

О. А. Змеев, Д. О. Змеев, Д. А. Соколов,

Национальный исследовательский Томский государственный университет

РЕАЛИЗАЦИЯ ПРОЕКТНОГО МЕТОДА ОБУЧЕНИЯ НА ОСНОВЕ ОБОБЩЕННОЙ МОДЕЛИ ПРОЦЕССА РАЗРАБОТКИ

Аннотация

В статье рассматривается проектное обучение процессам разработки программного обеспечения, которое основано на обобщенной модели процесса разработки программного обеспечения, спроектированной и реализованной в среде управления проектами Redmine.

Ключевые слова: процессы разработки программного обеспечения, проектный метод обучения, электронный учебно-методический комплекс.

Введение

В дискуссиях, связанных с обсуждением модели выпускника бакалавриата, в последнее время очень много внимания уделяется вопросам формирования так называемых «soft skills» [5]. Многие федеральные образовательные стандарты высшего образования третьего поколения включают соответствующие требования в раздел общекультурных, реже — профессиональных компетенций. Не исключено, что в новом поколении стандартов этот тренд найдет свое дальнейшее отражение. Но если рассматривать вопросы, связанные с подготовкой бакалавров ИТ-направлений, то многие знания, умения и навыки, которые обычно считаются частью «soft skills», являются обязательными. Действительно, в настоящее время очень трудно начать профессиональную деятельность в качестве ИТ-специалиста, не обладая компетенциями в области проектной деятельности, командной работы, тайм-менеджмента [4].

В ИТ-области соответствующие компетенции тесно переплетаются с такими сложными, сугубо профессиональными вопросами, как процессы разработки про-

граммного обеспечения, формирование и управление командами разработчиков в рамках этих процессов, экономическая эффективность проектной деятельности, работа в современных системах управления проектами и т. д. Таким образом, соответствующий набор знаний, умений и навыков должен быть неотъемлемой частью модели выпускника основной образовательной программы, связанной с информационными технологиями. Но сформировать этот набор в рамках классической модели организации образовательного процесса, предусматривающей стандартные лекции, практики, лабораторные работы, очень сложно.

Для решения этой задачи можно воспользоваться проектным методом обучения [3]. Разрабатывая программу дисциплины, основанную на этом подходе, преподаватель должен решить как минимум следующие задачи:

- сформировать у студентов представление о современных методологических вопросах в области процессов разработки программного обеспечения;
- показать, как теоретические изыскания отражаются в реальных процессах разработки;

Контактная информация

Змеев Олег Алексеевич, доктор физ.-мат. наук, профессор, зав. кафедрой программной инженерии Национального исследовательского Томского государственного университета; *адрес:* 634050, г. Томск, пр-т Ленина, д. 36; *телефон:* (382) 252-98-70; *e-mail:* ozmeyer@gmail.com

Змеев Денис Олегович, ассистент кафедры программной инженерии Национального исследовательского Томского государственного университета; *адрес:* 634050, г. Томск, пр-т Ленина, д. 36; *телефон:* (382) 252-98-70; *e-mail:* denis.zmееv@studdb.ru

Соколов Данила Александрович, ассистент кафедры программной инженерии Национального исследовательского Томского государственного университета; *адрес:* 634050, г. Томск, пр-т Ленина, д. 36; *телефон:* (382) 252-98-70; *e-mail:* danila.sokolov@studdb.ru

O. A. Zmееv, D. O. Zmееv, D. A. Sokolov,
National ResearchTomsk State University

IMPLEMENTATION OF PROJECT EDUCATION METHOD BASED ON GENERALIZED MODEL OF SOFTWARE DEVELOPMENT PROCESS

Abstract

The article describes the case of project education method which found on generalized model of software development process. This case was designed and implemented in project management software Redmine.

Keywords: software development processes, project method of training, electronic learning methodical complex.

- обеспечить практический опыт участия в проекте по разработке программного обеспечения;
- сформировать опыт практического использования современной среды управления такими проектами.

