

Методические рекомендации
по выполнению курсового проекта по предмету
«Технология разработки программного обеспечения»
для студентов специальности «ПО»

Автор: Арбатский Е.В.

Курс «Технология разработки программного обеспечения» (ТРПО) рассчитан на то, чтобы студенты изучили основные вопросы связанные с проектированием и разработкой современного программного обеспечения (ПО), имели представление об основных моделях данных и управления, знали принципы управления программными проектами.

В рамках этого курса студенты на практике осваивают проектирование и разработку программного продукта в команде.

Разбивка на команды и закрепление тем.....	1
График работ.....	2
Результаты работы.....	2
Документация по проекту.....	3
План управления программным проектом (SPMP).....	3
Спецификация требований к программному обеспечению (SRS).....	4
Программные продукты.....	5
Сбор информации.....	6
Организация совместной работы.....	6

Разбивка на команды и закрепление тем

Весь контингент студентов разбивается на равные команды по 4-5 человек. Требования к команде: обязательно наличие представителей каждого пола. Исходя из состава студентов рекомендуемый минимальный состав команды был : 2(м) + 2(ж). Обычно используется следующая комбинация: 3(м)+2(ж). В некоторых случаях контингент студентов разбит на несколько подгрупп несбалансированных по полу, тогда следует применять индивидуальные комбинации разбивки. Разбитие студентов по половому признаку производится по следующим причинам:

- Отсутствие начальной информации о способностях каждого из студентов, и разбиение по этому признаку позволяет организовать соизмеримые по способностям команды;
- Потребность в том, чтобы студенты освоили особенности совместной работы в команде с представителями другого пола.

Из предложенного списка тем команды делают выбор. Одна тема выдается только на одну команду. Исключением являются темы, в которых возможна реализация разных алгоритмов и различных технологий. Темы при этом должны формулироваться таким образом, чтобы их реализация была возможна силами одной команды.

График работ

В рамках выполнения всего комплекса работ и с учетом отсутствия практических занятий рекомендуемый график работ выглядит следующим образом:

1. Первоначальная подготовка. Сбор информации, создание плана работы над проектом и проработка технического задания (последняя неделя первого месяца).
2. Определение этапов разработки и ответственных (первая неделя второго месяца).
3. Разработка и тестирование (второй месяц).
4. Создание первого варианта продукта (последняя неделя второго месяца).
5. Дополнительный анализ и доопределение требований (первая неделя третьего месяца).
6. Определение этапов разработки и ответственных (середина третьего месяца).
7. Разработка и тестирование (третий-четвертый месяц).
8. Создание конечного варианта продукта (последняя неделя четвертого месяца).

Результаты работы

Результатом работы над курсовым проектом является готовый программный продукт, который реализует заранее определенную функциональность, согласно имеющейся документации. Отчет, сдаваемый по результатам работы имеет следующую структуру:

- Титульный лист;
 - Задание;
 - Содержание;
 - Введение;
 - Описание предметной области;
 - Проектирование;
 - Разработка;
 - Тестирование;
 - Заключение;
 - Список используемой литературы;
 - Приложения (SPMP,SRS, руководства).
2. Текст работы должен быть напечатан на листах формата А4. Размеры полей: левое - 30 мм, правое - 10 мм, верхнее и нижнее - 20 мм. Шрифт 12 типа Times или Arial. Интервал между строками - полтора интервала.
 3. Титульный лист не нумеруется. Все последующие страницы начиная со второй нумеруются арабскими цифрами в правом нижнем углу листа без всяких дополнительных знаков.
 4. Все рисунки и схемы должны нумероваться и иметь название. Номер и название должны быть под рисунком или схемой. Схемы нумеруются отдельно от рисунков.
 5. В содержание следует указать порядковый номер страниц всех разделов и подразделов работы, заключение, списка используемой литературы, приложения (приложений).
 6. Заголовки разделов печатаются ПРОПИСНЫМИ буквами, заголовки подразделов - строчными. Перед заголовком пишется номер раздела или подраздела. После заголовка точка не ставится.
 7. Расстояние между заголовком и текстом - 2 интервала. Каждый раздел начинается с новой страницы.

В зависимости от проекта руководство пользователя может дополняться или заменяться

руководством администратора, разработчика.

