1. Які критерії визначають якість вихідного коду програмного забезпечення?
2. Які інструменти аналізу вихідного коду можна використовувати для оцінки його якості?
3. Які принципи Clean Code можна застосувати для покращення якості вихідного коду?
4. Як виявити потенційні проблеми в коді, такі як антипатерни та запахи коду?
5. Які аспекти кодового стилю слід враховувати під час аналізу якості вихідного коду?
6. Як можна оцінити читабельність вихідного коду та його розуміння для інших розробників?
7. Як визначити ефективність та продуктивність вихідного коду під час аналізу його якості?
8. Які вимоги до безпеки можна врахувати під час аналізу якості вихідного коду?
9. Які стратегії можна застосувати для рефакторингу вихідного коду на основі його аналізу?
10. Які переваги може мати покращення якості вихідного коду для розробки програмного забезпечення та кінцевих користувачів?
11. Що таке тестування в реальному середовищі і чому воно важливе для розробки програмного забезпечення?
12. Які основні відмінності між тестуванням у контрольованому середовищі та тестуванням в реальному середовищі?
13. Які типи тестів можуть бути проведені в реальному середовищі? (додаток А3)
14. Як визначити потрібний рівень тестування для конкретного продукту чи функції в реальному середовищі?
15. Як впливає реальне середовище на результати тестування та які чинники можуть впливати на його ефективність?
16. Які інструменти можуть бути використані для проведення тестування в реальному середовищі?
17. Як визначити обсяг та покриття тестування для забезпечення належної якості продукту в реальних умовах експлуатації?
18. Які можливі проблеми можуть виникнути під час тестування в реальному середовищі і як їх уникнути?
19. Як організувати та планувати процес тестування в реальному середовищі для забезпечення ефективного виконання?
20. Які можливі переваги та недоліки можуть виникнути в результаті тестування в реальному середовищі порівняно з іншими методами тестування?
21. Що таке методологія Scrum та які принципи лежать в її основі?
22. Які ключові ролі і відповідальності учасників в Scrum-команді?
23. Які основні артефакти використовуються в Scrum для управління проектом?
24. Як відбуваються Scrum-події (спринти, планування спринтів, ретроспективи тощо) та яка їх роль у процесі розробки?
25. Як визначити обсяг та тривалість спринта для проекту?
26. Як здійснюється планування та прийняття змін в ході реалізації проекту за методологією Scrum?
27. Як забезпечити комунікацію та співпрацю між учасниками Scrum-команди для досягнення спільних цілей?
28. Як визначити та оцінити ризики, пов'язані з впровадженням Scrum у власній організації?
29. Які можливі перешкоди можуть виникнути під час впровадження Scrum та як їх уникнути або подолати?
30. Які переваги та виклики можуть виникнути після успішного впровадження Scrum у робочий процес команди?
31. Чому оцінка продуктивності та реакції користувачів є важливою частиною процесу розробки програмного забезпечення?
32. Які методи використовуються для вимірювання продуктивності програмного забезпечення?
33. Як визначити критерії оцінки реакції користувачів на продукт?
34. Які метрики можуть використовуватися для оцінки реакції користувачів на продукт?
35. Як здійснюється збір даних про реакцію користувачів на продукт?
36. Як аналізувати результати тестування продуктивності та реакції користувачів?
37. Які можливі проблеми можуть виникнути під час оцінки продуктивності та реакції користувачів і як їх вирішити?
38. Як зробити висновки на основі результатів оцінки продуктивності та реакції користувачів?
39. Які можливі стратегії покращення продуктивності та реакції користувачів можна розглянути на основі отриманих даних?
40. Які переваги може мати вдосконалення продуктивності та реакції користувачів для команди розробників та кінцевих користувачів?
41. Чому важливо виявлення та управління помилками під час розробки програмного забезпечення?
42. Які основні кроки включає процес управління виявленими помилками?
43. Як класифікувати виявлені помилки за їхньою серйозністю та впливом на продукт?
44. Як здійснюється відстеження та документування виявлених помилок? Як визначити пріоритетність виявлених помилок для їх подальшого виправлення?
45. Як впливають виявлені помилки на графік та обсяг робіт з розробки програмного забезпечення?
46. Як забезпечити ефективне виправлення виявлених помилок у встановлені терміни?
47. Як оцінити ефективність процесу управління виявленими помилками та зробити відповідні висновки?
48. Які можливі стратегії запобігання виявленню помилок можна врахувати при розробці програмного забезпечення?
49. Як впливає процес управління виявленими помилками на якість та користування програмним забезпеченням?
50. Що таке аудит якості коду та процесів розробки та чому він важливий для проектів з розробки програмного забезпечення?
51. Які основні цілі та завдання проведення аудиту якості коду та процесів розробки?
52. Які критерії та метрики можуть використовуватися для оцінки якості коду?
53. Які методи можна використовувати для проведення аудиту якості коду та процесів розробки?
54. Які можливі проблеми можуть бути виявлені під час аудиту якості коду та процесів розробки?
55. Які кроки потрібно вжити для виправлення виявлених проблем після проведення аудиту?
56. Як забезпечити ефективне впровадження рекомендацій, отриманих під час аудиту?
57. Як оцінюється ефективність аудиту якості коду та процесів розробки?
58. Які можливі переваги може мати виконання аудиту якості коду та процесів розробки для команди розробників та продукту?
59. Як аудит якості коду та процесів розробки впливає на загальну якість та довгостроковий успіх проекту з розробки програмного забезпечення?
60. Чому створення рекомендацій з покращення якості продукту є важливою частиною процесу розробки програмного забезпечення?
61. Які основні критерії використовуються для оцінки якості програмного продукту?
62. Які методи можна використовувати для збору відгуків користувачів про продукт?
63. Як виявити слабкі місця та проблеми в продукті за допомогою аналізу відгуків користувачів?
64. Як враховувати потреби та очікування користувачів під час розробки рекомендацій?
65. Як вибрати пріоритетні напрями покращення на основі аналізу відгуків користувачів?
66. Які можливі стратегії покращення якості продукту можна розглянути на основі отриманих даних?
67. Як забезпечити ефективну комунікацію та співпрацю між розробниками та користувачами під час розробки рекомендацій?
68. Як оцінити вплив запропонованих змін на якість та використовуваність продукту?
69. Як впливають рекомендації з покращення на користувачів та бізнес продукту?