В рамках настоящей работы делается попытка обобщения многолетнего опыта реализации дисциплины «Программная инженерия» на факультете информатики Томского государственного университета.

Структура дисциплины

В структуре основной образовательной программы подготовки бакалавра дисциплина «Программная инженерия» завершает цикл курсов, посвященных объектно-ориентированным технологиям разработки программного обеспечения. Таким образом, с дисциплиной будут знакомиться студенты, которые имеют представления и практические навыки, связанные с такими важными вопросами, как объектно-ориентированное программирование, объектно-ориентированный анализ и проектирование, разработка архитектуры программных систем. Основная цель дисциплины — собрать все эти знания, умения и навыки в единое целое и показать, каким образом технологии разработки программного обеспечения используются в рамках проектной деятельности.

Применение проектного метода обучения подразумевает, что центральным элементом соответствующего курса становится проектная работа, вокруг которой организуется учебный процесс. При конструировании проекта необходимо иметь в виду некоторые особенности, которые вызваны спецификой предмета: проект должен быть исполнимым в рамках заданного срока и бюджета, иметь особенности, направленные на изучение процесса разработки программного обеспечения, и при этом он должен быть похожим на проекты, присутствующие на реальном рынке программного обеспечения.

Для реализации проекта студенты делятся на команды по 8–10 человек, в каждой из которых выбирается свой проектный менеджер. С технологической точки зрения проект содержит либо один сложный, либо два менее сложных архитектурных риска. Подразумевается, что способы преодоления рисков студентам известны из ранее пройденного материала. Организационно у проекта есть конкретный заказчик, роль которого обычно исполняет студент старших курсов (магистрант или аспирант).

Учитывая, что основной целью курса является реализация проекта, несколько меняется назначение лекционных занятий. Их условно можно поделить на два типа: фактологические — на них освещаются теоретические вопросы разработки современного программного обеспечения, и мастер-классы, на которых разбираются ошибки, сделанные командой в рамках исполнения проекта. Контекст мастер-классов понятен всем участникам, так как все команды, выполняя один и тот же проект, испытывают похожие затруднения. Очень важным элементом является тот факт, что каждый из студентов находится в ситуации, имитирующей процессы, происходящие в реальной IT-компании: он выполняет конкретные задачи, исполняя при этом определенную роль в рамках

определенного процесса разработки, взаимодействует с другими участниками, получает виртуальную заработную плату. Более того, итоговая оценка по дисциплине напрямую зависит от полученной студентом в ходе выполнения проекта заработной платы.

Для формирования необходимых студентам компетенций преподаватель должен решить еще ряд задач: изученный ранее теоретический материал должен быть напрямую связан с практической работой, и, конечно, необходимо погрузить студентов в реально действующую среду, чтобы, во-первых, они освоили эту среду и, во-вторых, эта среда позволяла бы преподавателю контролировать весь процесс выполнения проекта.

Модель процесса разработки и его практическая реализация

Для реализации комплексного решения, связывающего теоретический материал, элементы текущего процесса разработки и конкретные задачи, которые реализует каждый исполнитель в рамках проекта, необходимо разработать обобщенную схему взаимодействия между этими элементами. Дополнительным преимуществом такого подхода становится тот факт, что такое обобщенное моделирование позволяет избавиться от конкретных технологий, которые используются для организации проектной работы в текущий момент времени.

В основе реализации такой обобщенной схемы лежит обобщенная модель процесса разработки программного обеспечения (рис. 1). Предлагаемая модель не претендует на полноту описания самого процесса разработки, ее основной задачей является наглядное представление структуры основных рабочих элементов, которые выполняются в процессе разработки таким образом, чтобы эту структуру можно было перенести в виде задач и проектов в среду управления проектами. Именно эта особенность отличает предложенную модель от других способов описания процессов разработки программного обеспечения.

