

ВОПРОСЫ К ВЕБИНАРУ:

- 1) Дайте понятие системы контроля версий. Приведите примеры систем контроля версий.
- 2) Опишите разницу между полным и инкрементальным копированием в системах контроля версий
- 3) Опишите возможности систем управления версиями
- 4) Особенности и недостатки работы локальных систем контроля версий на примере системы RCS.
- 5) Особенности и недостатки работы централизованных систем контроля версий на примере систем CVS, Subversion и Perforce.
- 6) Особенности и недостатки работы распределенных систем контроля версий на примере системы Git.
- 7) Дайте определение понятию «репозиторий». Локальные и удаленные репозитории.
- 8) Для каких целей используются частные(приватные) и открытые репозитории
- 9) Ведение версий в системах контроля версий (возможности по работе с локальным и удаленным репозиторием; версии и фиксации; ветки и работа с ними; слияния и разрешения конфликтов; варианты отката изменений; теги и комментирование; возможности по просмотру истории изменений).
- 10) Способ хранения данных в системе Git. Способ отслеживания целостности данных.
- 11) Состояния файлов в СКВ Git.
- 12) Описание каталогов в СКВ Git.
- 13) Реализация ветвления в Git. Использование ветки master. Указатель HEAD.
- 14) Долгоживущие и тематические ветки. Удаленные ветки.
- 15) Перемещение веток. Опишите процесс слияния (merge) веток.
- 16) Перемещение веток. Опишите процесс слияния (rebase) веток.
- 17) Разница между интерпретируемыми и компилируемыми языками программирования. Компиляция интерпретирующего типа.

- 18) Понятие сборки программного обеспечения. Процессы, входящие в сборку ПО.
- 19) Утилита make – задачи, описание. Содержимое make – файлов.
- 20) Стандартные цели для сборки дистрибутивов GNU.
- 21) Определение автоматизации сборки. Действия, которые включает в себя систем автоматизации.
- 22) Виды автоматизации, применяемые в различных инструментах
- 23) Система сборки Gradle. Назначение, область применения, аналоги Gradle, достоинства и недостатки Gradle. Сравнение с аналогами.
- 24) Стандартные задачи системы сборки
- 25) Фазы сборки программного обеспечения
- 26) Понятие жизненного цикла программного обеспечения. Модели жизненного цикла программного обеспечения
- 27) Определение каскадной модели жизненного цикла. Ее преимущества и недостатки. Область применения.
- 28) Определение инкрементной модели жизненного цикла. Ее преимущества и недостатки. Область применения.
- 29) Определение V модели жизненного цикла. Ее преимущества и недостатки. Область применения.
- 30) Определение модели жизненного цикла на основе прототипа. Ее преимущества и недостатки. Область применения.
- 31) Определение спиральной модели жизненного цикла. Ее преимущества и недостатки. Область применения.
- 32) Документирование знаний при разработке программного обеспечения. Системы Confluence, MediaWiki.
- 33) Понятие концепции CI/CD.
- 34) Основные принципы CI/CD.
- 35) Организация управления требованиями в CI/CD
- 36) Конвейер развертывания. Основные этапы и задачи. Настройка конвейера развертывания.
- 37) Понятие среды окружения. Объекты среды окружения.
- 38) Основные переменные среды окружения ОС Linux.

39) Понятие системного окружения (яруса) в концепции методологии CI/CD.

40) Архитектуры сред окружения. Основные понятия, назначение, решаемые задачи.

41) Определение понятия «виртуализация». Задачи виртуализации. Типы виртуализации.

42) Определение понятия «виртуализация». Преимущества виртуализации.

43) Определение понятия «виртуализация». Недостатки виртуализации.

44) Виртуализация операционной системы. Задачи, применение.

45) Виртуализация серверов приложений. Задачи, применение.

46) Виртуализация сети. Задачи, применение.

47) Виртуализация аппаратного обеспечения. Задачи, применение.

48) Виртуализация систем хранения. Задачи, применение.

49) Виртуализация сервисов. Задачи, применение.

50) Решения виртуализации. Примеры решений виртуализации

51) Программная виртуализация. Задачи программной виртуализации.

Преимущества и недостатки программной виртуализации. Приведите примеры программной виртуализации.

52) Виртуализация ИТ-инфраструктуры. Задачи, преимущества и недостатки виртуализация ИТ-инфраструктуры, область ее применения.

53) Контейнерная виртуализация. Задачи, применение.

54) Применение контейнеризации в современных вычислительных системах.

55) Контейнеры приложений.

