

Открытые системы №08, 2006**Computing Curricula: Software Engineering и российское образование***Андрей А. Терехов, Андрей Н. Терехов*

В 2004 году совместный комитет по образованию сообществ ACM и IEEE Computer Society выпустил документ "Computing Curricula 2001: Software Engineering", содержащий рекомендации по преподаванию программной инженерии. Особенностью этого документа является наличие рекомендаций по адаптации учебных программ к различным условиям преподавания и специфике отдельных стран. К сожалению, среди множества шаблонов составления учебных программ нет шаблона, учитывающего особенности российского образования.

Документ Computing Curricula 2001: Software Engineering является частью начатого в 1968 году международного проекта Computing Curricula, в рамках которого ИТ-специалисты из различных стран мира работают над созданием рекомендаций по преподаванию информационных технологий в высших учебных заведениях. Данные рекомендации обновлялись примерно каждые десять лет, однако в конце 90-х годов стало ясно, что информационные технологии слишком трудно, если вообще возможно, полностью осветить в рамках одного университетского курса. В связи с этим было принято решение о разделении курса на четыре дисциплины: информатика (computer science), программная инженерия (software engineering), проектирование аппаратных платформ (hardware engineering) и информационные системы (information systems). Первый том, посвященный информатике, был выпущен в конце 2001 года, русский перевод этого документа был опубликован годом позже [1]. В качестве официальных рекомендаций по преподаванию информационных систем был утвержден документ Information Systems 2002, разработанный в результате совместного проекта ACM, AIS (Association for Information Systems) и АИТР (Association of Information Technology Professionals). Рекомендации по преподаванию программной инженерии были выпущены в августе 2004 года. Наконец, документ с рекомендациями по преподаванию проектирования аппаратных платформ был утвержден в декабре 2004 года.

В данный момент разворачивается проект по созданию следующей версии рекомендаций Computing Curricula, который будет носить название Computing Curricula 2005, и пока выпущен только один обзорный том (www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/CC2005-March06Final.pdf), в нем впервые сформулирована потребность в выделении из общей области "вычислительные науки" еще одной самостоятельной дисциплины под названием "информационные технологии".

Особенностью документа Computing Curricula: Software Engineering (далее – CC:SE) является наличие рекомендаций по адаптации учебных программ к различным условиям преподавания. Например, преподавание программной инженерии может вестись как на факультете информатики, так и в контексте преподавания инженерных дисциплин. Кроме того, в CC:SE приведены примеры адаптации рекомендаций к специфике отдельных стран (США, Великобритания, Япония, Израиль, Австралия), но нет шаблона, учитывающего особенности российского образования. Нам хотелось проверить, возможно ли наложение требований CC:SE на российские образовательные стандарты, и если это возможно, то предложить вариант решения российским преподавателям. В качестве потенциальных кандидатов на соответствие требованиям Computing Curricula: Software Engineering мы рассмотрели существующие учебные специальности 351500 и 010400, а также проект стандарта по преподаванию программной инженерии, предложенный Высшей школой экономики (ВШЭ).

В Санкт-Петербургском государственном университете ИТ традиционно преподаются на базе специальности 351500 "Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей", которая является развитием более ранней учебной специальности 220400, предложенной еще в 1985 году. К сожалению, у программы обучения 351500 есть ряд "наследственных" недостатков, например, на младших курсах избыточное, с нашей точки зрения, внимание уделяется математике, а на первых пяти семестрах на обучение собственно программированию отводится всего четыре-шесть часов в неделю! В связи с этим выполнить в рамках специальности 351500 требования CC:SE трудно, если вообще возможно.

В 2006 году мы приняли участие в экспертизе проекта стандарта по программной инженерии, предложенного ВШЭ. Это вполне добротная программа, составленная в соответствии с основными требованиями CC:SE, однако в проекте ВШЭ наблюдается перекоп в другую сторону по сравнению с 351500 — слишком мало математики, зато очень много экономики, маркетинга и менеджмента. Напротив, в CC:SE явным образом подчеркивается важность математического образования, а также формулируется связь программной инженерии с различными разделами математики и информатики. Поэтому мы полагаем, что программа, предложенная ВШЭ, будет хороша для экономических факультетов и вузов, а также для вузов, ориентирующихся на MBA с уклоном в ИТ, но вряд ли окажется востребована математическими и техническими факультетами.