В основе предлагаемой модели лежит паттерн компоновщик [1], который позволяет создавать вложенные иерархические структуры из базовых элементов. Базовым элементом является «единица работы», которая в реализации процесса разработки превращается в задачу; агрегирующей сущностью является «процесс», который в зависимости от сложности структуры процесса разработки может быть проектом в целом (у процесса нет сложной иерархической структуры деятельности) или же может быть итерацией, спринтом, фазой и т. д. (если процесс разработки подразумевает наличие подобных механизмов). Особенностью предлагаемого компоновщика является необходимость при создании агрегации из сущностей учитывать порядок их добавления (в большинстве процессов разработки задачи идут друг за другом в строгой последовательности и нельзя, например, проводить тестирование без собранной сборки системы).

Другие важные сущности — это «артефакт» и «роль». Артефакт подразумевает, что любая деятельность в любом процессе разработки направлена на

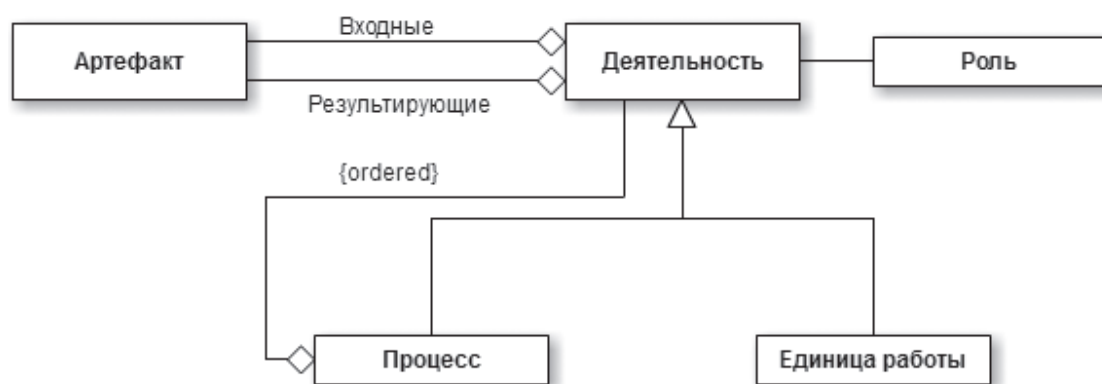


Рис. 1. Обобщенная модель процесса разработки программного обеспечения

создание или изменение результирующего множества артефактов (код программного обеспечения, акты согласования сдачи проекта, техническое задание, рабочие артефакты для проектирования системы и т. д.) из входного множества артефактов (которое может быть пустым, особенно на начальных этапах проекта, например, сбор первичных требований к разрабатываемому продукту). Роль соответствует некоторому связанному набору функциональных обязанностей, необходимых для выполнения той или иной деятельности в рамках процесса разработки (архитектор, инженер по тестированию, разработчик и т. п.). Менеджер проекта, выдавая конкретному исполнителю реальную задачу, назначает его на соответствующую роль. В случае структурированных процессов разработки роли определяются детально и точно, тем не менее в разные промежутки времени каждый член команды может исполнять различные роли. А в случае неструктурированных по ролям процессов разработки ролей две — менеджер и исполнитель.

И наконец, для организации взаимодействия между теоретическим процессом разработки и реальной средой управления проектами можно воспользоваться таким явлением, как дихотомия [7], но в рамках процессов разработки. Если применить этот подход к процессам разработки, то абстрактные деятельности, итерации, фазы, описанные в процессах разработки (классы), на практике в реальных проектах имплементируются в конкретные задачи, категории и трудозатраты, которые характерны для проектов (объекты). Благодаря этому явлению в описании процессов разработки достаточно описать только принципиально различные сущности (понятие спринта в общем виде), которые на практике будут использоваться для описания конкретных элементов среды управления проектами, в рамках которой реализуется учебный проект.

Продемонстрируем все вышесказанное на примере текущей среды для организации соответствующего курса «Программная инженерия».

Программная реализация системы управления проектами

На практике в текущей реализации курса используется следующее сочетание представленных выше элементов.

