

**Вопросы к экзамену по дисциплине
"МиСПО" 2019/2020**

1. ISO/IEC 12207:2010: Жизненный цикл ПО. Группы процессов ЖЦ.
2. Модели ЖЦ (последовательная, инкрементная, эволюционная).
3. Водопадная (каскадная) модель.
4. Методология Ройса.
5. Традиционная V-chart model J.Munson, B.Boehm.
6. Многопроходная модель (Incremental model).
7. Модель прототипирования (80-е).
8. RAD методология.
9. Спиральная модель.
10. UML Диаграммы: Структурные и поведенческие.
11. UML: Use-case модель.
12. UML: Диаграмма классов.
13. UML: Диаграмма последовательностей
14. UML: Диаграмма размещения
15. *UP методологии (90-е). RUP: основы процесса.
16. RUP: Фаза «Начало».
17. RUP: Фаза «Проектирование».
18. RUP: Фаза «Построение».
19. RUP: Фаза «Внедрение».
20. Манифест Agile (2001).
21. Scrum.
22. Disciplined Agile 2.X (2013).
23. Требования. Иерархия требований.
24. Свойства и типы требований (FURPS+).
25. Формулирование требований. Функциональные требования.
26. Требования к удобству использования и надежности.
27. Требования к производительности и поддерживаемости.
28. Атрибуты требований.
29. Описание прецедента.
30. Риски. Типы Рисков.
31. Управления рисками. Деятельности, связанные с оценкой.
32. Управления рисками. Деятельности, связанные контролем и управлением.
33. Изменение. Общая модель управления изменениями.
34. Системы контроля версий. Одновременная модификация файлов.
35. Subversion. Архитектура системы и репозиторий.
36. Subversion: Основной цикл разработчика. Команды.
37. Subversion: Конфликты. Слияние изменений.
38. GIT: Архитектура и команды.
39. GIT: Организация ветвей репозитория.
40. GIT: Плагин git-flow.
41. Системы автоматической сборки: предпосылки появления
42. Системы сборки: Make и Makefile.
43. Системы сборки: Ant. Команды Ant.
44. Системы сборки: Ant-ivy.
45. Системы сборки: Maven. POM. Репозитории и зависимости.
46. Maven: Структура проекта. GAV.
47. Maven: Зависимости. Жизненный цикл сборки. Плагины.
48. Системы сборки: Maven. POM. Репозитории и зависимости.
49. Системы сборки: GNU autotools. Создание конфигурации проекта.

50. Системы сборки: GNU autotools. Конфигурация и сборка проекта.
51. Сервера сборки/непрерывной интеграции.
52. Основные понятия тестирования. Цели тестирования.
53. Понятие полного тестового покрытия и его достижимости. Пример.
54. Статическое и динамическое тестирование.
55. Автоматизация тестов и ручное тестирование.
56. Источники данных для тестирования. Роли и деятельности в тестировании.
57. Понятие тестового случая и сценария.
58. Выбор тестового покрытия и количества тестов. Анализ эквивалентности.
59. Модульное тестирование. Junit 4.
60. Интеграционное тестирование. Стратегии интеграции.
61. Функциональное тестирование. Selenium.
62. Техники статического тестирования. Статический анализ кода.
63. Тестирование системы в целом. Системное тестирование. Тестирование производительности.
64. Тестирование системы в целом. Альфа- и бета-тестирование.
65. Аспекты быстродействия системы. Влияние средств измерения на результаты.
66. Ключевые характеристики производительности.
67. Нисходящий метод поиска узких мест.
68. Пирамида памяти и ее влияние на производительность.
69. Мониторинг производительности: процессы.
70. Мониторинг производительности: виртуальная память.
71. Мониторинг производительности: буферизированный файловый ввод-вывод.
72. Мониторинг производительности: Windows и Linux.
73. Системный анализ Linux "за 60 секунд".
74. Создание тестовой нагрузки и нагрузчики.
75. Профилирование приложений. Основные подходы.
76. Компромиссы (trade-offs) в производительности.
77. Рецепты повышения производительности при высоком %SYS.
78. Рецепты повышения производительности при высоком %IO wait.
79. Рецепты повышения производительности при высоком %Idle.
80. Рецепты повышения производительности при высоком %User.