Групповой проект по технологии производства программного обеспечения

Назначение документа:

Методические указания для студентов-разработчиков.

Разработан:

Д. Ж. Корзун, доцент каф. ИМО, к.ф.-м.н.
под редакцией зав. кафедрой ИМО, доцента, к.т.н. Ю. А. Богоявленского

Занятия:

весенний семестр

1 Цель дисциплины

В рамках данной учебной дисциплины студенты приобретают, осваивают и закрепляют базовые знания и навыки по технологии производства программного обеспечения (ПО) в соответствии с российскими и международными стандартами и практикой. На примере проекта, приближенного к реальному, затрагиваются особенности процесса создания и эксплуатации ПО, а также вопросы использования моделей, методов и CASE-средств. Содержание дисциплины входит в необходимый минимум профессиональных знаний выпускников по специальностям, связанных с разработкой ПО, а также дает важную дополнительную составляющую для квалифицированного использования знаний ряда дисциплин специализации, выполнения курсовых, бакалаврских и дипломных работ, а также магистерских диссертаций.

2 Основные требования

1.Командная работа. Формирование команды происходит заранее — в конце осеннего семестра. Команда самостоятельно выбирает менеджера, секретаря, модель внутренней организации, модель процесса разработки и способы внутреннего взаимодействия. Основная форма командной работы — собрания разработчиков. Выбранные правила, регулирующие командную работу, фиксируются в плане проекта.

2.Индивидуальная работа. Разработчики сочетают командную работу с индивидуальной. В процессе разработки каждому из разработчиков должны поручаться конкретные задачи для решения. Зоны ответственности, роли разработчиков и другие правила, регулирующие ин-

дивидуальную работу и ее сочетание с командной, фиксируются в плане проекта. Текущее распределение задач отражается в протоколах собраний и индивидуальных журналах разработчиков.

3.Регулярность. Разработка начинается в первую неделю семестра (см. примерный график работ ниже). В процессе разработки команда ведет журнал проекта (еженедельное заполнение). Каждый разработчик также ведет индивидуальный журнал своей работы (см. шаблоны документов).

4.Внешнее взаимодействие и отчетность. Команда еженедельно проводит официальные собрания, на которых присутствует инструктор, закрепленный за проектом. Инструктор отслеживает ход выполнения работ и оценивает качество процесса с точки зрения организации и технологии разработки. Команда предоставляет инструктору все необходимые материалы заранее. Собрания документируются в виде протоколов собраний. В собраниях также могут участвовать представители заказчика и куратор учебной дисциплины. Правила внешнего взаимодействия и отчетности согласовываются с заказчиком и инструктором и фиксируются в плане проекта.

5.Документация. В рамках проекта команда производит документацию, согласно представленным шаблонам. В некоторых случаях структура документации может быть изменена с учетом специфики проекта. Эти изменения должны быть согласованы с инструктором и, возможно, с заказчиком и куратором.

3 Организация

Команда разработчиков состоит из 4–6 студентов. Разработчики выполняют собственно разработку программного продукта в рамках выбранного проекта.

Заказчик определяет тематику проекта и требования к программному продукту. В процессе разработки заказчик консультирует команду по вопросам предметной области и требований. По окончании проекта заказчик выполняет аттестацию проекта. Заказчик не принимает участие собственно в разработке проекта (разработка решений проектирования, кодирование, тестирование, создание документации проекта и т.п.).

За каждым проектом закреплен инструктор. Он отслеживает выполнение организационно-технологических требований производства ПО в проектах. Инструктор консультирует разработчиков и заказчиков при возникновении у них вопросов, связанных с выполнением проекта. Он также участвует в

аттестации проектов и определяет окончательные индивидуальные оценки студентов на основе еженедельных оценок в течение всего проекта и соображений заказчика и куратора. Встречи инструктора с командой разработчиков происходят один раз в неделю в рамках официальных собраний разработчиков. Инструктор не принимает участие собственно в разработке проекта (разработка спецификации, решений проектирования, кодирование, тестирование, создание документации проекта и т.п.).

Общее руководство всеми учебными проектами осуществляет куратор учебной дисциплины. Куратор отслеживает общее состояние всех проектов на основе производимых разработчиками и инструктором отчетов (журналы). Куратор выборочно посещает собрания разработчиков. Он участвует в аттестации проектов и итоговой аттестации студентов по данной учебной дисциплине.

4 Трудозатраты разработчиков и примерный график работ

Собственно процесс разработки разбивается на 15 недель (весенний семестр). В течение каждой недели каждый из разработчиков тратит не менее 6 часов (учебная практика) на решение задач, связанных с проектом, — индивидуальная и, частично, командная работа. Кроме того, 2 часа в неделю (лабораторная работа) используются для проведения официальных собраний — командная работа. Итого, 8 часов в неделю. Дополнительная (16-я неделя) семестра используется для завершения и аттестации проекта.

