

# Подход к преподаванию ИТ на основе проектов по разработке свободного программного обеспечения

А. К. Петренко, В. В. Кулямин, О. Л. Петренко, В. В. Рубанов, А. В. Хорошилов  
Институт системного программирования РАН (ИСП РАН), Москва

## Аннотация

В статье показывается, как использование открытых проектов по разработке программного обеспечения позволяет естественным образом реализовать принципы открытого образования в обучении студентов информационным технологиям. Представлена рамочная структура практических курсов по ИТ, основанная на личном участии студентов в таких проектах. Приводятся примеры ее использования на кафедрах системного программирования МФТИ и ВМиК МГУ.

## Открытая образовательная среда

В настоящее время общепризнанна необходимость трансформации российской экономики из преимущественно сырьевой в инновационную. Система образования должна соответствовать потребностям и тенденциям развития общества, поэтому большое внимание уделяется построению системы инновационного образования в России. При этом крайне важно не только следовать основополагающим концепциям инновационного образования, но и выбрать подходящий для современных условий способ их реализации.

Концептуальной основой инновационного образования можно считать сформулированный в конце 50-х годов академиком М. А. Лаврентьевым принцип: «наука–кадры–производство» (знаменитый «треугольник Лаврентьева» [1,2]). Большинство имеющихся сейчас подходов к инновационному образованию является реализацией этой идеи в современных условиях: обучение происходит в процессе создания новых знаний за счет интеграции фундаментальной науки, непосредственно учебного процесса и производства, позволяющей сохранять ориентацию на практическое применение получаемых знаний.

Традиционная система профессионального образования строится на передаче студентам фундаментальных знаний, позволяющих ориентироваться в широкой области, и конкретных знаний, непосредственно используемых в дальнейшей практической работе. Опыт показывает, однако, что в условиях интенсивного технологического развития такой подход к образованию порождает трудности, связанные, с одной стороны, с медленным накоплением у студента опыта, позволяющего самому преобразовывать фундаментальные знания в практически применимые, а с другой — с быстрым устареванием конкретных технических знаний. В таких условиях важнейшими навыками для эффективной деятельности становятся умение адаптировать и пополнять имеющиеся знания для нужд конкретных проектов, умение самостоятельно принимать решения и планировать свою деятельность.

Все сказанное приложимо к сфере информационно-коммуникационных технологий (ИКТ), поскольку эта область очень быстро развивается, одновременно служа одной из основных инфраструктурных составляющих современной инновационной экономики. Поэтому и обучение специалистов в этой области помимо передачи им необходимого объема знаний, должно быть нацелено на выработку необходимых навыков, связанных со способностью к дальнейшему самостоятельному развитию. Схематично набор навыков, необходимый современным ИКТ-специалистам, можно представить так, как это показано на Рис. 1.

Таким образом, только узкопрофессиональных умений недостаточно для успешной работы в области информационных технологий (ИТ). Меморандум Европейской Комиссии о непрерывном обучении [3] подчеркивает важность для современного специалиста таких

социальных умений, как умение действовать уверенно, направлять собственные действия на достижение нужных результатов и принимать рискованные решения. Кроме того, там же указывается необходимость овладения такими когнитивными навыками, как способность учиться, приспосабливаться к изменяющемуся окружению, быстро осваивать необходимые новые навыки и находить нужную информацию в широком информационном потоке, обрушивающемся на каждого вовлеченного в общественную жизнь человека.



**Рисунок 1. Навыки современного ИКТ-специалиста.**

Признание этих принципов непрерывного образования побуждает к активному переходу к так называемой *открытой образовательной системе*, ориентированной не столько на передачу обучающимся некоторого объема знаний, сколько на воспитание самостоятельно целеполагающего, самообучающегося индивида, способного к эффективному взаимодействию с постоянно меняющимся окружением — человека, который *умеет учиться*.

В такой системе образовательное пространство открыто для студента, ему видны возможные пути собственного развития и потому становится возможным обсуждение средств и методов достижения индивидуальных целей. Важнейшей характеристикой открытого образования является самостоятельное принятие студентом решений и ответственности за них.

Основные принципы открытого образования можно сформулировать так [4].

- Индивидуализация обучения.
- Ответственность за собственные успехи.
- Сотрудничество.
- Нацеленность на непрерывное образование.

Переход к открытому образованию может осуществляться следующим образом.

