

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ПРОГРАММНОЙ ИНЖЕНЕРИИ В ПРОЦЕССЕ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

В.М. Гриняк

Введение. Изучение программирования на языках высокого уровня составляет неотъемлемую часть подготовки по направлениям высшего профессионального образования, связанным с информационными технологиями [3-7]. Несмотря на эволюцию ИТ отрасли и уменьшение доли присутствия в ней собственно разработки программных средств, соответствующие компетенции остаются востребованными рынком труда [2], а дисциплины, включающие изучение программирование на языках высокого уровня, являются обязательной составляющей учебных планов вузов.

Учебно-методический комплекс таких дисциплин, как правило, включает в себя: лекционный материал, задания для лабораторных работ, задания для самостоятельной работы, вопросы для самоконтроля, вопросы к экзамену (устному или в форме компьютерного тестирования). Традиционная технология обучения программированию включает в себя проведение лекционных занятий с обзором конструкций языка и базовых алгоритмических задач, проведение лабораторных работ, во время которых студенты, консультируясь с преподавателем, пишут простейшие программы, а также выполнение заданий для самостоятельной работы (которая тоже заключается в написании той или иной программы) и контроль их выполнения.

Эволюция информационных технологий привела к тому, что потребность в специалистах, разрабатывающих программное обеспечение самостоятельно или в составе небольших творческих коллективов, основательно снизилась. В настоящее время сфера информационных технологий характеризуется в основном промышленной разработкой сложных программных комплексов и систем, причем большими территориально распределенными коллективами. Обучение программированию в рамках традиционной образовательной технологии формирует компетенции выпускников, ориентированные скорее на хакинг и фриланс и не способствует формированию у студентов компетенций, необходимых для работы на крупных проектах, где наряду с навыками самостоятельной разработки программ необходимо умение работать в команде, владение средствами поддержки такой работы и наличие «культуры» программирования. Подготовка профессионалов для работы в крупных ИТ компаниях нуждается в принципиально иных методах, по сравнению с традиционными, необходимость их разработки становится все более острой.

Для решения обозначенной проблемы авторы предлагают использовать при обучении программированию методики, применяемые в промышленной разработке программного обеспечения и ассоциированные в настоящее время с понятием «программная инженерия». В предлагаемом подходе применяются активные формы обучения: имитируется процесс разработки программного обеспечения, принятый в ИТ отрасли; тем самым процесс обучения приближается к условиям реального рынка труда.

Предлагаемая модель процесса обучения. Для описания модели процесса обучения введём следующие понятия.

Участник процесса обучения. Участниками процесса обучения являются студент и преподаватель.

Роль. Под ролью понимается совокупность действий и функций, выполняемых участниками процесса. Студент и преподаватель в процессе обучения с использованием методов программной инженерии могут пребывать в различных ролях, причем одна и та же роль может выполняться как студентом, так и преподавателем.

Рабочий продукт. Результат работы участника процесса в той или иной роли.

Примем, что основным объектом в описываемой модели является задача по программированию, над которой идет коллективная работа. Введем следующие роли участников этого коллективного образовательного процесса.

Автор. В задачу автора входит формулировка задачи для разработки студентом.

Модератор. В задачу модератора входит назначение задачи на исполнителей, контроль процесса работы над задачей и вынесение окончательного решения о результате её решения.

Кодировщик. Задача кодировщика - написание программы для решения назначенной ему задачи.

Инспектор. Инспектор занимается анализом кода разработанной программы на предмет её корректности и качества кодирования, адекватности применяемых программных решений. Инспектор выносит один из трёх возможных вердиктов:

- Принять задачу без замечаний. Это означает, что инспектору совершенно понятны все программные решения, код написан грамотно и правильно, текст программы снабжен всеми необходимыми комментариями.
- Принять задачу при условии устранения замечаний. Это означает, что программа в целом написана правильно, однако некоторые моменты представляются непонятными или некорректными. Инспектор формулирует свои замечания и проверяет задачу вновь после их устранения кодировщиком. Процесс устранения замечаний продолжается итерационно до тех пор, пока инспектор не вынесет решение «Принять без замечаний».
- Отклонить задачу. Это означает, что программа не выполняет требуемых функций или количество замечаний столь велико, что она не может быть принята без существенной доработки.