Наилучшим кандидатом на роль базового стандарта образования в области информатики и программной инженерии мы считаем направление подготовки бакалавров и магистров 010400 "Информационные технологии" (до 2006 года 511900) [2, 3]. Это направление является первым российским образовательным стандартом,

разработанным с учетом рекомендаций "Computing Curricula". С 2003/04 учебного года обучение по этому направлению ведут МГУ, СПбГУ, ННГУ, МИЭМ, СПбГЭТУ и МАТИ. Примерный учебный план подготовки бакалавров по направлению 010400 представлен в [таблице 2](#).

Наш метод приведения учебного плана 010400 в соответствие рекомендациям CC:SE можно разбить на три последовательных этапа.

1. Сокращение учебной нагрузки за счет курсов, не входящих в список обязательных областей знаний по программной инженерии в CC:SE.
2. Сопоставление базовых курсов учебной специальности 010400 с Computing Curricula: Computer Science.
3. Выбор наиболее подходящей для России методики преподавания программной инженерии из нескольких вариантов, предложенных в CC:SE.

Сокращение учебной нагрузки

Уже при первом взгляде на учебный план можно заметить, что он подразумевает чрезмерную учебную нагрузку на студентов - 36 часов в неделю в третьем семестре и особенно 40 часов в неделю в четвертом - это слишком много. С нашей точки зрения, при выборе курсов для сокращения необходимо руководствоваться следующим критерием – освещает ли данный курс вопросы, входящие в *[[[Обязательный набор знаний по программной инженерии]]]* (Software Engineering Education Knowledge, SEEK)?

Исходя из такой постановки вопроса, мы сократили следующие курсы.

- В третьем и четвертом семестре отведено по 4 часа в неделю на физику - весьма далекую от программирования науку. В специальности 351500 мы уже несколько лет обходимся без нее.
- В третьем семестре предусмотрено 4 часа в неделю на курс "Языки программирования", однако столь подробно рассказывать студентам про конкретные языки программирования в лекционном курсе не стоит. Вместо этого достаточно познакомить студентов с основными парадигмами программирования (объектно-ориентированное, функциональное и т.д.) в рамках базовых курсов. Именно такой подход сформулирован в CC:SE: "необходимо подчеркивать основные и постоянные принципы программной инженерии, а не детали современных или конкретных средств". Кроме того, студенты в любом случае должны детально ознакомиться с несколькими языками программирования и закрепить эти знания на практике.
- Некоторое недоумение у нас вызвало включение в обязательную программу специальности 010400 курса "Неклассические логики" (4 часа в неделю в шестом семестре). По нашему мнению, такой предмет может быть прочитан в каком-нибудь спецкурсе по выбору студента.

Сопоставление базовых курсов 010400 с CC:CS

Прежде всего необходимо сказать о принятой в Computing Curricula системе обозначений курсов. Наименование всегда начинается с двухбуквенного кода, указывающего на принадлежность курса к тому или иному образовательному блоку учебной программы (SE для программной инженерии, CS для информатики, MA для математики и NT для нетехнических дисциплин). Далее идет цифра от 1 до 4, указывающая на год обучения, в котором ведется преподавание данного предмета. Следующие две цифры служат преимущественно для различения курсов между собой. Наконец, некоторые курсы обладают суффиксом, состоящим из одной буквы и обозначающим модель преподавания, в рамках которой используется данный курс. Например, суффикс "I" указывает на подход, основанный на преподавании императивного программирования, а "O" – на объектно-ориентированный подход.

Поскольку мы слегка расчистили первоначальную программу 010400, мы сочли возможным добавить традиционный для отечественной школы преподавания информатики и важный с точки зрения CC:SE курс CS103 "Алгоритмы и структуры данных" (третий семестр, 2 часа в неделю). Кроме того, мы включили в программу обучения незаслуженно забытый курс CS240 "Трансляция языков программирования" (шестой семестр, 4 часа в неделю). Мы полагаем, что знания, изложенные в данных курсах, являются базовыми и необходимыми для любого профессионального программиста.

Выбор наиболее подходящей методики преподавания программной инженерии

Более трудной работой оказалось совмещение требований 010400 с CC:SE. Документ Computing Curricula: Software Engineering содержит подробное описание целого ряда курсов, которые могут преподаваться в рамках обучения бакалавров по программной инженерии. Кроме того, отдельная глава посвящена изложению различных подходов к преподаванию программной инженерии на младших курсах. Таким образом, CC:SE предоставляет богатый выбор продуманных учебных материалов и траекторий обучения. В данном разделе мы покажем, как эти рекомендации могут быть применены при проектировании конкретного учебного плана по программной инженерии.

