

Паспорт курсового проекта

по дисциплине «Технология разработки программного обеспечения», 2 семестр

1. Методика оценки

Курсовой проект выполняется в форме разработки пакета документов и прототипа программной системы по одному из 4 вариантов (см. выбор темы). Пояснительная записка должна содержать:

- Материал по разделам проектирования (см. ниже).
- Выводы
- Список литературы

При защите курсового проекта оцениваются следующие характеристики работы:

- Объем и полнота разработки перечисленных разделов
- Оформление текста пояснительной записки, логика, стиль и грамотность изложения

Выбор темы

1. Может быть выбрана в соответствии с темой магистерской диссертации. Не обязательно это компонента, напрямую входящая в работу. Можно описать проект, имеющий отношение к данной области, или решающий вспомогательную задачу. Например, стенд для моделирования основного функционала проекта и т.п..
2. Тема может быть выбрана из заданий к л.р. семестра 1. При этом бригада может выполнять одну тему, но должны быть выделены компоненты или функционал, например, «Система продажи театральных билетов. Серверная компонента». Тогда часть материала по проектированию может быть общей.
3. Тема может быть аналогично типовой, но предложена самим студентом, например, из выполненных им индивидуальных проектов или проектов, в которых он принимал участие.
4. Для индивидуальной темы объем и содержание разделов может быть изменено по согласованию с преподавателем, например, **полный бизнес-анализ за счет исключения конструирования.**

Разработка должна быть доведена до уровня прототипа: каркас проекта – основные интерфейсы, абстрактные классы, производные классы без функционального наполнения + БД с минимальным (тестовым) наполнением + реализация одного-двух существенных прецедентов (объем проекта порядка 10 пакетов и 100 классов)

Раздел 1. Бизнес-анализ и моделирование предметной области (анализ предметной области)

Исходные данные к проекту. Постановка задачи. История вопроса. Описание окружения. Основные требования (неформальное изложение)

1.1 Анализ предметной области (аналитический обзор).

Первичный сбор материалов (литература, интернет-источники) по всем этапам разработки, ключевым моментам, используемым средствам, существующим решениям. Классификация материала по темам: **существующие аналоги (прототипы), средства разработки, библиотеки, стандарты, проблемы, графический интерфейс, алгоритмы** (в пределах 10 рубрик по 5-15 ссылок в каждой).

Анализ существующих решений и прототипов: выбор классификационных признаков, например, виды поддерживаемого функционала, наличие открытого кода, свободное/коммерческое ПО, объем разработки, год выпуска, стоимость. Краткая аннотация по каждому прототипу. Сводная таблица характеристик. Определение характеристик собственной разработки («экологическая ниша», преимущества).

Сюда же: анализ средств разработки, готовых частных решений, фреймворков по той же схеме: аннотированный обзор, классификационные признаки, сводная таблица характеристик, обоснование выбора. Анализ должен производиться с точки зрения основных требований со стороны системы разработки (например, платформенная независимость, совместимость).

1.2 Системная инженерия (системный анализ)

Краткое описание системы и ее окружения. Список заинтересованных лиц (стейкхолдеров), в т.ч. разработчики, службы сопровождения и эксплуатации, заказчик, пользователи, руководитель ВКРБ, секретарь ГЭК (приемка), «злоумышленники». Описание их интересов в системе и целей, из которых следуют эти интересы. Определение основных **архитектурных видов (аспектов) и архитектурных точек зрения** (см. 6.1. Системная инженерия в папке «Программная инженерия»).

1.3 Глоссарий

Содержательная (не масло масляное) описание **наиболее важных** элементов функционала, архитектуры, реализации, особых или жаргонных терминов, используемых в разработке.

1.4 Бизнес-анализ

- 1. бизнес-требования** – обоснование полезности проекта, особенности проекта, обеспечивающие его привлекательность, предполагаемые отличия от аналогов, проблемы предметной области и способы их решения в проекте, возможности коммерческого использования, способы монетизации.
- 2. границы проекта** – перечень бизнес-процессов, поддерживаемых и не поддерживаемых системой.