- Необходимо научить студента быть самостоятельным в обучении. Для этого обучение непременно должно быть активным, побуждающим студентов к самостоятельной постановке вопросов, самостоятельному выбору тем.
- Студенты должны быть вовлечены в процесс живого межиндивидуального общения, активно затрагивающего изучаемые темы.
- Каждый этап образовательного пути должен быть индивидуализирован.
- Необходимо ввести в обучение этап рефлексии — размышления студента над полученной информацией, ее сопоставления с ранее известными вещами, оценивания пройденного пути и полученных знаний.

Важным шагом в сторону открытого образования является создание специальной *образовательной среды*, которая подталкивает студента к включению в социальные институты изучаемой области и к активному взаимодействию внутри них, обеспечивает наличие множества возможных путей реализации учебного процесса и свободу их выбора. На важность такой среды для качественного обучения студентов указывал еще академик П. Л. Капица в своем письме Сталину в 1946 году о создании Физтеха: «Ведение воспитания с первых же шагов в атмосфере технических исследований и конструктивного творчества с использованием для этого лучших лабораторий страны» [5,6].

## **Открытое ПО в открытом образовании**

М. А. Лаврентьев писал: «Существует множество "путепроводов", по которым научные идеи вливаются в промышленность. Формы сотрудничества науки с производством также требуют творческого научного подхода, изобретательности и выбора оптимального решения в каждом конкретном случае» [7]. В ИТ важнейшим таким путепроводом в последнее время стало открытое и свободное программное обеспечение (СПО). Преимущества использования СПО в высших учебных заведениях и, в особенности, при обучении студентов информационным технологиям активно обсуждаются [8-13]. На первом месте традиционно указывается возможность изучения внутреннего устройства СПО, появляющаяся ввиду доступности его исходного кода. Но это далеко не единственное достоинство СПО. Проекты по разработке СПО часто являются *открытыми*, т.е. придерживаются открытого процесса разработки, в ходе которого все обсуждения по принятию проектных решений, развитию архитектуры ПО и разработке проектной документации проводятся публично. Все это в комплексе создает прекрасный материал и готовую инфраструктуру для обучения студентов программной инженерии и информационным технологиям в целом.

Использование открытых проектов по разработке ПО может служить основой для построения открытой образовательной среды в области программной инженерии. В частности, это позволяет разработчикам учебного курса получить практически для любой темы в данной дисциплине следующие инструменты.

- Готовый учебный материал: требования к ПО, спецификации его архитектуры, исходный код, работающее ПО, уже решенные и все еще актуальные проблемы и дефекты, возникшие в ходе его разработки, обсуждения разработчиками различных проектных решений и проблем, часто с развернутой аргументацией.
- Готовую инфраструктуру для ведения практических занятий: реальные проекты по разработке ПО, описание их проблем и дефектов, которые можно устранять в рамках заданий, доступное общение с реальными командами разработчиков.
- Возможность создания сильной мотивации для студентов в их обучении и практической деятельности, связанной с их участием в реальных проектах, имеющих общественную значимость.

Конечно, в качестве учебных материалов можно просто взять документы и код одного или нескольких крупных проектов. Но, помимо этого, существование большого разнообразия проектов по разработке СПО позволяет обеспечить беспрецедентную свободу выбора: студент может самостоятельно, на основе своих предпочтений и собственных представлений о карьере, выбирать предметную область проекта, используемые технологии и инструменты разработки. Такая свобода является особенно важной в контексте принципов *открытого образования* [4], которое требует предоставления предельной самостоятельности в обучении, в частности, в выборе собственной образовательной траектории.

Материалы открытых проектов служат отличной доступной базой знаний в различных разделах информационных технологий, своего рода аналогом классической публичной библиотеки, благодаря которой можно повысить активность и самостоятельность студентов в

учебном процессе, а также получить больше возможности для индивидуализации образовательного процесса. Кроме того, предоставляя студентам свободу выбора предметной области, языков программирования и применяемых технологий в рамках изучения отдельной темы, можно повысить личную заинтересованность и мотивированность, а также уровень ответственности за самостоятельно сделанный выбор.

## **Открытое образование в ИСП РАН**

Институт системного программирования РАН (ИСП РАН) является базовой организацией для кафедры системного программирования ВМиК МГУ им. М. В. Ломоносова и одноименной кафедры МФТИ. В ИСП РАН разработана *рамочная структура практических курсов* для обучения студентов по направлению ИТ, основанная на принципах открытого образования и использующая возможности, предоставляемые открытыми проектами по разработке СПО.

