Задание на курсовое проектирование

по дисциплине Технологии разработки программного обеспечения

**Цель работы:**

Изучить теоретические принципы унифицированного процесса разработки СОИУ и составляющих его этапов. Получить практические навыки применения шаблонов при проектировании и разработке СОИУ. Освоить применение CASE средств для разработки СОИУ.

**Исходные данные для выполнения КР:**

* Описание исходных требований к создаваемой СОИУ (выдается преподавателем).
* Набор паттернов для реализации (выдается преподавателем).
* Пример проектирования СОИУ при унифицированном процессе [1].
* Описание унифицированного процесса и его этапов [1,2].
* Пример использования CASE -средств для проектирования [1, стр. 266-280].
* Описание паттернов проектирования Gof [4].
* Описание шаблонов разработки [5].
* Свободно-распространяемое CASE, например, StarUML[3] или Software IdeasModeller [6].

**Порядок и время проведения работы:**

Работа выполняется группой студентов от 1 до 3 человек в часы внеаудиторных занятий. Каждый студент выполняет свою часть проекта. По итогам выполнения КР составляется и защищается отчет в бумажном виде, а также проводится демонстрация работающей программы.

**Задание:**

Выполнить проектирование СОИУ в соответствии с описанием ее функциональности (определяется вариантом). Для проектирования использовать этапы и модели унифицированного процесса. По результатам проектирования получить работающую программу с паттернами (по варианту). Для построения диаграмм использовать среду, например, STARUml или Software Ideas Modeller. Ниже перечислены этапы проектирования и их задачи, выполняемые в КР:

1. Этап анализа и планирования требований (Начало) – **Выполняет студент 1:**

* Определить область применения СОИУ (предназначение, границы, интерфейсы с внешней средой, критерии сдачи-приемки).
* Определить прецеденты, критические для системы (основные функции и главные решения).
* Определить основные элементы архитектуры (для выполнения основного сценария).
* Определить и оценить самые опасные риски (угрожающие успеху разработки).
* Оценить затраты и время выполнения проекта.

**В отчет:**

* Постановка задачи (задание по варианту).
* Спецификация основных проектных требований, ключевых характеристик и главных ограничений (словесное описание требований-кандидатов, функциональных и нефункциональных требований).
* Описание бизнес-процессов (диаграмма бизнес-прецедентов или объектная бизнес-модель).
* Модель предметной области (диаграмма классов предметной области и глоссарий понятий).
* Выявленные актеры.
* Выявленные прецеденты.
* Диаграмма основных прецедентов.
* Описание (словесное) прецедентов.
* Перечень критических рисков (и рекомендации по управлению).
* Список приоритетов прецедентов.
* Описание возможной архитектуры (перечень архитектурно-значимых прецедентов, обобщенные механизмы проектирования, системное ПО, используемые компоненты и/или каркасы). По каждой из выбранных технологий (обобщенные механизмы проектирования, системное ПО и т.д.) необходимо привести варианты, их достоинства и недостатки и обоснование выбора.
* Начальная оценка стоимости, затрат и длительности проекта (COCOMO II этапа композиции приложения на основе объектных указателей).
* Начальный план выпусков версий.

1. Этап проектирования (Развитие):

**Выполняет студент 2:**

* Уточнить и структурировать функциональные требования в виде прецедентов (выделить актеров и прецеденты; создать прототип пользовательского интерфейса; определить приоритеты, детализировать и структурировать прецеденты) – по основным прецедентам.
* Выполнить анализ (анализ архитектуры - выделить пакеты анализа и сервисные пакеты, определить классы сущностей и общие специальные требования; анализ прецедентов — определить классы анализа и их взаимодействие; анализ классов — определить ответственности, атрибуты и связи; анализ пакетов).
* Спроектировать базовый уровень архитектуры (определить узлы и сетевые конфигурации, подсистемы и интерфейсы между ними, архитектурно-значимые и активные классы, обобщенные механизмы проектирования).
* Уточнить затраты и время выполнения проекта.

**Выполняет студент 3:**

* Определить используемые шаблоны проектирования (по варианту), в том числе:
  + структурный шаблон (MVC, PCMEF и т.д.),
  + шаблон(ы) организации бизнес-логики (сценарий транзакций, модуль таблицы и т.д.),
  + шаблон(ы) работы с БД (шлюз записи данных, активная запись и т.д.),
  + шаблон(ы) организации пользовательского интерфейса (Gof и др.).
* Реализовать базовый уровень архитектуры на основе модели проектирования.
* Оценить качество проекта.
* Отслеживать риски, устранить наиболее серьезные.
* Уточнить затраты и время выполнения проекта.
* Составить план итераций следующего этапа.

**Программная реализация - Выполняет студент 3 (по своему варианту):**

* Работающая программа, реализующая основные прецеденты (базовый уровень архитектуры) – простейшее решение.