Методологическая часть курса основана на унифицированном процессе разработки [6], что позволяет преподавателю достаточно полно сосредоточиться на главных теоретических предпосылках, лежащих в основе реально существующих процессов разработки.

В качестве практического процесса используется специально адаптированный для учебных целей вариант ICONIX [8]. Во-первых, это достаточно простой процесс разработки, реализация которого избавлена от большого числа несущественных деталей. Во-вторых, этот процесс акцентирует внимание студентов на принципиальных в современных процессах разработки рабочих потоках: управление требованиями, устойчивая архитектура. И наконец, это итерационный и инкрементный процесс. В качестве основы для разработки среды управления проектами выбрана Redmine [9] — одна из наиболее популярных бесплатных и несвязанных с платформой разработки сред для управления проектами, которая активно используется в реальном IT-бизнесе.

На первом этапе построения интегрированной проектной среды строится обобщенная модель процесса разработки.

В качестве иллюстрации приведем два примера. На рисунке 2 показана диаграмма верхнего уровня, описывающая основные элементы процесса: проект, фаза, итерация, деятельность, конкретная задача.

На рисунке 3 основной акцент делается на описании декомпозиции конкретной деятельности на конечные задачи. На диаграмме рассмотрен пример из типичных задач унифицированного процесса разработки «Найти актеров и варианты использования» и то, как эта деятельность реализуется в виде конкретного набора ролей, задач и артефактов.

На втором этапе разработанная обобщенная модель реализуется в рамках конкретного менеджера проектов. Для этого необходимо построить соответствие между элементами теоретической модели и конкретными элементами реальной программной среды. Результат этой работы представлен на рисунке 4.

На последнем этапе реализованное решение наполняется справочными материалами, которые облегчают студентам понимание различных деятельностей, артефактов и т. д. На рисунках 5–8 представлены различные виды реализованного решения.