3. **перечень пользователей проекта**, их квалификация, типовые задачи, которые ими решаются, специфические требования
4. словесное описание бизнес-процессов или использование диаграмм BPMN (см. папку «Бизнес-анализ» в программной инженерии)

1.5 Видение проекта

Основные данные фазы исследования, включенные в видение:

1. название проекта
2. цели проекта
3. результаты проекта
4. допущения и ограничения
5. ключевые участники и заинтересованные стороны
6. ресурсы проекта
7. сроки
8. риски
9. критерии приемки
10. обоснование полезности проекта

1.6 Моделирование предметной области

1. Модель предметной области как таковой (если проект встраивается в существующие бизнес-процессы)
2. Представление предметной области в программной системе
3. Классы бизнес-объектов на этапе проектирования

Раздел 2. Системная аналитика и архитектурное проектирование (структурное проектирование)

2.1 Анализ требований

1. Диаграммы прецедентов
2. Ключевые сценарии
3. Иерархический список требований, в т.ч. вносимых на более поздних этапах проектирования (GUI, архитектура, системные требования, форматы)

2.2 Графический интерфейс

1. Подробная диаграмма экранов, количественные данные в таблице (ссылки, активные элементы (списки, кнопки, поля, события))
2. Графический дизайн
3. Подсказки, справки в программе, ориентация на различных пользователей, программная документация
4. Обсуждение вопросов поиска, навигации, визуализации данных.

2.3. Архитектура

1. Ключевые вопросы архитектурного решения (основные идеи)
2. Компоненты архитектуры: слои, интерфейсы, протоколы, сервисы, подсистемы
3. Конкретные решения, паттерны, стили, фреймворки (и реализуемые ими перечисленные выше компоненты)
4. стандарты представления и обмена данными (форматы, протоколы)
5. Описание поведения в наиболее существенных или критических состояниях или сценариях.

Описание архитектуры согласно определения:

6. **значимые решения** по поводу организации ПС
7. **структурные элементы и их интерфейсы**, при помощи которых компонуется система
8. **поведение** - взаимодействие между этими элементами
9. **компоновка элементов** в иерархию подсистем
10. **стиль архитектуры** который направляет эту организацию

Краткое описание специфики архитектуры с каждой из сторон ее представления (модель 4+1): функциональное, логическое, процедурное, реализация, развертывание.

Раздел 3. Детальное проектирование и конструирование (функциональное проектирование)

3.1 Детальное проектирование

Архитектурные решения по перечисленному списку ключевых моментов проектирования:

1. ключевые идеи и концепции: абстрагирование, связность и соединение, модульность и декомпозиция, инкапсуляция, разделение интерфейса и реализации
2. параллелизм
3. контроль и обработка событий
4. распределение компонентов
5. обработка ошибок и исключительных ситуаций и обеспечение отказоустойчивости;
6. взаимодействие и представление (Model-View-Controller - MVC)
7. сохраняемость данных (доступность «долгоживущих» данных)
8. виды представления архитектуры - структурное, поведенческое, логическое, физическое, реализация кода
9. архитектурные стили
10. шаблоны проектирования
11. методы проектирования: нисходящее проектирование, модульное, абстрагирование, итеративность, функционально-ориентированное или структурное проектирование, объектно-ориентированное проектирование, проектирование на основе структур данных, компонентное проектирование

Архитектурные классы и диаграммы классов: архитектурные слои: представление, контроллеры, бизнес-модель, DAO (ORM), БД, коммуникации (сеть, протоколы), валидность, исключения.

Основные документы по проектированию:

1. Структура БД
2. ORM- или DAO-объекты
3. Основные бизнес-объекты
4. Форматы файлов (сохранение данных, импорт/экспорт, конфигурирование и настройка)
5. Описание протоколов: базовый протокол (web-сокеты, http, tcp/ip), форматы сообщений, процедуры обмена, синхронный/асинхронный, симметричный/несимметричный, буферизация и мультиплексирование данных, параллелизм, синхронизация.
6. Форматы собственных API проекта
7. Диаграммы состояний – автоматные модели для описания поведения сущностей системы, состояния протокола и т.п..

3.2 Конструирование

Прототип программного проекта в среде разработки в виде набора классов-пустышек. Пакеты: базовые интерфейсы и сервисы, паттерны (параллелизм, исключения, валидность данных), файлы и сериализация, коммуникация к БД, DAO (ORM-классы), бизнес-модель, контроллеры, представление (View), коммуникации с сетью (линейные и протокольные компоненты). Определяются абстракции (абстрактные БД и интерфейсы), а также классы-наследники с соответствующим наполнением. Класс содержит основные данные (свойства), а также заголовки наиболее существенных методов. Ориентировочно в прототипе должно быть порядка 50-70 классов и интерфейсов и порядка 10 пакетов.