Один из первых вопросов, возникающих при подготовке учебного плана по программной инженерии, - с чего следует начинать преподавание: с собственно программной инженерии или с информатики? В первом случае

акцент делается на инженерную составляющую, и студенты с самого начала учатся мыслить в терминах решаемых задач, формулирования требований, проектирования, ресурсных ограничений. При таком подходе студенты с самого первого дня готовятся к проектной работе и взаимодействию с другими членами команды. Во втором случае основной упор на младших курсах делается на преподавание теоретических основ информатики, являющейся базовой наукой для программной инженерии, и на получение студентами навыков "программирования-в-малом". Только после овладения этими базовыми навыками студенты переходят к изучению программной инженерии как таковой.

Необходимо отметить, что сегодня нет свидетельств безусловного превосходства одного из этих подходов, поэтому выбор во многом является делом вкуса или удобства в конкретной ситуации. Мы полагаем, что в России будет востребован второй вариант. Дело в том, что информатика у нас преподается уже несколько десятков лет, тогда как программная инженерия находится лишь в стадии становления. Таким образом, следует ожидать, что в ближайшее время подавляющее большинство учебных программ по программной инженерии будет строиться именно на базе факультетов и кафедр информатики. Поэтому мы выбрали последовательность преподавания, в которой первыми тремя "профильными" курсами являются *базовые курсы по информатике* CS101→CS102→CS103. В этом варианте первый курс по программной инженерии, SE201, появляется лишь в параллели к последнему курсу по информатике, CS103.

Возможен и другой, предложенный в CC:SE, вариант, в котором тройка курсов CS101, CS102, CS103 сжимается в два - CS111 и CS112, но такой подход вредит восприятию студентами базового материала. Кроме того, вводные курсы по информатике иногда используются преподавателями для подтягивания более слабых студентов до среднего уровня знаний, что крайне трудно сделать при интенсивном изложении материала. Вводные курсы – не лучшее место для экономии.

Еще одна интересная деталь, связанная с базовыми курсами по информатике, заключается в том, что документ «Computing Curricula: Computer Science» предлагает шесть различных подходов к преподаванию двух первых курсов CS101-102: с ориентацией на императивное программирование, объектно-ориентированное программирование, функциональное программирование, с ориентацией на алгоритмы, подход "от аппаратуры" или подход с максимальным охватом материала. Авторы CC:SE отмечают, что с точки зрения преподавания программной инженерии наиболее подходящими являются императивный или объектно-ориентированный подход. Мы отдали предпочтение первому варианту, как более традиционному. Таким образом, в предложенной нами программе фигурируют курсы CS101 и CS102.

Далее необходимо определиться с основными курсами по программной инженерии, преподаваемыми на втором и третьем курсах. CC:SE предлагает две схожие последовательности изложения, которые отличаются лишь в распределении материала по конкретным курсам. Обе последовательности позволяют полностью покрыть обязательный набор знаний по программной инженерии (SEEK). Для обеих последовательностей разработаны подробные описания, включающие формальный календарный план, требования к слушателям, задачи обучения, преподаваемые модули, отображение на SEEK и другие материалы. Необходимо отметить, что обе последовательности курсов определены не полностью и оставляют некоторое количество учебных часов свободными, что позволяет учебным учреждениям и преподавателям легко адаптировать их к своим потребностям. Мы выбрали последовательность преподавания, в которую входят следующие курсы:

- SE211 Конструирование программного обеспечения;
- SE212 Подход программной инженерии к человеко-машинному взаимодействию;
- SE311 Проектирование и архитектура программного обеспечения;
- SE321 Обеспечение качества и тестирование программного обеспечения;
- SE322 Анализ требований к программному обеспечению;
- SE323 Управление программными проектами.

Основное отличие альтернативной последовательности преподавания заключается в том, что в ней материал по формальным методам в программной инженерии выделен в самостоятельный курс. С нашей точки зрения, лучше иметь отдельный курс по требованиям к программному обеспечению, чем отдельный курс по формальным методам. Мы полагаем, что анализ требований все-таки чаще применяется на практике, чем формальные методы. В любом случае, базовый материал и по требованиям, и по формальным методам содержится в обоих пакетах, дело только в расстановке акцентов и уровне подробности.

Последним шагом в формировании учебного плана является распределение курсов по годам и семестрам. В CC:SE предусмотрено несколько шаблонов, описывающих варианты преподавания в различных странах и университетах. Например, в некоторых университетах Великобритании и США учебный год делится не на два семестра, как в России, а на три триместра.

На наш взгляд, российским традициям лучше всего соответствует шаблон N2S-1, рекомендованный для университетов Северной Америки, причем его вариант N2S-1с, ориентированный на факультет информатики (Таблица 1), а не альтернативный вариант N2S-1е для инженерных факультетов.