В рамках обучения к каждому студенту прикрепляется личный куратор, т.е. преподаватель, основные обязанности которого — поддерживать постоянный контакт со студентом, отвечать на его вопросы самому или указывать необходимые источники информации. К одному куратору прикрепляется не более 4-х студентов. Параллельно с практическим курсом должен проводиться теоретический курс по соответствующей теме, при этом важно, чтобы промежуток времени между лекцией по какой-то теме и использованием материала этой лекции на практике был небольшим. Возможны и опережающие практические задания, в ходе которых студентам нужно использовать материал лекций, которые будут прочитаны в промежутке между выдачей им заданий и контролем их выполнения.

Основные элементы предлагаемой структуры практикума следующие.

- **Выбор открытого проекта по разработке ПО.** Студенту предлагается самостоятельно выбрать проект для своей работы, удовлетворяющий рекомендациям разработчиков курса. Куратор помогает студенту в этом выборе, ничего не навязывая, и проверяет соответствие выбираемого проекта предъявляемым требованиям.
- **Знакомство с проектом.** Студент детально знакомится с выбранным проектом, обращая особое внимание на аспекты, связанные с изучаемым курсом. По завершении этого этапа он проводит презентацию своего проекта преподавателям и другим участникам курса, в ходе которой могут задаваться любые вопросы. Преподаватели особое внимание обращают на умение студента сопоставлять теоретический материал курса с увиденным им на практике, дополнительно проверяя это умение с помощью вопросов. В дополнение к презентации от студента можно потребовать подготовить какой-либо документ о проекте с заранее определенной структурой.
- **Практические задачи.** В рамках курса готовится набор задач, покрывающих все аспекты изучаемого предмета. Каждая задача состоит в выполнении практического задания с определенной формой отчетности и имеет некоторый весовой коэффициент. Студенту предоставляется возможность самостоятельного выбора задач, так, чтобы сумма их весов была выше обязательного порога. Задания могут зависеть друг от друга, и для выполнения одного может потребоваться сначала выполнить другое.

При подготовке практических заданий важно учитывать дополнительные возможности, предоставляемые открытыми проектами и существенно помогающие процессу обучения.

Для оценки результативности работы студента нужно использовать реакцию команды проекта и сообщества, существующего вокруг него. Это позволяет значительно повысить мотивацию студента к обучению и учит его оценивать свои результаты более объективно.

Необходимо развивать социальные навыки студентов, стимулируя их к активному общению с другими членами проекта, а также к обсуждению деталей проекта с преподавателями и другими студентами. Такое общение постепенно учит студентов эффективно общаться на

профессиональные темы и углубляет их понимание предмета.

Свобода выбора заданий в ходе обучения также способствует повышению мотивации, более сознательному отношению студента к своему обучению [10,12]. Она обеспечивает индивидуальный выбор глубины и методов погружения в материал курса и предоставляет студенту возможность самостоятельно формировать свои знания и навыки, исходя из его предпочтений и потребностей его личностного развития.

На основе предложенных принципов нами разработан курс практических занятий по дисциплине "Технологии программирования", который проходит апробацию на кафедре системного программирования ВМиК МГУ и в рамках научно-исследовательской работы на кафедре системного программирования МФТИ. Параллельно первым двум этапам практикума читается теоретический курс "Технологии программирования. Компонентный подход" [14], кроме того практические занятия предваряет дополнительная лекция, посвященная принципам открытого программного обеспечения.

Первоначально мы не ставили никаких ограничений при выборе открытого проекта. Единственным обязательным требованием является открытость не только исходного кода, но и всего процесса разработки, проектных документов и обсуждений. Получив первый опыт работы студентов с материалами открытых проектов, мы осознали, что для обучения программной инженерии нужно использовать достаточно зрелые и достаточно большие проекты (более 25000-30000 строк кода на любом из языков высокого уровня). Это необходимо, чтобы в проекте проявлялись проблемы, специфичные для разработки сложных программных систем, а также для возможности изучать принимаемые в проекте решения в их развитии.

На выбор и знакомство с проектом отводится два месяца, по завершению которых студенты проводят презентации, представляя выбранный проект и отвечая на вопросы из заранее известного им списка. Ключевая идея в организации списка вопросов заключается в том, чтобы вынудить студентов найти в выбранном проекте отражение идей и подходов, рассматриваемых в теоретическом курсе. В качестве дополнительного задания при знакомстве с проектом перед студентами ставится задача подготовить один из проектных документов по заранее заданному шаблону (описание концепции проекта, техническое задание к некоторой подсистеме, описание архитектуры, план тестирования и пр.).