**В отчет:**

**Выполняет студент 2:**

* Расширенные описания прецедентов (предусловие, запуск и инициатор, порядок выполнения и поток событий, постусловия, запрещенные и альтернативные пути, атрибуты и ресурсы) – по основным прецедентам.
* Диаграммы деятельности для основных прецедентов.
* Структурированная диаграмма(ы) прецедентов.
* Прототип пользовательского интерфейса.
* Классы анализа (ответственности, атрибуты, специальные требования) для основных коопераций (реализаций прецедентов).
* Диаграмма(ы) классов анализа.
* Диаграмма(ы) взаимодействия (классов анализа) для основных прецедентов.
* Пакеты анализа и сервисные пакеты (и их зависимости).
* Уточненная оценка стоимости, затрат и длительности проекта (на основе функциональных указателей).

**Выполняет студент 3:**

* Архитектура: Диаграмма развертывания (узлы и линии связи, подсистемы и интерфейсы, активные классы).
* Архитектура: Подсистемы, их уровни и интерфейсы (и зависимости). Выбор структурного шаблона (варианты, их достоинства и недостатки, обоснованный выбор).
* Архитектура: Обобщенные механизмы проектирования. Выбор и обоснование шаблонов организации бизнес-логики, работы с БД, пользовательского интерфейса (варианты, их достоинства и недостатки, обоснованный выбор).
* Архитектура: Диаграмма(ы) взаимодействия подсистем для архитектурно-значимых прецедентов (с учетом выбранных шаблонов).
* Начальная оценка качества проекта (WMC, NOC — для каждого класса и по всей системе, DIT, NOM, NC (количество классов) — по всей системе).
* Диаграммы компонентов и развертывания для реализации архитектуры.
* Исходный код и исполняемая программа, реализующая фрагмент архитектурно-значимых прецедентов.
* Переработанный список рисков.
* Уточненная оценка стоимости, затрат и длительности проекта (COCOMO II пост-архитектуры на основе вычисленных ранее функциональных указателей).
* Перечень и состав итераций следующего этапа.

1. Этап построения (Конструирование):

Этап построения содержит несколько итераций. Каждая итерация реализует определенный ранее набор прецедентов (сценариев) – выполняется всеми студентами (каждый по своей подсистеме):

**Каждый студент из группы выполняет:**

* Полностью определить и реализовать прецеденты (по варианту).
* Завершить анализ, проектирование, реализацию и тестирование своей подсистемы.
* Полностью разработать компоненты с использованием паттернов (по вариантам).
* Оценить качество проекта.
* Провести тестирование и отладку.

**Программная реализация - выполняет каждый студент из группы:**

* Работающая программа, реализующая подсистему (по варианту).
* Программа должна содержать реализацию паттернов проектирования (по варианту).

**В отчет по каждой итерации - выполняет каждый студент из группы:**

* Сценарии/прецеденты итерации.
* Диаграмма(ы) классов подсистемы,
* Диаграммы последовательностей для основных функций программы,
* Диаграммы классов и последовательностей для иллюстрации примененных паттернов.
* Оценка качества проекта (метрики WMC, NOC, CBO, RFC, LCOM, CS, NOO, NOA, SI, OSavg, Npavg - для каждого класса и средние по системе, DIT, NOM, NC, LOCsum — по всей системе)

**В отчет по этапу**

**Выполняет студент 3:**

* Уточненные подсистемы, их интерфейсы, содержимое и зависимости.
* Уточненная диаграмма компонентов.
* Уточненная диаграмма развертывания.
* Оценка качества проекта (метрики Абреу).

**Выполняет студент 1:**

* Тестовые варианты (структурные, функциональные, нагрузочные).

1. Этап внедрения (Переход) – **выполняет студент 2**:

* Применить ПО в среде заказчика.
* Завершить реализацию продукта.

**В отчет:**

* Перечень программ и рекомендации по установке.
* Перечень документации для пользователей и заказчиков.
* Рекомендации по внедрению (консультации, обучение и т.д.).

**Литература:**

1. Орлов С.А. Технологии разработки программного обеспечения. - СПб: Питер, - 2002 г. (Материалы сервера [http://wmate.ru](http://wmate.ru/) )
2. Якобсон А, Дуч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. - Спб.: Питер. - 2002 г.
3. Материалы сервера [http://staruml.sourceforge.net](http://staruml.sourceforge.net/)
4. Гамма Э. и др. Приемы объектно-ориентированного проектирования. Паттерны проектирования. - Спб.:Питер. - 2009 г.
5. Фаулер М. Архитектура корпоративных приложений. - М.:Изд.дом Вильямс. - 2008 г.
6. Материалы сервера <http://www.softwareideas.net/>