Таблица 1. Шаблон N2S-1с

1 курс		2 курс		3 курс		4 курс	
1 семестр	2 семестр	3 семестр	4 семестр	5 семестр	6 семестр	7 семестр	8 семестр
CS101	CS102	CS103	CS (Int)	CS (Int)	CS (Int)	Дипломный проект	
Мат.анализ	Мат.анализ	CS106	SE A	MA 271	SE D	SE F	Tech Elect
NT 181	CS 105	SE 201	SE B	SE C	SE E	Tech Elect	Tech Elect
--	--	NT 272	--	NT 291	Tech Elect	--	--
--	--	--	--	--	--	--	--

Данный шаблон типичен для учебных планов по программной инженерии, которые могут преподаваться на базе факультетов информатики. Преимуществом такой программы является то, что базовые курсы по информатике могут преподаваться одновременно как студентам, обучающимся информатике, так и студентам, изучающим программную инженерию. Серым фоном в шаблоне выделены курсы, посвященные собственно программной инженерии. Обучение программной инженерии начинается с курса SE201, проходящего параллельно последнему вводному курсу по информатике, CS103, и продолжается последовательностью из 6 основных курсов (SE A-F), которые мы перечислили ранее. Кроме того, в шаблон входит три нетехнических курса для покрытия важных областей SEEK.

В результате, у нас получилось следующее предложение по преподаванию программной инженерии в российских университетах на базе специальности 010400 (*таблица 3*):

- В первом семестре добавился курс NT181 по групповой динамике (психологии работы в коллективе). Умение работать в группе, знание основ психологии программирования необходимо студентам с самого начала обучения. Для курса NT291 по профессиональной деятельности в российской учебной программе 010400 нашелся хороший аналог - курс по социальным и этическим вопросам (в четвертом семестре).
- В третий семестр добавляются курсы SE201 - начальный курс по программной инженерии и NT272 - по инженерной экономике. В четвертый семестр добавляются курсы SE211 "Конструирование программного обеспечения" и SE212 "Подход программной инженерии к человеко-машинному взаимодействию". Чтобы уменьшить нагрузку, в четвертом семестре вполне можно пожертвовать факультативом (4 часа) или практикумом (4 часа). Мы предпочли аннулировать практикумы в 4-6 семестрах. К этому времени студенты уже должны уметь программировать.
- В пятый семестр мы добавили курс SE311 "Проектирование и архитектура программного обеспечения", а в шестой - еще два курса, входящих в основной набор по программной инженерии: SE321 "Обеспечение качества и тестирование программного обеспечения" и SE322 "Анализ требований к программному обеспечению". Последние два курса в нашем варианте подменяют собой единственный в шестом семестре курс по программной инженерии, на который в типовом плане учебной специальности 010400 отводится 4 часа в неделю.
- Последний курс по программной инженерии - SE323 "Управление программными проектами" - поместился только в девятом семестре, несмотря на то, что его обозначение означает, что он должен читаться на третьем курсе.
- Завершает программу обучения дипломный проект (SE400).

Заключение

Мы представили конкретный пример адаптации рекомендаций документа «Computing Curricula: Software Engineering» к условиям российского образования. Мотивацией стало открытие в 2006/07 учебном году специальности 010400 на математико-механическом факультете Санкт-Петербургского государственного университета, преподавание по которой будет вестись по изложенному учебному плану. Однако предложенный нами вариант преподавания далеко не единственный, и при составлении учебных планов всегда необходимо учитывать специфику конкретного университета. Поэтому мы рекомендуем университетам не столько результат нашего анализа, сколько методику построения учебных планов с учетом требований международных стандартов.

Надеемся, что статья послужит отправной точкой для дальнейшего обсуждения методик преподавания программной инженерии в России и массового внедрения данной специальности в российских университетах.

Андрей А. Терехов (andreyte@microsoft.com) – менеджер академических программ по Центральной и Восточной Европе, Microsoft (Мюнхен), Андрей Н. Терехов (ant@tercom.ru) – зав. кафедрой системного программирования Санкт-Петербургского государственного университета (Санкт-Петербург).

Литература

1. "Рекомендации по преподаванию информатики в университетах", Издательство С.-Петербургского университета, 2002 (под ред. В.Л. Павлова и А.А. Терехова).
 2. В.А. Сухомлин, "ИТ-образование: концепция, образовательные стандарты, процесс стандартизации". М.: Горячая линия - Телеком, 2005.
 3. В.А. Сухомлин, В.В. Сухомлин, "*Концепция нового образовательного направления*", *Открытые системы*, 2003, № 2.
-

Постоянный URL статьи: <http://www.osp.ru/os/2006/08/3282281/>

© «Открытые системы», 1992-2011. Все права защищены.