По прототипу должна быть справка: количество пакетов, классов, свойств, методов. Эти данные используются при оценке проекта. Данные можно представить в табличной форме

Раздел 4. Тестирование, сопровождение, управление проектом

4.1 Тестирование

1. Модульное тестирование (примеры модульных тестов, возможно в табличной форме)
2. Интеграционное тестирование (описание тестовых сценариев, тестовых наборов, результаты)
3. Специфическое тестирование (например, производительности)

4.2. Документация по установке и использованию

1. Руководство по сборке проекта
2. Руководство по установке, конфигурированию, настройке, в т.ч. описание конфигурационных и настроечных файлов и процедур

3. Краткое руководство пользованием, примеры базовых сценариев

4.3. Управление проектом

1. Оценка трудоемкости и сроков исполнения проекта по COCOMO и по «смете». Во втором случае из проектной документации берутся данные о количестве компонент разработки (таблиц БД, полей, запросов, окон, событий) и берется **примерная** оценка трудоемкости на единицу
2. Список исполнителей проекта, иерархическая структура работ, расписание

Выходные документы

1. Пояснительная записка к КП
2. Исходный код проекта
3. Презентация

2. Критерии оценки

- проект считается **невыполненным**, если отсутствуют необходимые разделы описания программы, либо программа не работает, либо студент не в состоянии объяснить принципов ее работы (алгоритма) – оценка составляет <50% баллов;
- проект засчитывается на **пороговом** уровне, разработка не полностью соответствует заданию, имеются ошибки при тестировании, пояснительная записка оформлена со значительными структурными, стилистическими и грамматическими ошибками - оценка составляет не более 50-65% баллов
- проект засчитывается на **базовом** уровне, если функционал разработки соответствует заданию, отсутствуют ошибки тестирования, пояснительная записка оформлена в целом грамотно - оценка составляет 65-85% баллов
- проект засчитывается на **продвинутом** уровне, если функционал реализован полностью и эффективно, в пояснительной записке отражены все аспекты структурного, функционального проектирования и тестирования - оценка составляет 85-100% баллов

3. Шкала оценки

В общей оценке по дисциплине баллы за проект учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Перечень тем курсового проекта

1. **Система продажи билетов в кинотеатре** (клиент, кассир, билетер, директор, планирование сеансов, резервирование мест кассиром, бронирование через Интернет, финансовые отчеты, план зала, анонсы).
2. **Система продажи театральных билетов.** Приложение кассира (множество точек продажи), приложение распространителя, бронирование по интернет, связь с платежными системами, план зала (мета-уровень описания), репертуар, анонсы.
3. **Система автоматизации диспетчерской службы такси** (диспетчер, водитель, клиент, директор, прием заказов, ведение очередей, ручное распределение заказов, приложение водителя)

4. **Система автоматизированного заказа такси через Интернет** (серверное приложение для автоматического распределения заказов с учетом нагрузки, web-приложение и мобильное приложение для клиентов, приложение водителя, приложение администратора для форсмажорных и конфликтных ситуаций)
5. **Система ведения корпоративной адресной базы для мобильных клиентов** (хранение список контактов, обмен контактами, доп. данные по контактам, иерархическая многомерная общая адресная книга).
6. **Мессенджер мобильных клиентов** аналогичный WhatsUp. Регистрация по номеру мобильного телефона. Передача сообщений, файлов, синхронизация адресных книг, поиск абонента по общей адресной книге.
7. **Мессенджер с прямой связью клиентов.** Сервер контактов, регистрация, авторизация с сообщением текущего IP-адреса, прямая связь с передачей сообщений и файлов.
8. **Система заказных грузоперевозок по городу** (клиент, диспетчер, магазин, водитель, прием и оформление заказов, отчеты и сопроводительные документы, распределение заказов диспетчером,).
9. **Система продажи билетов на междугородные автобусы.** Планирование рейсов, визуализация раскладки, платежные терминалы, бронирование через Интернет, сводные отчеты по маршруту, дате.
10. **Система мониторинга междугородных автобусных перевозок** (Планирование рейсов, GPS-навигация транспортных средств, отслеживание графика движения по маршруту, обработка аварийных ситуаций).
11. **Система планирования междугородных транспортных перевозок.** Транспортные средства (тип, тоннаж, вместимость), БД населенных пунктов, дорожной сети, прием заявок, поиск подходящего транспорта с учетом его местонахождения и порожнего прогона, оформление и проводка заявок, отчеты по периоду времени, транспортному средству, водителю.
12. **Система мониторинга междугородных транспортных перевозок** (Планирование поездок, GPS-навигация транспортных средств, отслеживание графика движения по маршруту, обработка аварийных ситуаций, отслеживание заправки, отчеты по расходу топлива и рабочему времени водителей).
13. **Система мониторинга и анализа дорожной обстановки** и движения городского транспорта на основе данных сайта Новосибирскгортранса. (Периодический опрос сервера, сбор данных о состоянии транспортных средств на линии, статистические отчеты по маршруту, временному промежутку – день недели, часы, интервалам движения).
14. **Логистическая система интернет-магазина** с пунктами выдачи и доставкой по городу. Прием заказов, оформление заявок поставщикам, уведомление клиентов, отслеживание работы курьеров, отчеты по пунктам выдачи).
15. **Система авансовых платежей за услуги ЖКХ** (жилец – управляющая компания – поставщик, ведение лицевых счетов жильцов, управляющих компаний и поставщиков (эл. энергия, тепло, водоснабжение)), ввод показаний, списывание денежных средств по факту потребленных услуг с учетом текущего тарифа, подсчет общедомовых расходов.
16. **Система электронного документооборота учебного процесса** в ВУЗе (приказы о зачислении/отчислении, назначении тем дипломов и магистерских, распределение на практику, распоряжения по подразделениям, прохождение приказа: создание, визирование, утверждение, нумерация, рассылка. Мета-уровень описания структуры).