Документация по проекту

В рамках работы над курсовым проектом для освоения навыков планирования, анализа, документирования, проектирования требуется подготовить следующие документы:

1. План управления программным проектом (SPMP);
2. Спецификация требований к программному обеспечению (SRS);
3. Руководство пользователя / администратора / разработчика.

Наименование третьего документа определяется исходя из сущности и направленности проекта.

План управления программным проектом (SPMP)

1. Введение
 - 1.1. Обзор проекта
 - 1.2. Результирующие артефакты проекта (ТЗ, макет с документацией, готовый продукт с документацией и сроки)
 - 1.3. Развитие SPMP (указывает кто отвечает за внесение изменений в этот план)
 - 1.4. Ссылочные материалы (Ссылки на все необходимые материалы для проекта должны быть перечислены в этом пункте)
 - 1.5. Аббревиатуры
2. Организация проекта
 - 2.1. Модель процесса (Указать с помощью какого процесса разработки будет делать какая версия / макет приложения)
 - 2.2. Организационная структура (Описание ролей: лидер команды, ответственный за конфигурацию, качество, проектирование, реализацию, программирование. Системный аналитик, программист, документалист, дизайнер)
 - 2.3. Организационные рамки и взаимосвязи (Указать с кем будет взаимодействовать команда)
 - 2.4. Ответственность за проект (Указать кто за какую часть проекта несет ответственность)
3. Управляющий процесс
 - 3.1. Цели и приоритеты (Установить приоритеты – максимум функций, надежности, привлекательности)
 - 3.2. Допущения, зависимости и ограничения (Указать какие допущения могут быть сделаны в случае каких-либо внешних событий)
 - 3.3. Управление рисками (Указать какие неприятности могут возникнуть и как их возможно будет разрешить)
 - 3.4. Механизмы мониторинга и контроля (Указывается время проведения совещаний, которое легко отменить, но трудно всех собрать в нужный момент спонтанно. Желательно назначить кто о чем будет докладывать.)
 - 3.5. План расстановки кадров (Указать кто отвечает за резервирование и инспектирование проекта)
4. Технический процесс
 - 4.1. Методы, инструменты и технологии (Указать какое ПО будет использовать для

- документирования, резервирования, разработки, визуализации)
- 4.2. Документация программного обеспечения
- 4.3. Функции сопровождения проекта (в рамках курсового это не будет нужно)
- 5. Распределение работ, план-график и бюджет
 - 5.1. Распределение работ (Указывает кто и что будет выполнять. В начале работы это не заполняется, а заполняется лишь по мере появления информации)
 - 5.2. Зависимости (Указывается зависимость этапов работ друг от друга. Например: разработка интерфейса возможна лишь после определения списка пользователей и их функций)
 - 5.3. Потребности в ресурсах (Указать что требуется для реализации проекта. Например: бюджет, оборудование)
 - 5.4. Выделение бюджета и ресурсов (Определяется и указывается фактический расход бюджета и использование оборудования. Либо указывает оценка расходов в виде интервала (мин.-макс.))
 - 5.5. План-график (Указывает график выполнения этапов работы над проектом)
 - 5.6. Дополнения

Спецификация требований к программному обеспечению (SRS)

- 1. Введение
 - 1.1. Цель (цель проекта)
 - 1.2. Область применения
 - 1.3. Термины и сокращения
 - 1.4. Ссылки (ссылки на документы и материалы)
 - 1.5. Обзор
- 2. Общее описание (описать проект в общем виде)
 - 2.1. Перспективы продукта (сравнение с альтернативными вариантами, варианты развития продукта)
 - a) Концепции операций (общее представление о приложении, например в виде перечня возможных состояний системы (Инициализация, настройка и тому подобное))
 - b) Концепции пользовательского интерфейса (Наброски пользовательского интерфейса)
 - Концепции зоны в пользовательском интерфейсе
 - Концепция пользовательского интерфейса для установки значений характеристик
 - c) Аппаратные интерфейсы (особые виды аппаратуры, используемой для продукта)
 - d) Программные интерфейсы (для совместной работы с другими продуктами)
 - e) Коммуникационные интерфейсы (для коммуникации и передачи данных)
 - f) Ограничения по памяти
 - g) Операции (перечень требуемых стандартных операций – например сохранение и восстановление данных)
 - h) Требования по адаптации (требования выполнения на конкретном языке, его версии или оборудовании)
 - 2.2. Функции продукта (описывается общая функциональность продукта)
 - a) Вариант использования “Функция 1”
 - b) Вариант использования “Функция 2”
 - 2.3. Пользовательские характеристики (Описать какие люди будут типичными пользователями продукта)