Разработчики берут за основу следующий ниже понедельный график работ. В плане проекта рекомендуется предусмотреть небольшое опережение этого графика, чтобы иметь некоторый резерв времени. Каждую неделю на официальном собрании выполняется отчет команды в целом и индивидуальные отчеты разработчиков о прохождении соответствующего этапа.

Этап регистрации

Неделя 0. Формирование команд проектов и выбор тематики (в течении декабря осеннего семестра, т. е. до начала собственно проекта).

Этап планирования и анализа требований

Неделя 1. Установочная встреча с заказчиком и инструктором.

Неделя 2. Отчет о формировании плана проекта.

Неделя 3. Отчет об основных требованиях пользователя и моделях предметной области. Уточнение плана проекта.

Неделя 4. Отчет о ходе анализа требований. Черновые варианты моделей требований и высокоуровневой архитектуры. Уточнение плана проекта.

Неделя 5. Отчет о создании спецификации требований. Модели требований и высокоуровневой архитектуры. Критерии аттестации. Документ спецификации требований (техническое задание). Согласование с заказчиком. Уточнение плана проекта.

Этап проектирования

Неделя 6. Аттестация технического задания заказчиком. Переход к стадии проектирования.

Неделя 7. Отчет о проектировании. Архитектура системы. Интерфейс пользователя. Структура тестовых сценариев.

Неделя 8. Отчет о проектировании. Подсистемы, интерфейс подсистем, структуры данных, базовые алгоритмы. Черновой вариант покрывающего множества тестов.

Неделя 9. Отчет о проектировании. Документ проектирования и план тестирования. Структура руководства пользователя.

Этап кодирования и блочного тестирования

Неделя 10. Отчет о реализации. Структура кода. Стиль кодирования и комментирования. Способы управления кодом.

Неделя 11. Отчет о реализации и тестировании блоков, отладка.

Неделя 12. Отчет о реализации и интеграционном тестировании, отладка. Документ реализации. Черновой вариант журнала о выполнении тестирования. Метрики кода и тестирования.

Этап системного тестирования и окончательной отладки

Неделя 13. Отчет о системном тестировании. Отладка. Черновой вариант руководства пользователя. Уточнение журнала о выполнении тестирования.

Неделя 14. Отчет о тестировании. Соответствие требованиям. Документ выполнения тестирования. Уточнение руководства пользователя.

Неделя 15. Аттестация и презентация проекта. Все документы и программный код.

Этап сдачи и подведения итогов

Неделя 16. Завершение и сдача проекта. Семинар с участием всех команд разработчиков, инструкторов, заказчиков и сторонних экспертов. (зачетная неделя).

5 Схема собрания

Собрание является точкой интеграции полученных результатов проекта и определения дальнейших действий проекта. На собрании вырабатываются высокоуровневые решения. Например, распределение задач, определение приоритетов, назначение ответственных, изменение плана и т.п. Кроме того, собрание используется для отчета инструктору о ходе процесса разработки. Все необходимые материалы предоставляются инструктору заранее.

Рекомендуется следующая схема проведения собрания.

1.Председатель собрания (менеджер проекта) назначает секретаря и оглашает заранее подготовленную повестку собрания и краткий отчет о текущем состоянии проекта. (~5 минут).

2.Участники собрания вносят замечания. В результате формируется окончательная повестка собрания. (~5 минут).

3.Собрание идет согласно выработанной повестке. Заслушиваются отчеты разработчиков о проделанной работе, включая отчеты об исправлении замечаний инструктора и заказчика, и выполняется анализ текущей документации и других артефактов. Обсуждаются и определяются дальнейшие направления и задачи. (~35–70 минут).

4.Подведение итогов. Распределение задач, назначение ответственных и т.п. (~10 минут).

Инструктор оценивает работу разработчиков. На основании этого он формирует замечания, комментарии и еженедельную оценку (команде в целом и индивидуально каждому разработчику).

Собрание также может использоваться для работы с заказчиком. В этом случае в п. 3 схемы проведения собрания обсуждаются вопросы, связанные с получением информации от заказчика.

Замечания и комментарии инструктора и заказчика в обязательном порядке заносятся в протокол собрания. Рекомендуется для этого использовать отдельные пункты в протоколе.

В случае недостаточности еженедельных собраний для командной работы разработчиков, команда может использовать дополнительное время. Например, это полезно на начальных этапах проекта в случае активной рабо-

ты с заказчиком.

