

Перечень вопросов к экзамену

Кафедра «Системы обработки информации и управления»

по дисциплине «Технологии разработки программного обеспечения»

1. Стратегии разработки ПО и модели жизненного цикла. Каскадная модель, спиральная модель и инкрементная стратегии разработки ПО. Перечень, последовательность и содержимое этапов разработки по методологиям. Привести примеры итераций, задач, макетов и моделей.
2. Стратегии разработки ПО и модели жизненного цикла. Макетирование и модель быстрой разработки RAD. Перечень, последовательность и содержимое этапов разработки по методологиям. Примеры итераций, задач, макетов и моделей.
3. Тяжеловесные и облегченные процессы разработки ПО. Экстремальное программирование (XP-процесс). Перечень, последовательность и содержимое этапов разработки по методологиям. Примеры итераций, задач, макетов и моделей.
4. Стратегии разработки ПО и модели жизненного цикла. Agile (Scrum) процесс и компонентно-ориентированная модель. Перечень, последовательность и содержимое этапов разработки по методологиям. Примеры итераций, задач, макетов и моделей.
5. Руководство программным проектом. Планирование проекта. Структурная декомпозиция работ. Ресурсы и риски. Распределение задач и ресурсов. Вехи. Привести определения и примеры.
6. Руководство программным проектом. Планирование проекта. Анализ и управление рисками. Типы рисков. Календари ресурсов. Привести определения и примеры.
7. Оценка бюджета проекта: экспертная, по аналогии, алгоритмическая, на основе графика выполнения. Точность оценки. Оценка затрат, длительности и стоимости разработки. Метрический базис. Привести определения и примеры.
8. Оценка затрат, длительности и стоимости разработки. Оценка размера проекта на основе LOC, FP или OP. Привести определения и примеры.
9. Оценка затрат, длительности и стоимости разработки. Размерно-ориентированные и функционально-ориентированные метрики, их соотношение. Привести определения и примеры.
10. Вычисление затрат, длительности и стоимости разработки на основе моделей COSOMO. Модели композиции приложения, раннего этапа проектирования и этапа пост-архитектуры. Привести определения и примеры.
11. Оценка затрат, длительности и стоимости разработки. Анализ чувствительности программного проекта с применением модели COSOMO. Цель и методы выполнения. Привести определения и примеры.
12. Проектирование программных систем. Декомпозиция подсистем на модули. Модуль и модульность. Виды связности модуля. Виды сцепление модулей. Привести определения и примеры.
13. Оценка качества программных систем. Функциональные метрики программных систем. Сложность модуля. Сложность программной системы.

Иерархическая структура. Метрики высоты, ширины, невязки. Привести определения и примеры.

14. Оценка качества программных систем. Метрики объектно-ориентированных систем: Чидамбера и Кемерера, Лоренца и Кидда, Абреу. Их назначение и методы вычислений. Отличие от функциональных метрик. Привести определения и примеры.

15. Унифицированный процесс разработки ПО (RUP). Перечень, последовательность и содержимое этапов разработки. Рабочие процессы и их модели. Примеры итераций, задач, макетов и моделей для RUP.

16. Унифицированный процесс RUP. Определение архитектурно-значимых прецедентов. Модель прецедентов, модель анализа (пакеты, классы и кооперации), модель проектирования (диаграммы развертывания, подсистемы, их интерфейсы и активные классы) и модель реализации (подсистемы и компоненты). Трассировка между элементами моделей. Привести определения и примеры моделей.

17. Унифицированный процесс RUP. Рабочий процесс Определение требований. Кандидаты в требования. Контекст системы. Функциональные и нефункциональные требования. Модели прецедентов (требований) и предметной области. Нахождение актеров и прецедентов. Определение приоритетов и детализация прецедентов. Создание прототипа пользовательского интерфейса. Структурирование модели прецедентов. Привести определения и примеры моделей.

18. Унифицированный процесс RUP. Рабочий процесс Анализ. Модель анализа (пакеты, их зависимости и состав, диаграммы коопераций анализа). Анализ архитектуры, прецедентов, классов и пакетов (деятельность и модели). Привести определения и примеры моделей.

19. Унифицированный процесс RUP. Рабочий процесс Проектирование. Модель проектирования (подсистемы всех уровней, их состав, интерфейсы и зависимости, диаграмма развертывания и активные классы, примеры диаграмм взаимодействия подсистем). Проектирование архитектуры, прецедентов, классов и подсистем (деятельность и модели). Привести определения и примеры моделей.

20. Унифицированный процесс RUP. Рабочий процесс Реализация. Модели реализации (компоненты, интерфейсы и подсистемы реализации, стереотипы и зависимости файлов). Реализация архитектуры, подсистем, классов (деятельность и модели). Сборка системы. Тестирование модулей. Привести определения и примеры моделей.

21. Унифицированный процесс RUP. Рабочий процесс Тестирование (деятельность и модели). Процедуры, методы и варианты тестирования. Привести определения и примеры.

22. Проектирование программных систем. Модели архитектуры: хранилище данных, клиент-сервер, трехуровневая модель, многоуровневая модель, каналы и фильтры, компонентная и каркасы. Привести определения и примеры.

23. Проектирование программных систем. Структурирование системы. Определение моделей архитектуры, управления и интерфейсов взаимодействия. Привести определения и примеры.