56) Классификация контейнеров. Терминология Linux контейнеров

57) Оркестрация контейнеров. Цели и задачи.

58) Образ контейнера. Определение. Два состояния контейнера.

59) Контейнерный движок. Задачи контейнерного движка.

60) Контейнерный хост. Определение. Задачи.

61) Сервер реестра. Определение. Задачи.

- 62) Определение системы докер. Докер-демон и докер – клиент.
- 63) Определение системы докер. Докер-образ и докер – реестр.
- 64) Пространство имен в системе докер. Основные пространства имен.
- 65) Конфигурационное управление. Цели и задачи конфигурационного управления.
- 66) Конфигурационное управление. Процедуры управления конфигурацией.
- 67) Методология DevOps. Основные положения.
- 68) Непрерывная доставка. Определение. Основные элементы.
- 69) Непрерывная интеграция. Определение. Основные элементы.
- 70) Основы микросервисной архитектуры. Определение. Принципы проектирования.
- 71) Основы микросервисной архитектуры. Определение. Преимущества и недостатки.
- 72) Технология разработки программного обеспечения. Основные понятия, область применения. Отличие технологии разработки программного обеспечения от программной инженерии.
- 73) Фаза разработки программных приложений. Этапы фазы разработки программного обеспечения.
- 74) Коллективная разработка программных приложений. Особенности коллективной разработки.
- 75) Основные правила оформления программного кода. Приведите примеры правильного и неправильного оформления кода.
- 76) Рефакторинг программного кода. Масштабируемый рефакторинг программного кода. Основные цели, задачи и приемы рефакторинга.
- 77) Комментирование программного кода. Основные правила комментирования.
- 78) Комментирование объектов баз данных, предназначенных для хранения данных. Основные правила их комментирования.
- 79) Современные сервисы для организации коллективной разработки программных приложений. Приведите примеры подобных сервисов.

80) Понятие версии программного продукта и системы контроля версий. Модели версионирования и их сравнение.

81) Последовательность и состав действий при работе с локальным и удаленным хранилищем системы контроля версий.

82) Система Subversion и ее архитектура. Хранилище системы, его структура, правки. Сценарий объединения правок. Конфликты и способы их разрешения.

83) Пакеты и менеджеры пакетов ОС Linux. Основные понятия, состав, назначение, решаемые задачи. Приведите примеры пакетов и менеджеров пакетов

84) Зависимости пакетов и управление зависимостями менеджерами пакетов. Основные понятия, назначение, решаемые задачи. Приведите примеры управления зависимостями.

85) Управление пакетами в ОС Linux. Приведите примеры управления пакетами в ОС Linux.

86) Непрерывная интеграция CI и экстремальное программирование. Организация, преимущества и недостатки непрерывной интеграции. Приведите примеры непрерывной интеграции.

87) Возможности GitLab CI для непрерывной интеграции. Приведите примеры применения GitLab CI для непрерывной интеграции.

88) Развёртывание программного обеспечения. Основные понятия, назначение, решаемые задачи. Приведите примеры развёртывания программного обеспечения.

89) Окружения развёртывания программного обеспечения. Ярусы. Приведите примеры окружений и ярусов.

90) Действия при развёртывании программного обеспечения. Подготовка программных приложений к развёртыванию

91) Технологии и решения развёртывания программного обеспечения. Приведите примеры современных систем развёртывания программного обеспечения.

92) Типовые ошибки развёртывания программного обеспечения. Их причины и проявление этих ошибок. Приведите примеры.

93) Поясните принцип работы командного процессора и приведите пример для ОС Linux

94) Система управления конфигурациями Ansible. Назначение, краткая характеристика системы. Приведите примеры применения системы.

95) Архитектура Ansible. Преимущества системы управления конфигурациями Ansible. Модули системы. Приведите примеры применения Ansible.

96) Решения по управлению конфигурацией компании «Систематика - консалтинг».

97) Сценарии (playbooks) Ansible. Основные понятия, назначение, решаемые задачи. Приведите примеры сценариев

98) Опишите особенности графических и консольных клиентов на примере Git.

99) Дайте характеристику трём этапам процесса: создание, состояние и завершение

100) Системы управления конфигурацией. Инструменты управления конфигурациями. Приведите примеры современных систем управления конфигурацией.

101) Конфигурация программного обеспечения. Основные понятия, состав, назначение, решаемые задачи. Приведите примеры конфигурации программного обеспечения.

102) Распределенная сборка. Определение, назначение, цели, типовые действия при распределенной сборке. Приведите примеры распределенной сборки.