

Проведение международных студенческих школ по программной инженерии с использованием гибких методологий

Кринкин К. В.¹, Чернокульский В. В.², Самойленко В. П.³, Размочаева Н. В.⁴

¹Кринкин Кирилл Владимирович / Krinkin Kirill Vladimirovich - кандидат технических наук;

²Чернокульский Владимир Викторович / Chernokulskiy Vladimir Viktorovich - кандидат технических наук;

³Самойленко Владимир Петрович / Samoilenko Vladimir Petrovich – кандидат технических наук, доцент;

⁴Размочаева Наталья Владимировна / Razmochaeva Natalia Vladimirovna - кандидат технических наук, доцент,

кафедра математического обеспечения и применения ЭВМ,
факультет компьютерных технологий и информатики,

Санкт-Петербургский государственный электротехнический университет имени В. И. Ульянова
(Ленина), г. Санкт-Петербург

Аннотация: в статье анализируется опыт проведения студенческих школ по программной инженерии с использованием гибких методологий разработки; предлагается адаптированная к интернациональной среде методика обучения.

Ключевые слова: программная инженерия, гибкие методики разработки программного обеспечения.

Введение

Программная инженерия является одной из наиболее востребованных специальностей настоящих дней. Подготовку по соответствующим дисциплинам ведет большое количество образовательных учреждений. Сложность подготовки студентов по данной специальности обуславливается следующими факторами:

- крайне высокая скорость изменений компьютерных и программных технологий [1], заставляющая постоянно модернизировать учебные программы, для того чтобы успевать за меняющимися условиями;
- существенно возросшие требования индустрии к коммуникационным навыкам участников разработки программного обеспечения, к так называемым «soft» skills [2], – то есть, эффективность разработчиков определяется не только техническими, но и гуманитарными умениями;
- распределенный характер исполнения многих программных проектов (примерами могут служить многие проекты с открытым программным кодом, использующие усилия сообщества, такие как Linux), требующий поддержания определенного уровня процесса создания программного продукта и владения инструментами коллективной разработки.

Развитие навыков, перечисленных выше, является актуальной задачей в рамках подготовки профессиональных разработчиков. Хорошо зарекомендовавшей себя технологией развития командной работы является проведение краткосрочных школ по программной инженерии. В данной статье мы обобщаем опыт, полученный при проведении двух российско-немецких международных студенческих школ [3, 4], и обсуждаем методику их организации.

Цель школы и принципы формирования команд

Основной целью проведения школы является создание такой среды для обучаемых, в которой они вынуждены построить процесс разработки программного проекта «с нуля», а также продемонстрировать свои способности по самоорганизации. В нашем случае, мы усложняем задачу построения команды, помещая студентов в интернациональную среду, вводя следующее требование – каждая команда должна состоять из студентов Германии и России. Таким образом, общение между участниками будет осуществляться на английском языке, носителем которого не являются члены команды.

В рамках проведения первой школы в 2012 году было собрано три интернациональных команды, состоящих из немецких и русских студентов, в рамках проведения второй школы в 2016 году, команд было собрано 5 команд. Время, отведенное на разработку проекта, составило 4 дня в первом случае, и 5 дней – во втором.

Выбор и распределение проектов для разработки

Изначально, организаторами обеих сторон (русской и немецкой) выдвигается несколько конкретных идей для проектов, которые тестируются на предмет возможности реализации в ограниченный промежуток времени. Для того чтобы создать условия максимальной заинтересованности каждого разработчика в выполняемом проекте, участники (включая студентов) вовлекаются на самом начальном этапе в процесс формирования тем проектов. Кроме того, в начале школы проводится сессия мозгового штурма, на которой идея каждого проекта должна быть кратко представлена.

Распределение команд по проектам производится в режиме живого голосования. Проекты записываются на доске, а участникам раздаются стикеры, на которых они пишут свою фамилию и приоритет (от 1 до 3). Каждый имеет возможность проголосовать за участие в проекте, приклеивая один из своих стикеров на доску рядом с наименованием проекта. Последняя фаза – балансировка участников

проектов в командах. Каждый мог подать заявку для участия в любом из выбранных проектов, но было выдвинуто несколько ограничений — команды должны содержать как минимум по одному представителю из России и Германии, и в проекте не может быть больше 5 и меньше 3 человек. Таким образом, после совещательного процесса создаются сбалансированные и максимально мотивированные команды.

Управление требованиями

Ни для кого не секрет, что требования - это краеугольный камень в разработке программного обеспечения. Их сложно сформулировать, донести до каждого члена команды и следить за изменениями и меняющимися условиями. Множество всемирно признанных авторов, таких как Ф. Брукс (F. Brooks), Т. Де Марко (T. De Marco), Дж. Спольски (J. Spolsky), предлагают модели управления требованиями в команде, однако, необходимой является отработка их на практике.