6 Взаимодействие с заказчиком

В первую неделю проводится установочная встреча заказчика с командой разработчиков. Эта встреча является официальным собранием проекта, на котором заказчик дает обзор поставленной в рамках проекта задачи. До этого собрания разработчики ДОЛЖНЫ быть ознакомлены с предложениями по созданию ПО, на основе которых они ЗАРАНЕЕ готовят вопросы к заказчику и черновой вариант плана проекта.

В первой половине собрания заказчик делает доклад, между заказчиком и разработчиками происходит обсуждение задачи и плана проекта. Также определяются формы взаимодействия разработчиков с заказчиком. Позднее эти формы взаимодействия будут официально утверждены в плане проекта. Вторая половина собрания посвящена уточнению плана проекта и разработке структуры спецификации требований. Присутствие заказчика на второй половине собрания необязательно.

На начальном этапе (первые 5 недель семестра) проекта необходимо сформировать спецификацию требований к ПО — техническое задание. Непосредственно созданием этого документа занимаются сами разработчики, используя при этом заказчика как основной источник информации. До недели 6 семестра (включительно) заказчик и разработчики должны придти к единому пониманию поставленной задачи, что заключается в создании такого варианта спецификации требований, который бы устраивал обе стороны. Этот вариант официально утверждается заказчиком. На этом этапе заказчик регулярно взаимодействует с командой и обеспечивают:

1. своевременное предоставление необходимой для разработки информации;
2. проверку на отражение всех требований и их соответствие мнению заказчика (полнота и непротиворечивость);
3. проверку на понятность спецификации заказчику;
4. согласование правил приемки ПО заказчиком (критерии аттестации);

Заказчик консультирует разработчиков в рамках схемы взаимодействия, вырабатываемой на установочной встрече. Это может включать 1) присутствие заказчика на ряде официальных собраний разработчиков, 2) организацию заказчиком специальных семинаров с разработчиками, 3) интервьюирование заказчика разработчиками, 4) обмен информацией по электронной почте, 5) предоставление ссылок на информационные ресурсы (публикации, Интернет-ресурсы) и другие способы.

Заказчик может требовать от разработчиков взаимодействия с ним на языке, ориентированном на проблемную область задачи, а не на языке узкоспециализированных программных терминов, которым владеют разработчики. С этой точки зрения, следующие документы: должны быть написаны разработчиками так, чтобы быть понятными заказчику:

1. спецификация требований (полностью);
2. план тестирования и журнал о выполнении тестирования (те части, которые касаются критериев аттестации);
3. руководство пользователя (полностью).

Рекомендуемая структура спецификации требований представлена в шаблонах документации. По соглашению с инструктором, заказчиком и разработчиками эта структура может быть изменена, чтобы учесть специфику конкретного проекта. Часть материала наследуется (с детализацией) из предложений по написанию ПО и плана проекта.

На заключительном этапе проекта (недели 14–16) заказчик аттестует разработанное ПО и может принять участие в оценивании работы команды разработчиков. Это оформляется в виде официального собрания команды разработчиков. Аттестация выполняется на основе критериев аттестации, определенных в спецификации требований. Мнение заказчика, а также список всех имеющихся замечаний и пожеланий, заносятся разработчиками в документ «Показатели проекта», раздел «Аттестация». Результаты аттестации представляются на заключительном семинаре, где команды разработчиков будут выступать с краткими докладами (презентация проекта).

7 Инструментальные средства

Настоятельное рекомендуется использование в проектах следующих классов инструментальных средств.

1.Электронная почта.

2.Централизованное хранение файлов (возможно, система управления версиями).

3.Представление данных о ходе проекта в виде web-ресурса.

4.Инструментальные средства разработки ПО.

- Редакторы (текстовые, графические, моделей) и другие средства документирования, включая web-средства.
- Компиляторы, отладчики, редакторы связей, менеджеры сборки и конфигурации.
- Стандартные библиотеки и оболочки разработки ПО.
- Системы тестирования и статического анализа кода.

Возможно предоставление и других средств, в зависимости от специфики проекта. Поощряется самостоятельное нахождение студентами, изучение и использование специализированных инструментальных средств (в рамках существующего законодательства РФ).

Рекомендуется использовать студенческий сервер kappa.cs.karelia.ru, работающий под ОС Linux, обеспечивающий базовый набор перечисленных выше инструментов на основе свободно-распространяемого ПО.

8 Аттестация и презентация проекта

Для аттестации проекта необходимо следующее.

1. Программный продукт и документация проекта.
2. Заключение заказчика.
3. Заключение инструктора с еженедельными и итоговыми оценками (журнал выполнения проекта).
4. Заключение куратора.
5. Заключение сторонних экспертов (необязательно).
6. Презентация проекта (см. шаблоны документации).

На основании этих материалов аттестуется проект в целом. Также определяются индивидуальные итоговые оценки каждого разработчика. Последние определяют итоговую оценку по данной учебной дисциплине.