Тестер. В задачу тестера входит разработка теста, который позволит проверить качество написания программы путем поиска в ней ошибок и собственно проверка. Работа тестера начинается после того, как инспектор вынес вердикт «Принять без замечаний». При этом тестер не должен анализировать код программы, он лишь должен запускать её и прогонять на сценариях, адекватных условию задачи.

Рабочими продуктами рассматриваемой модели процесса обучения являются: у автора - формулировка задачи; у модератора - оценка работы исполнителей; у кодировщика - код программы (текст); у инспектора - замечания и набор рекомендаций по доработке кода; у тестера - сценарий тестов.

Последовательность выполнения ролей в процессе работы над программой:

- Автор формулирует задачу, для решения которой необходимо написать программу и передает ее кодировщику.

- Модератор назначает роли.
- Кодировщик пишет программу и передает ее инспектору.
- Инспектор анализирует программный код и выносит свое решение. Если код принимается без замечаний, то программа передается тестеру. Если у инспектора есть замечания, он сообщает их кодировщику. Далее происходит взаимодействие кодировщика с инспектором по устранению замечаний до вынесения решения «Принять без замечаний».
- Тестер разрабатывает сценарий и проверяет качество программы.
- Модератор выносит окончательное решение о качестве программы и оценивает результаты работы.

Возможно несколько вариантов распределения ролей между участниками процесса. В таблице 1 показано распределение в рамках традиционных форм процесса обучения.

Таблица 1. Распределение ролей между участниками процесса - традиционная форма обучения

Роль	Участник	Функция
Автор	Преподаватель	Формулировка задачи
Модератор	Преподаватель	Распределение задач между студентами (выдача заданий)
Кодировщик	Студент	Написание программного кода
Инспектор	Преподаватель	Проверка кода программы, оценка работы студента
Тестер	Преподаватель	Запуск программы и её тестирование, оценка работы студента

В традиционном варианте роли автора, модератора, инспектора и тестера выполняет преподаватель, а студент выполняет лишь роль кодировщика. Применение такого подхода в настоящее время оправдано лишь для обучения студентов непрофильных (не связанных непосредственно с ИТ) направлений подготовки (относящихся, например, к группе «Экономика», «Менеджмент» и т.п.).

В таблице 2 показано распределение ролей для первого варианта предлагаемой технологии обучения.

Таблица 2. Распределение ролей между участниками образовательного процесса - предлагаемая технология обучения с использованием методов программной инженерии (вариант 1)

Роль	Участник	Функция
Автор	Преподаватель	Формулировка задачи
Модератор	Преподаватель	Распределение задач между студентами, определение роли каждого из студентов, оценка работы студента
Кодировщик	Студент 1	Написание программного кода
Инспектор	Студент 2	Проверка кода программы, формулировка замечаний по улучшению и доработке
Тестер	Студент 3	Разработка сценариев тестирования, запуск программы и её тестирование

В данном варианте преподаватель выполняет роли автора и модератора, а роли кодировщика, инспектора и тестера выполняют студенты. Преподаватель формулирует

задачи на ту или иную тему, распределяет роли между студентами, контролирует процесс их работы и оценивает результат. Студент-кодировщик пишет программный код и передает его студенту-инспектору. Инспектор анализирует представленный код, формулирует свои замечания и (при необходимости) возвращает его кодировщику на доработку. При отсутствии замечаний программа передается студенту-тестеру. Тестер разрабатывает сценарий теста и «прогоняет» через него программу с формулировкой вывода типа «пройдено / не пройдено» по каждому пункту сценария. По окончании работы тестера преподаватель выносит оценку каждому участнику.

В том случае, если для группы студентов реализуется углубленное изучение программирования, возможен второй вариант распределения ролей участников (таблица 3). В данном варианте роль автора выполняют студенты.