- 2.4. Ограничения (все условия, которые могут ограничить возможности разработчика и использования продукта)
- 2.5. Предположения и зависимости (любые предположения)
- 2.6. Распределения требований (определить в каком порядке должны выполняться требования)
3. Детальные требования
 - 3.1. Требования к внешнему интерфейсу
 - а) Пользовательские интерфейсы (принципы навигации, представления данных и тому подобное)
 - б) Аппаратные интерфейсы (описание оборудования на котором или с которым будет работать программа)
 - в) Программные интерфейсы (другие приложения с которым она должна взаимодействовать, например: драйвера какого-либо оборудования)
 - г) Коммуникационные интерфейсы (особенности коммуникационных функций программы – скорость обмена данными, формат обмена)
 - 3.2. Детальные требования
 - а) Диаграммы последовательности
 - Вариант использования “Операция 1”
 - б) Классы для классификации детальных требований
 - Требования по классам
 - 3.3. Требования к производительности (скорость, временные ограничения, память, жесткий диск и т.д.)
 - 3.4. Ограничения проектирования (дополнительные ограничения, не входящие в пред. разделы)
 - 3.5. Атрибуты программной системы
 - а) Надежность (количество сбоев на тысячу запусков)
 - б) Доступность (откуда и когда может быть доступна, с каких ОС)
 - в) Защита (по паролю, от прослушивания трафика и так далее)
 - г) Поддержка (определяется насколько сложно должно быть внесение изменений и каким образом это должно происходить)
 - Изменение части
 - Глобальные изменения
 - 3.6. Дополнительные требования
4. Дополнительная информация
 - 4.1. Оглавление и индекс
 - 4.2. Приложения

Программные продукты

В работе над курсовым проектом требуется обязательное использование системы управления проектами для освоения принципов командной работы, планирования. В рамках текущей дисциплины используется онлайн-система TUTOS (<http://www.irgups.ru/web-edu/~eugine/tutos/> , <http://web-edu.iriit/~eugine/tutos/>) , которая позволяет:

- Распределять задачи между разработчиками;
- Хранить документацию по проекту;

- Вести учет замечаний, ошибок.

Всем студентам выдается индивидуальная учетная запись для доступа к TUTOS. Если выбранная тема курсовой работы связана с другими проектами, которые делаются другими командами, либо делались в прежние года, то студентам предоставляется доступ на чтение к таким проектам.

Сбор информации

При выборе темы курсовой работы командам предоставляется общая информация о том, что должно получиться в результате. На основе этой информации и темы работы команде следует определить источники информации. Этими источниками может быть:

- Преподаватель;
- Сотрудники вуза;
- Студенты;
- Библиотека;
- Интернет.

Собрав необходимую информацию каждой команде надо понять какой именно программный продукт им требуется разработать. Результаты первоначального анализа должны быть представлены в виде:

- SPMP (По возможности заполняются все пункты);
- SRS (Заполняются разделы 1,2,4);
- Описания предметной области (Если формата SPMP и SRS недостаточно для представления всех особенностей предметной области, то следует ее описать в текстовом виде в произвольной форме).

Организация совместной работы

Основными проблемами при работе в команде у студентов является:

- Распределение обязанностей;
- Контроль выполнения работ.

В работе над курсовым проектом изначально следует выбрать лидера, который будет принимать окончательное решение в спорных моментах. В случае возникновения разлада и невозможности выбрать лидера, который бы устраивал каждого члена команды, следует разрешать спорные моменты с помощью преподавателя.

После решения вопроса с лидерством следует распределить задачи в работе над проектом. Если есть разногласия и часть команды старается отделиться, тогда следует разделить работу над проектом на малозависимые части. В этом случае нарушение сроков в работе нескольких членов команды не скажет на всем проекте.

Для совместной работы над исходным кодом возможно использовать системы управления кодом (для студентов может быть предоставлен доступ к внешней CVS). Если команде

требуется такая система, то следует обратиться к преподавателю.