Рис. 2. Фрагмент обобщенной модели процесса разработки

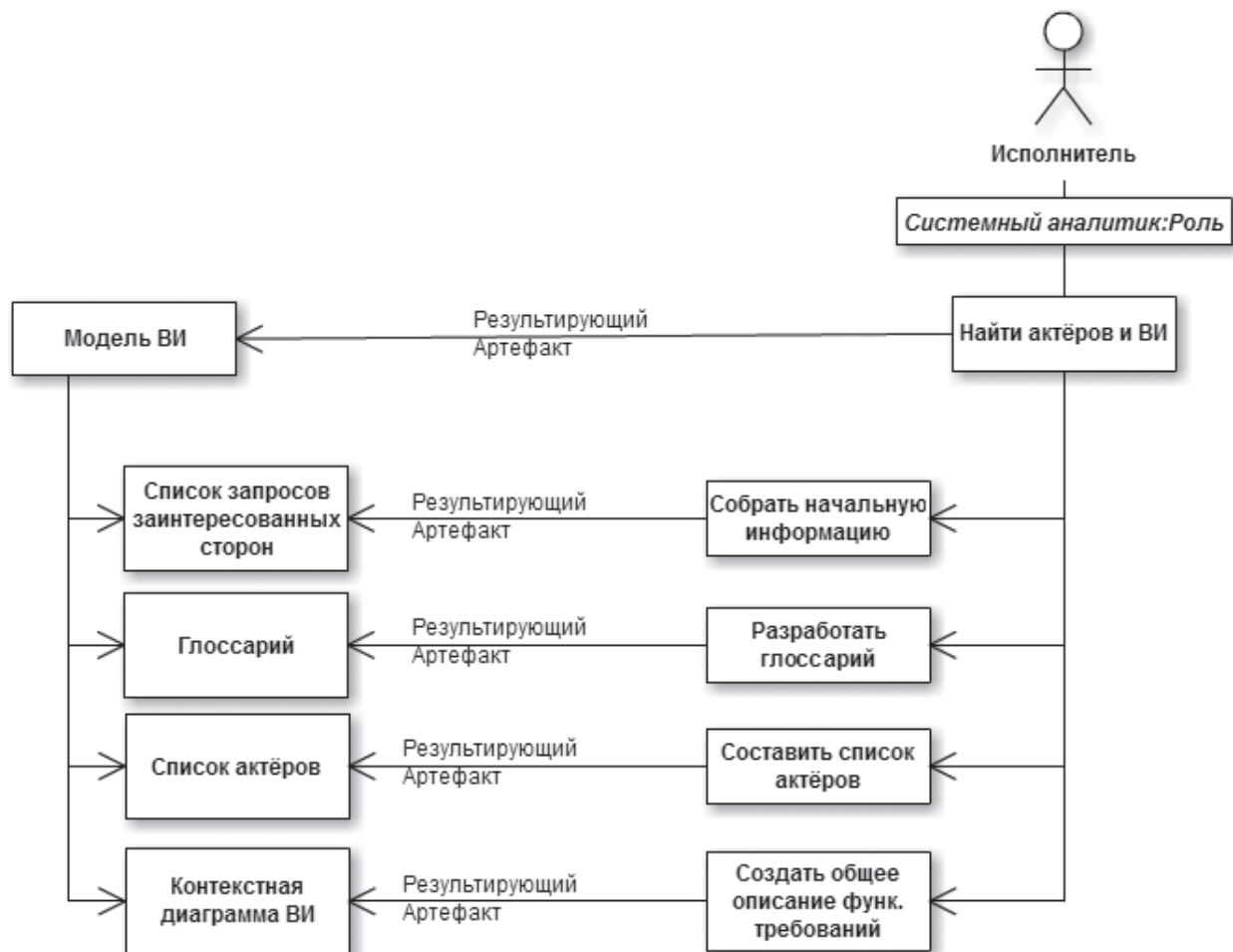


Рис. 3. Декомпозиция деятельности «Найти актёров и варианты использования»

