінити організацію-розробника без перепроектування.

4. Якість - відстеження і можливість контролювати, покращувати процес як атрибути якості.

**64. Які є техніки естімації в Agile?**

Техніки естімації в Agile глобально поділяють на Accurate та Rough.

Серед найбільш розповсюджених технік - Естимація за аналогією, Experience-Based, Three-Point estimation, Story points, T-Shirt sizes.

**65. Що таке система контролю версій?**

Система контролю версій (СКВ) - це програмний інструмент, який відстежує зміни у файлах (наприклад, у коді програми, документах, зображеннях) з часом, дозволяючи повертатися до попередніх версій та співпрацювати над проектами. Вона дозволяє бачити, хто і коли вносив зміни, а також відкочувати зміни у разі помилок.

**66. Що таке Endpoint?**

Endpoint - це кінцева точка, що представляє собою конкретну адресу (наприклад, URL), за якою клієнт може отримати доступ до ресурсу або виконати певну дію на сервері

**67. Що таке REST API?**

REST API (Representational State Transfer Application Programming Interface) - це архітектурний стиль для створення веб-сервісів, який використовує HTTP-протокол для обміну даними між клієнтом та сервером

**68. Найпоширеніші протоколи реалізації веб сервісів?**

Найпоширенішими протоколами реалізації веб-сервісів є SOAP, REST та XML-RPC.

SOAP (Simple Object Access Protocol) - це протокол обміну повідомленнями, який використовує XML для структурування даних.

REST (Representational State Transfer) - це архітектурний стиль, що базується на принципах HTTP і використовується для створення веб-сервісів.

XML-RPC (XML Remote Procedure Call) - це простий протокол, який використовує XML для виклику віддалених процедур.

**69. Що таке веб сервіс?**

Веб-сервіс (англ. web service) - ідентифікована веб-адресою програмна система зі стандартизованими інтерфейсами (взаємодіями). У побуті веб-сервісами називають послуги, що надаються в Інтернеті

**70. Що робить запит GET?**

Запит GET - це найпопулярніший тип запиту, який використовується для отримання або читання даних. Запити з використанням цього методу можуть тільки отримувати дані

**71. Які коди відповіді з класу "Успішні"**

Коди відповідей 200-299 є "Успішними".

Наприклад, якщо клієнт запросив будь-які дані, то вони знаходяться в заголовку та/або тілі повідомлення 200ОК; 201 Created - в результаті успішного виконання запиту було створено новий ресурс.

**72. Які коди відповіді з класу "Інформаційні"**

Коди відповіді з класу 100-199 є "Інформаційними"

**73. Які коди відповіді з класу "Клієнтські помилки"**

Коди відповіді з класу 400-499 - це помилки на стороні клієнта.

Наприклад, 400 - сервер виявив у запиті клієнта синтаксичну помилку; 401 - для доступу до запитуваного ресурсу потрібна автентифікація; 403 - доступ до сторінки заборонено або сервер не може його надати;  404 - помилка написання адреси Web-сторінки; 409 - запит не може бути виконаний через конфліктне звернення до ресурсу

**74. Які коди відповіді з класу "Серверні помилки"**

500-599 - це "Серверні помилки".

Наприклад, код 500 повідомляє, що на сервері сталася внутрішня непередбачена помилка або аварійна відмова.

**75. Які 5 класів кодей відповідей існує?**

Існує 5 класів кодів відповідей:

1. Інформаційні - 100-199;

2. Успішні - 200-299;

3. Перенаправлення (редірект) - 300-399;

4. Клієнтські помилки - 400-499;

5. Серверні помилки - 500-599.