Таблица 3. Распределение ролей между участниками образовательного процесса - предлагаемая технология обучения с использованием методов программной инженерии (вариант 2)

Роль	Участник	Функция
Автор	Студент 1	Формулировка задачи
Модератор	Преподаватель	Определение роли каждого из студентов, распределение задач между студентами, оценка работы студента
Кодировщик	Студент 2	Написание программного кода
Инспектор	Студент 3	Проверка кода программы, формулировка замечаний по улучшению и доработке
Тестер	Студент 4	Разработка сценариев тестирования, запуск программы и её тестирование

Оба варианта (таблицы 2 и 3) распределения ролей в предлагаемой модели процесса обучения повторяют, по сути, известную технологию разработки программного обеспечения, реализуемую, например, в системах баг-трекинга [8]. Как показывает опыт авторов, они могут быть использованы в процессе обучения студентов всех направлений подготовки, связанных с информационными технологиями.

Предлагаемая модель процесса обучения программированию наиболее эффективна при его информатизации, в частности, её можно реализовать с применением специализированных систем поддержки процесса разработки программного обеспечения [8]. Вместе с тем, развертывание такой системы и её использование в учебном процессе университета может быть сопряжено с рядом организационных и технических трудностей. Поэтому в учебном процессе авторам представляется целесообразным использовать не специализированные системы, а интерактивные обучающие среды, использующие чат и форум. Для реализации предлагаемой модели процесса обучения программированию автором был создан образовательный ресурс в рамках грантовой программы «Преподаватель Онлайн» при поддержке Благотворительного фонда В. Потанина [1]. Использование созданного образовательного ресурса позволило перенести основную часть работы студентов в интернет-среду, что дало возможность не только смоделировать реальный производственный процесс разработки программного обеспечения, но и повысить качество разрабатываемых программ за счет знакомства студентов с работой других участников процесса.

По результатам применения описанной технологии в процессе обучения дисциплине «Программирование на языке высокого уровня» [1] можно сделать вывод об эффективности технологии с точки зрения приобретения студентами компетенций в области разработки программного обеспечения. Помимо этого, её использование позволяет повысить мотивацию студентов к изучению программирования, познакомить их с приёмами коллективной работы, применяемыми в профессиональной среде и обеспечить прозрачность выставления оценок по дисциплине.

Пристатейный список литературы.

1. Самый лучший программист - персональный сайт Виктора Гриняка [Электронный ресурс]. - Режим доступа: <http://vgrinyak.professorjournal.ru/> (дата обращения 01.04.13).
2. Слугина Н. Л., Гриняк В. М. Повышение уровня подготовки кадров в области современных информационных технологий на базе центра компетенций // Современные проблемы науки и образования. - 2012. - № 5. - С. 205-205.
3. Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки Программная инженерия [Электронный ресурс]. - Режим доступа: http://www.edu.ru/db/mo/Data/d_09/prm542-1.pdf (дата обращения 01.04.13).
4. Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки Информационные системы и технологии [Электронный ресурс]. - Режим доступа: http://www.edu.ru/db/mo/Data/d_10/prm25-1.pdf (дата обращения 01.04.13).
5. Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки Прикладная информатика [Электронный ресурс]. - Режим доступа: http://www.edu.ru/db/mo/Data/d_09/prm783-1.pdf (дата обращения 01.04.13).
6. Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки Информатика и вычислительная техника [Электронный ресурс]. - Режим доступа: http://www.edu.ru/db/mo/Data/d_09/prm553-1.pdf (дата обращения 01.04.13).
7. Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки Математическое обеспечение и администрирование информационных систем [Электронный ресурс]. - Режим доступа: http://www.edu.ru/db/mo/Data/d_09/prm713-1.pdf (дата обращения 01.04.13).
8. Project Tracking | Atlassian JIRA [Электронный ресурс]. - Режим доступа: <http://www.atlassian.com/software/jira/overview/project-management> (дата обращения 01.04.13).