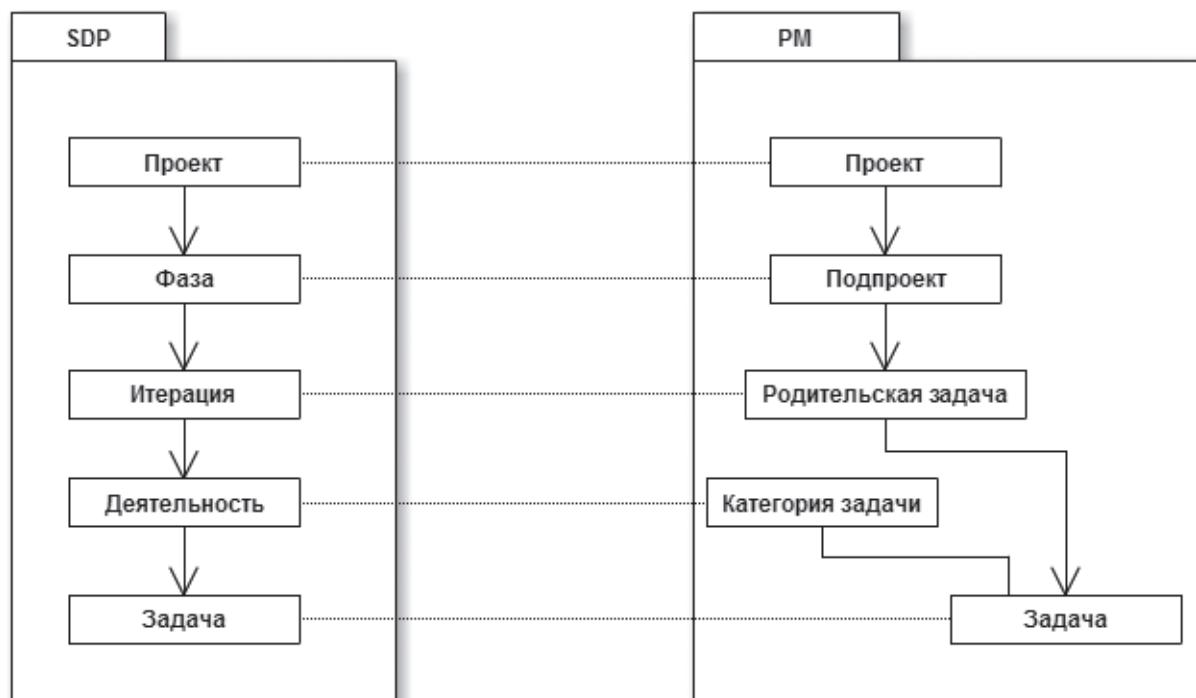


Рис. 4. Реализация модели в Redmine

Домашняя страница
Моя страница
Проекты
Сотрудники
Администрирование
Помощь

РМ "Программная инженерия"

Домашняя страница

Программная инженерия классически определяется как системный подход к анализу, проектированию, оценке, реализации, тестированию, обслуживанию и модернизации программного обеспечения, то есть применение инженерных принципов к разработке программного обеспечения. В рамках текущей концепции изложения специальных дисциплин на факультете информатики этот предмет является завершающей цепочкой по изучению объектно-ориентированных технологий разработки программного обеспечения.

Основной акцент в курсе лекций делается на Унифицированном процессе разработки программного обеспечения. Разбираются основные принципы построения современных процессов разработки, анализируются лучшие практики современной инженерии ПО. Практическая часть курса организована в виде бизнес-кейса, в рамках которого студенты делятся на проектные команды и выполняют проект по реализации заказного программного обеспечения в условиях некоторой виртуальной среды ведения бизнеса. Сейчас вы находитесь в этой среде.

Итоговая оценка по курсу выставляется на основе бизнес-кейса. В начале семестра студенты делятся на команды разработчиков. Каждая команда состоит из 8-12 человек. Каждая из сформированных команд управляется менеджером из числа студентов. Команды получают проект (общая продолжительность проекта примерно 1,5-2 человека месяца, проект ориентирован на работу вокруг богатой модели функциональных требований, содержит некоторое количество архитектурных рисков). Команды выполняют проект конкурируя между собой, в процессе выполнения каждый студент получает заработную плату в виде некоторой условной валюты. Рабочие артефакты проекта фиксируются только в специальной виртуальной среде УМК "Введение в Унифицированный процесс".

В конце семестра происходит презентация проекта и команды ранжируются согласно полученным результатам процесса разработки. В зависимости от занятого места команда получает базовую оценку за курс, а зарплата каждого участника пересчитывается в конкретную оценку. Кроме этого у преподавателя есть возможность отслеживать и оценивать любую активность студента во время выполнения проекта независимо от менеджера команды. Никакого альтернативного способа сдачи курса, кроме выполнения проекта в команде, НЕ СУЩЕСТВУЕТ.

Последние проекты

- Команда 5 Фаза 3 (15.12.2016 11:52)
- Команда 4 Фаза 3 (15.12.2016 11:51)
- Команда 3 Фаза 3 (15.12.2016 11:51)
- Команда 2 Фаза 3 (15.12.2016 11:50)
- Команда 1 Фаза 3 (15.12.2016 11:49)

Рис. 5. Стартовая страница среды

Проект » Фаза 1

Обзор Действия **Задачи** Диаграмма Ганта Календарь

Итерация #2011


Типовая итерация 1ой фазы
Добавил(а) Prof. Zmeev около 1 месяца назад. Обновлено около 1 месяца назад.