17. **Система книговыдачи библиотеки с использованием QR-кодов.** Систематический и алфавитный каталог книг с учетом экземпляров, генератор QR-кодов, ведение списка читателей, выдача и прием, ведение формуляра (с историей), уведомление о просрочках.
18. **Система книговыдачи библиотеки с использованием штрих-кодов.** Систематический и алфавитный каталог книг с учетом экземпляров, генератор штрих-кодов, ведение списка читателей, выдача и прием, ведение формуляра (с историей), уведомление о просрочках
19. **Система управления кафе/баром** (электронное меню, заказ, расчет, касса, клиент, официант, администратор, интеграция с «1С-Предприятие», электронные платежи, чек).
20. **Система бронирования мест для клубных мероприятий.** План концертов. Анонсы. Стоимость столиков. План зала (мета-уровень описания, настройка под конкретный клуб). Электронная предоплата. Бронирование. Приложение кассира. Мобильное приложение.
21. **Система бронирования мест в гостинице.** Мета-уровень описания конкретной гостиницы – расположение и типы номеров, поэтажные планы, список услуг, фото общие и отдельных номеров, расценки. Бронирование через интернет, визуализация свободных/занятых, расчет стоимости, квитанции, отчеты по периодам.
22. **Система планирования взаимосвязанных работ.** Учет сотрудников и их занятости запланированными работами. Определение работ в виде цепочек заданий с параллельными ветвями. Распределение заданий по работникам с учетом их занятости. Сдвиг сроков зависимых работ при задержке выполнения одной из них. Приложения сотрудника, руководителя подразделения, генерация отчетов.
23. **Система мониторинга обслуживания по заявкам.** Система с предварительным сбором заявок и обслуживанием (врач, электрик, сантехник...). Составление плана исполнения заявок, отслеживание времени исполнения, коррекция времени при задержках с уведомлением клиентов, отказы. Мобильный клиент сотрудника, приложение диспетчера.
24. **Система поддержки технологии SCRUM для удаленной работы.** Разработка основных элементов технологии SCRUM (митинги, беклоги, диаграммы сгорания, контроль исполнения, покер планирование) для разработчиков в режиме удаленной работы.
25. **Система многомерной организации документов.** В качестве документов могут выступать короткие заметки (записи в БД), файлы документов, изображений, звуковых. Поддерживается дерево редактирования версий файла. Создается произвольное количество систем классификаций (каталогов ссылок на файлы) по набору ключевых параметров (например, изображения могут классифицироваться по размеру, цветовой палитре, содержанию, наличию определенных предметов, звуковые файлы по исполнителю, стилю, наличию муз. инструментов).

5. Перечень вопросов к защите курсового проекта

Защита курсового проекта состоит в верификации и тестировании документов и прототипа. С этой целью преподаватель задает вопросы по исполнению отдельных сценариев в проекте разработки, оценивая возможные нестыковки и обнаруживаемые ошибки. Аналогично проверяется полнота проектирования. При необходимости проект

может быть отправлен на доработку, которая оформляется в виде отдельного раздела к пояснительной записке.