Мы применили две схожие методики управления требованиями. В 2012 году сбор и разработка требований выполнялась в виде создания видеоролика демонстрирующего готовый продукт. Этот ролик сразу позиционировался как рекламный, предназначенный для продажи продукта. В его создании участвовал каждый член команды, а, следовательно, его появление – это, с одной стороны, результат командной переработки требований, а с другой – идентификация основной функции (юзкейса), ради которого создается продукт. Во втором случае мы ограничились разыгрыванием сценки членами команды, то есть применили методику Software Theater, разработанную нашими немецкими коллегами [5].

Процесс разработки

Разработка программного продукта в рамках школы основывается на гибких методологиях (agile), которые адаптированы к условиям школы. Участники имеют очень ограниченное время для разработки и обучения, поэтому проводится небольшое количество scrum-встреч.

Первая встреча – коллективная, – участники всех проектов слушают, как каждая команда отчитывается о своем статусе, блокирующих проблемах, и берет обязательства. Такая коллективная форма позволяет командам выровняться друг относительно друга – соотнести скорость своего проекта со скоростью соседних проектов. Отстающие начинают ускоряться, видя, что у соседней команды больше прогресс, а преуспевающие начинают ускоряться также, поскольку их вдохновляет собственный успех.

Второй вид встреч – проектные. Только одна команда в выделенном тихом помещении без посторонних глаз анализирует статус. В этом случае команда более склонна открыть и обсудить проблемы, существующие в проекте, найти их решения.

Третий вид встреч, применяемый в рамках школы, – попытка реализации scrum of scrum. В данном случае каждая команда самостоятельно определяет свой статус, и выделенный участник рассказывает о статусе команды на своего рода мета-встрече. Это упражнение дает командам представление того, как процесс может быть отмасштабирован в рамках больших проектов

Последняя встреча – презентация результатов разработки. Проводится в конце школы в формате демонстрационной сессии. Каждая команда готовит презентацию проекта и демонстрирует его реальную работу. Также немаловажным является обсуждение проблем и новых навыков, полученных в процессе школы.

Заключение

Анализируя результаты двух проведенных школ, можно обнаружить явные аналогии процесса обучения любой дисциплине с процессом разработки проекта в небольшой команде. В работах Л. С. Выготского [6] вводятся пороги обучаемости и зона комфортного обучения. Обучение и разработка программного обеспечения – такие процессы, которые могут быть практически самоорганизуемыми, если ученики находятся в зоне комфорта. Нижняя граница зоны определяет интерес – наличие нерешенных нетривиальных задач стимулирует мозг к действию – мозг любит учиться. Верхняя же граница зоны определяется максимальными возможностями влияния самой команды. Если команда встречает сильную неопределенность в требованиях или отсутствие каких-то ресурсов, то скорость разработки проекта начинает уменьшаться. Во всех остальных случаях, находясь в зоне комфортного обучения или в комфортных условиях разработки, команда способна к самоорганизации. Следовательно, задача организаторов школы, создать максимально возможные условия для самоорганизации команд, отслеживая ситуации достижения верхней и нижней границы комфортного обучения.

Такой подход к обучению можно назвать Chaordic Teaching, позаимствовав идею у Dee Ward Hock [7], который, являясь директором крупной корпорации, применил такое слово, образованное из Chaos и Order (с англ. - хаос и порядок), обозначив границу между ними, считая, что именно на этой границе возможны различные инновации и открытия. Проведенные школы доказывают, что данный принцип работает как в сфере обучения, так и в сфере разработки.

Литература

1. The Joint Task Force on Computing Curricula, 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Technical Report. ACM. New York. NY. USA.
2. *D. González-Morales, L. M-M Antonio, J. L. R. García Teaching “Soft” Skills in Software Engineering // 2011 IEEE Global Engineering Education Conference (EDUCON). "Learning Environments and Ecosystems in Engineering Education". April 4 - 6, 2010. Amman Jordan. Pp 630-637. 978-1-61284-641-5 2011 IEEE.*
3. Joint Advanced Student School JASS 2012. [Electronic resource]. URL: https://www.bruegge.in.tum.de/lehrstuhl_1/projects/all-projects/413-jass-2012/ (date of access: 28.10.2016).
4. Joint Advanced Student School JASS, 2016. [Electronic resource]. URL: https://www.bruegge.in.tum.de/lehrstuhl_1/people/people-archive/42-projects/current-projects/738-jass-2016/ (date of access: 28.10.2016).
5. *H. Xu, S. Krusche, B. Bruegge Using Software Theater for the Demonstration of Innovative Ubiquitous Applications ESEC/FSE’15, August 30 – September 4, 2015, Bergamo, Italy ACM. 978-1-4503-3675-8/15/08 DOI:10.1145/2786805.2803207.*
6. *Выготский Л. С. Собрание сочинений: В 6-ти т. Т. 6. Научное наследство / Под ред. М. Г. Ярошевского. М.: Педагогика. С. 1984. 400 (Акад. пед. наук СССР).*
7. *D. Hock. “The chaordic organization: Out of control and into order”. World Business Academy Perspectives. Vol. 9. № 1. Pp. 5–18, 1995.*