Статус: Новое
Приоритет: Высокий
Дата завершения:
Готовность: 0%
Оценка трудозатрат: 0.00 h
Дата начала:

Описание

Самая вводная итерация

Подзадачи

Задача #2017: Собрать начальную информацию по проекту	Новое	
Задача #2018: Разработать глоссарий	Новое	
Задача #2019: Составить список актёров	Новое	
Задача #2020: Создать общее описание функциональных требований	Новое	
Задача #2021: Определить приоритеты ВИ(заказчик)	Новое	
Задача #2022: Определить АЗ ВИ	Новое	
Задача #2023: Составить общий отсортированный список ВИ	Новое	
Задача #2024: Детализировать ВИ в порядке приоритетов	Новое	
Задача #2025: Составить список рисков архитектуры	Новое	
Задача #2026: Проанализировать нефункциональные требования	Новое	
Задача #2027: Оценить ВИ по приоритетам	Новое	
Задача #2028: Определить стек технологий	Новое	
Задача #2029: Анализировать АЗ ВИ	Новое	

Рис. 6. Описание типовой итерации процесса разработки


Итерация #2011: Типовая итерация 1ой фазы
Определить приоритеты ВИ(заказчик)
Добавил(а) УМК Admin 11 месяца назад. Обновлено 6 месяца назад.

« Предыдущее | 20/24 | Следующее »

Статус: Новое
Приоритет: Обычный
Назначена: -
Категория: Выполняется практически всегда
Версия: -
Деятельность типовой итерации: Расставить ВИ по приоритетам
Дата начала:

Дата завершения:
Готовность: 0%
Артефакт: Список значимых ВИ для заказчика

Описание [Цитировать](#)

Одним из наиболее важных результатов 1ой фазы - определение той функциональности, которая должна входить в эволюционный прототип, а для этого необходимо понимать, что в первую очередь захочет увидеть заказчик, чтобы сделать то, что ему понравится(и больше срубить профитов). Способы получения приоритетов ВИ от заказчика могут быть различными, один из наиболее стандартных - голосование по приоритетам...

Выделяются 3 группы для голосования

- 1) То что в продукте по-любому должно быть - 3 балла
- 2) То что очень хочется в продукте - 1 балл
- 3) То без чего продукт может обойтись - 0 баллов

Все 3 уровня есть всегда, но напрямую спрашивать рискованно, поэтому можно заменить на баллы

Заказчику(или всем заинтересованным сторонам) даётся листок для голосования, и говорится, что у него есть возможность распределить приоритеты, при этом оценок у него на каждую категория 1/3 от общего числа возможных вариантов.

Примечание! Помните, что ВИ есть связанные по функциональности, что не всегда понятно заказчику, соответственно он может проголосовать так, что 2 ВИ которые нужны для работы с друг с другом, окажутся в двух принципиально разных категориях. Как разрешить этот парадокс - придумайте сами.

Подзадачи [Добавить](#)

Связанные задачи [Добавить](#)

