

Экзамен по курсу Технологии разработки программного обеспечения 2020/2021 г.

Билет содержит два вопроса. По каждому вопросу необходимо написать теорию и привести примеры для конкретной темы (тема выдается отдельно).

Темы вопросов:

Методологии и стратегии разработки ПО и модели жизненного цикла:

- каскадная модель,
- спиральная модель,
- инкрементная модель,
- макетирование,
- модель быстрой разработки RAD,
- экстремальное программирование (XP-процесс),
- компонентно-ориентированная модель,
- тяжеловесные (прогнозирующие) процессы разработки,
- адаптивные (облегченные) процессы разработки,
- Agile (Scrum) процесс,
- унифицированный процесс (RUP).

Перечень, последовательность и содержимое этапов разработки по методологиям.

Примеры итераций, задач, макетов и моделей.

Руководство программным проектом. Планирование проекта. Структурная декомпозиция работ.

Распределение задач и ресурсов. Вехи.

Ресурсы и риски. Анализ и управление рисками. Календари ресурсов.

Оценка затрат на разработку. Точность оценки.

Оценка затрат, длительности и стоимости разработки. Метрический базис.

Оценка размера проекта на основе LOC, FP или OP.

Размерно-ориентированные и функционально-ориентированные метрики, их соотношение.

Вычисление затрат, длительности и стоимости разработки на основе моделей COCOMO –II.

Модели композиции приложения, раннего этапа проектирования и этапа пост-архитектуры.

Анализ чувствительности программного проекта с применением COCOMO –II.

Унифицированный процесс RUP. Этапы и итерации. Цели, задачи и деятельность этапов.

Вехи. Управление рисками.

Рабочие процессы и их модели:

- Определение требований (деятельность и модели). 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.
- Анализ. Модель анализа (пакеты, их зависимости и состав, диаграммы коопераций анализа). Анализ архитектуры, прецедентов, классов и пакетов (деятельность и модели). Стереотипы классов анализа.
- 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).
- Реализация. Модели реализации (компоненты, интерфейсы и подсистемы реализации, стереотипы и зависимости файлов). Реализация архитектуры, подсистем, классов (деятельность и модели). Сборка системы. Тестирование модулей.
- Тестирование (деятельность и модели). Процедуры, методы и варианты тестирования.

Трассировка между элементами моделей различных рабочих процессов.

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.

Проектирование программных систем. Переход от моделей анализа к моделям проектирования. Структурирование системы. Модели архитектуры:

- хранилище данных,
- клиент-сервер,
- трехуровневая модель,
- многоуровневая модель,
- каналы и фильтры,
- компонентная,
- каркасы.

Модели управления:

- вызов-возврат,
- менеджер,
- широковещательная,
- с прерываниями.

Интерфейсы взаимодействия на основе:

- API,
- иерархии классов,
- форматов данных,
- компонентных технологий.

Структурный проект. Выделение уровней. Управление зависимостями. Устранение циклической зависимости. Зависимости наследования, уровней, методов. Делегирование. Зависимости реализации и интерфейсов.

Интерфейсы предоставляемые и требуемые. Доминирующий класс.

Использование интерфейсов для устранения циклических зависимостей. Отделенный интерфейс. (Примеры диаграмм классов и взаимодействия)

Синхронные и асинхронные сообщения. Организация подписки. Знакомства. Пакет знакомств. (Примеры диаграмм классов и взаимодействия)

Структурные шаблоны:

- MVC,
- PCMEF,
- PCMEF+.

Подсистемы шаблонов, их назначение и схема взаимодействия. Требования и особенности. (Примеры диаграмм классов и взаимодействия)

Паттерны проектирования. Назначение и принцип описания. Параметры.

Паттерны для реализации бизнес-логики (Примеры диаграмм классов и взаимодействия):

- сценарий транзакций,
- модель предметной области,
- модуль таблицы,
- слой служб.

Назначение, особенности реализации и область применения.

Паттерны для работы с БД (Примеры диаграмм классов и взаимодействия):

- шлюз таблицы,
- шлюз записи,
- активная запись,
- преобразователь данных.

Назначение, особенности реализации и область применения.

Паттерны GoF:

- Порождающие: абстрактная фабрика, строитель, одиночка, фабричный метод, прототип;
- Структурные: фасад, адаптер, компоновщик, мост, приспособленец, декоратор, заместитель;
- Поведенческие: хранитель, стратегия, команда, состояние, шаблонный метод, итератор, посетитель, посредник, цепочка обязанностей, наблюдатель.

Паттерны классов и объектов. Примеры диаграмм классов и взаимодействия для паттернов GoF. Назначение, особенности реализации и область применения.

Тестирование ПО. Назначение и цели. Тестовый вариант. Критерии разработки и оценки качества.

Функциональное и структурное тестирование. Особенности и возможности.
Тестирование базового пути. Тестирование условий.
Тестирование данных. Тестирование циклов.
Разбиение по классам эквивалентности. Анализ граничных значений.

Виды тестирования: модульное, интеграционное, регрессионное и системное. Виды системного тестирования.

Восходящее и нисходящее тестирование. Драйверы и заглушки. Автоматизация тестирования.

Автоматизированные средства разработки ПО, их назначение и возможности:

- управление проектами (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.
- разработка 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.
- тестирование ПО (виды автоматического тестирования: модульное, нагрузочное, веб-серверное, пользовательского интерфейса; задание исходных параметров тестирования, результаты тестов и их интерпретация; покрытие кода).
- ведение 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.
- Создание документации (принципы и возможности автоматического составления документации; возможности по комментированию исходных кодов: проекты, модули, файлы, классы и их компоненты; параметры комментариев; настройки генератора документации).

Литература

- Орлов С.А. Технологии разработки программного обеспечения. - СПб.: Питер. - 2002 г. (ссылку см. на сайте кафедры).
- Унифицированный процесс разработки программного обеспечения: учебное пособие / Виноградова М.В., Белоусова В.И. – М.: МГТУ им.Н.Э. Баумана. – 2015 г. <http://ebooks.bmstu.ru/catalog/193/book1303.html>
- Фаулер М. Архитектура корпоративных программных приложений. – М.: Издательский дом Вильямс. – 2008г.
- Гамма Э. и др. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.:Питер. - 2009 г.

Дополнительная

- Дубина О. Обзор паттернов проектирования. - Ciforum, 2005 г. - Доступ <http://citforum.ru/SE/project/pattern/>
- Гудлиф П. Ремесло программиста. -Спб.: Символ-плюс. - 2009 г.
- Мацяшек Л.А., Лионг Б.Л. Практическая программная инженерия. - М.: Бином. - 2009 г.
- Якобсон А, Дуч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. - СПб.: Питер. - 2002 г.