24. Проектирование программных систем. Модели управления: вызов-возврат, менеджер, широковещательная и с прерываниями. Привести определения и примеры.
25. Структурный проект. Выделение уровней. Управление зависимостями. Устранение циклической зависимости. Зависимости наследования, уровней, методов. Делегирование. Зависимости реализации и интерфейсов. Привести определения и примеры иллюстрирующих диаграмм.
26. Структурный проект. Интерфейсы предоставляемые и требуемые. Доминирующий класс. Использование интерфейсов для устранения циклических зависимостей. Отделенный интерфейс. Привести примеры иллюстрирующих диаграмм классов и взаимодействия.
27. Структурный проект. Синхронные и асинхронные сообщения. Организация подписки. Знакомства. Пакет знакомств. Привести примеры иллюстрирующих диаграмм классов и взаимодействия.
28. Проектирование программных систем. Интерфейсы взаимодействия на основе: API, иерархии классов, форматов данных и компонентных технологий. Привести определения и примеры.
29. Структурные шаблоны. MVC: подсистемы, их назначение и схема взаимодействия. Требования и особенности. Реализация обратного вызова. Привести примеры иллюстрирующих диаграмм классов и взаимодействия.
30. Структурные шаблоны. PCMEF: подсистемы, их назначение и схема взаимодействия. Требования и особенности. Делегирование. Привести примеры иллюстрирующих диаграмм классов и взаимодействия.
31. Структурные шаблоны. PCMEF+: подсистемы, их назначение и схема взаимодействия. Требования и особенности. Привести примеры иллюстрирующих диаграмм классов и взаимодействия.
32. Паттерны проектирования. Назначение и принцип описания. Типы и виды паттернов (бизнес-логика, БД, Gof). Параметры паттернов. Привести определения и примеры.
33. Паттерны для реализации бизнес-логики. Сценарий транзакций и модель предметной области. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
34. Паттерны для реализации бизнес-логики. Модуль таблицы и слой служб. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
35. Паттерны для работы с БД. Шлюз таблицы и шлюз записи. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
36. Паттерны для работы с БД. Активная запись и преобразователь данных. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
37. Паттерны GoF. Порождающие, структурные и поведенческие. Абстрактная фабрика и фасад. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.

38. Паттерны GoF. Порождающие, структурные и поведенческие. Строитель и посредник. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
39. Паттерны GoF. Порождающие, структурные и поведенческие. Одиночка и стратегия. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
40. Паттерны GoF. Порождающие, структурные и поведенческие. Компоновщик и шаблонный метод. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
41. Паттерны GoF. Порождающие, структурные и поведенческие. Мост и итератор. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
42. Паттерны GoF. Порождающие, структурные и поведенческие. Приспособленец и наблюдатель. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия.
43. Паттерны GoF. Порождающие, структурные и поведенческие. Декоратор и посетитель. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
44. Паттерны GoF. Порождающие, структурные и поведенческие. Заместитель и состояние. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
45. Паттерны GoF. Порождающие, структурные и поведенческие. Прототип и хранитель. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
46. Паттерны GoF. Порождающие, структурные и поведенческие. Паттерны классов и объектов. Адаптер и цепочка обязанностей. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
47. Паттерны GoF. Порождающие, структурные и поведенческие. Фабричный метод и команда. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.
48. Тестирование ПО. Восходящее и нисходящее тестирование. Драйверы и заглушки. Автоматизация тестирования. Тестовый вариант. Привести определения и примеры.
49. Тестирование ПО. Виды тестирования: модульное, интеграционное, регрессионное и системное. Виды системного тестирования. Привести определения и примеры.
50. Тестирование ПО. Виды тестирования. Тестирование базового пути. Тестирование условий. Примеры и определения. Привести определения и примеры.
51. Тестирование ПО. Виды тестирования. Тестирование данных. Тестирование циклов. Привести примеры и определения.
52. Тестирование ПО. Виды тестирования. Разбиение по классам эквивалентности. Анализ граничных значений. Привести примеры и определения.

53. Тестирование ПО. Назначение и цели. Тестовый вариант. Критерии разработки и оценки качества. Функциональное и структурное тестирование. Особенности и возможности. Привести примеры и определения.
54. Автоматизированные средства управления проектами, их назначение и возможности. Создание графика, версий и итераций; определение сюжетов и задач; планирование времени и ресурсов; отслеживание состояния проекта; взаимодействие участников проекта, их роли и возможности. Последовательность и состав действий при управлении проектом. Привести примеры и определения.
55. Автоматизированные средства разработки ПО. Построение моделей требований, анализа, проектирования и реализации (виды моделей и используемые нотации); прямое и обратное проектирование; проверка моделей и согласование моделей; использование профилей и стереотипов. Последовательность и состав действий при разработке ПО. Привести примеры и определения.
56. Автоматизированные средства тестирования ПО. Виды автоматического тестирования: модульное, нагрузочное, веб-серверное, пользовательского интерфейса; задание исходных параметров тестирования, результаты тестов и их интерпретация; покрытие кода. Привести примеры и определения.
57. Автоматизированные средства ведения 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. Последовательность и состав действий при работе с локальным и удаленным хранилищем. Привести примеры и определения.
58. Автоматизированные средства создания документации. Принципы и возможности автоматического составления документации; возможности по комментированию исходных кодов: проекты, модули, файлы, классы и их компоненты; параметры комментариев; настройки генератора документации. Привести примеры и определения.