Рис. 7. Справочная страница деятельности

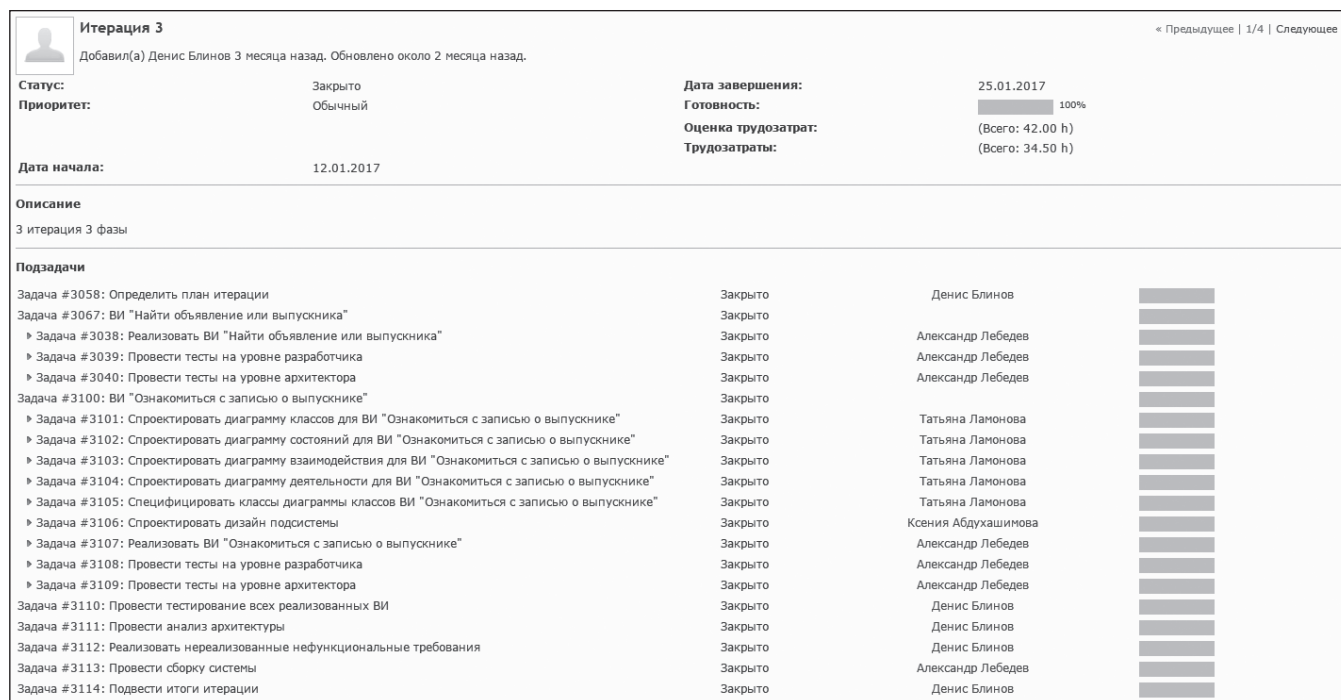


Рис. 8. Итерация реального проекта

Заключение

Представленное программное решение позволяет добиться следующих результатов: с одной стороны, студенты получают ценный практический опыт, осваивают знания, умения и навыки, предусмотренные рабочей программой дисциплины, с другой — у них формируется необходимый в IT-сфере набор профессиональных компетенций.

В планах дальнейшего развития среды разработки набора аналитических инструментов для преподавателя, интеграция с системой контроля версий.

Список использованных источников

1. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2001.
2. Змеев О. А., Ерин Е. А. Учебно-методический комплекс «Введение в унифицированный процесс разработки программного обеспечения» // Материалы VII Всероссий-

ской научно-практической конференции с международным участием «Информационные технологии и математическое моделирование» (ИТММ — 2008), 14–15 ноября 2008 г. Томск, 2008.

3. Метод проектов // Википедия. https://ru.wikipedia.org/wiki/Метод_проектов

4. Профессиональные стандарты в области ИТ // Ассоциация предприятий компьютерных и информационных технологий. <http://www.apkit.ru/committees/education/meetings/standarts.php>

5. Триумвират «скиллз» как формула востребованности выпускника вуза // Веб-сайт ТГУ. http://www.tsu.ru/university/rector_page/triumvirat-skillz-kak-formula-vostrebovannosti-vypusknika-vuza/

6. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб.: Питер, 2002.

7. Geisler R., Klar M., Pons C. Dimensions and Dichotomy in Metamodeling // EWIC. 1998. <http://ewic.bcs.org/content/ConWebDoc/4361>

8. ICONIX. <http://www.iconixsw.com>

9. Redmine. <http://www.redmine.org>

НОВОСТИ

«Гибридный» рекорд

В Microsoft Research научили искусственный разум играть в одну из популярных компьютерных игр, Ms. Pac-Man, и сделали из него чемпиона, побившего мировой рекорд, установленный человеком. Играя в версию аркады Ms. Pac-Man, выпущенную для одной из первых домашних консолей Atari 2600, искусственный интеллект смог набрать максимальное количество возможных очков — результат умной машины составил 999 990 баллов, тогда как лучший результат,

показанный человеком, равен 266 360. При обучении искусственного интеллекта использовался метод «гибридной архитектуры наград». Он заключается в том, что 150 специальным программам-агентам назначается конкретная задача: избегать призраков, правильно передвигаться, собирать гранулы и т. д. С помощью программ-агентов искусственный интеллект самостоятельно распределял приоритеты для достижения максимального результата.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)