Практические задачи включают в себя около 30-ти заданий, покрывающих различные области программной инженерии:

- моделирование предметной области;
- работа с требованиями;
- архитектура ПО;
- обеспечение и контроль качества ПО;
- отладка и исправление ошибок;
- планирование и внесение изменений;
- разработка пользовательской документации;
- поддержка пользователей;
- комплексные задачи.

Первый опыт работы привел к следующим выводам: студенты на третьем-четвертом курсе обучения плохо знакомы с принципами открытого и свободного ПО, и это направление не является популярным. При самостоятельном выборе проектов студенты склоняются к проектам, находящимся на начальной стадии, так как считают, что в такие проекты им будет проще вписаться. Возможность выбора предметной области проекта, действительно, существенно повышает заинтересованность студентов в выполнении последующих заданий. Знакомство с выбранными проектами по разработке СПО показывает, что эти проекты могут быть использованы как для демонстрации грамотных подходов к разработке ПО, так и для

демонстрации часто встречающихся ошибок, решений и организационных элементов, которых необходимо избегать в проектах по разработке ПО.

## **Заключение**

В работе представлен подход к организации учебных курсов по информационным технологиям с использованием материалов и возможностей личного участия, предоставляемых открытыми проектами по разработке программного обеспечения. Описана рамочная структура практических курсов, опирающаяся на участие студентов в одном из таких проектов, свободно выбираемом ими в соответствии со своими предпочтениями и целевыми установками, а также ограничениями, устанавливаемыми разработчиками учебного курса. Таким образом реализуются основные элементы открытой образовательной среды, необходимой для выработки у студентов навыков самостоятельной деятельности в выбранной ими области специализации, эффективного общения в рамках профессионального сообщества и адаптации получаемых знаний под потребности решаемых задач.

Описываемый подход используется в ИСП РАН не так давно, и поэтому пока отсутствуют статистические данные, иллюстрирующие эффект его использования. Однако личное общение со студентами демонстрирует заметное повышение у них мотивации к обучению, отмечаемое многими другими авторами, использовавшими открытые проекты в ИТ-образовании [10,12], а также большую самостоятельность при принятии решений.

## **Литература**

- [1] С. А. Христианович, М. А. Лаврентьев, С. А. Лебедев. Назревшие задачи организации научной работы. Правда, 14.02.1956.
- [2] Н. Л. Добрецов. «Треугольник Лаврентьева»: принципы организации науки в Сибири. Вестник Российской академии наук, т. 71, № 5, 2001, стр. 428–436.
- [3] Commission of the European Communities. A Memorandum on Lifelong Learning. Brussels, 2000. См. <http://ec.europa.eu/education/policies/III/life/>.
- [4] C. Wedemeyer. Characteristics of open learning systems. In Open Learning Systems, Washington, National Association of Educational Broadcasters, 1974.
- [5] П. Л. Капица. Письма о науке. М., 1989.
- [6] <http://museum.phystech.edu/books/book-museum/chapter4.html>.
- [7] М. А. Лаврентьев. Магистраль сибирской науки. Известия, 13.02.1971.
- [8] K. J. O'Hara, J. S. Kay. Open source software and computer science education. J. Comput. Small Coll., 18(3):1–7, 2003.
- [9] Allen, E.; Cartwright, R.; Reis, C. Production programming in the classroom. Proc. of the 34-th SIGCSE technical symposium on Computer science education, Reno, Nevada, USA, pp. 89-93, 2003.
- [10] D. Carrington, S.-K. Kim. Teaching software design with open source software. Proc. of 33-rd ASEE/IEEE Frontiers in Education Conf., pp. 9-14, November 2003.
- [11] C. P. Fuhrman. Appreciation of software design concerns via open-source tools and projects. Proc. of 10-th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts, at ECOOP 2006, Nantes, France, July 2006.
- [12] M. Pedroni, T. Bay, M. Oriol, A. Pedroni. Open source projects in programming courses. ACM SIGCSE Bulletin, 39(1):454-458, March 2007.
- [13] <http://www.flosscom.net>.
- [14] В. В. Кулямин. Технологии программирования. Компонентный подход. М., ИНТУИТ-Бином, 2007.