

Внедрение гибких методологий разработки в процесс обучения

Александрова Н.А.¹, Шлюпкин П.В.²

¹ *aleksandrovan@bk.ru*, ² *pavel_shliupkin@bk.ru*

^{1,2} Саратовский государственный университет имени Н.Г. Чернышевского

В данной статье рассказывается о гибких методологиях разработки и внедрении их в обучающий процесс. Также представлены результаты исследования деятельности обучающихся после использования методологий. Упомянуты роли и функции участников процесса.

Ключевые слова: Scrum, Agile, гибкие методологии, процесс обучения.

В настоящее время все больше внимания уделяется вопросам формирования у обучающихся навыков, необходимых для функционирования в условиях парадигмы непрерывного образования. Для их формирования необходимы образовательные инновации, роль которых для учебных заведений возрастает в силу того, что в открытой системе образования они функционируют в весьма непростых условиях конкуренции.

Современная система высшего образования выпускает достаточно много специалистов, обладающих знаниями о современных информационных

технологиях. Тем не менее, образовательные курсы рассматривают каждого студента как изолированную единицу. Факультеты выпускают бакалавров и магистров по различным ИТ-специальностям, в процессе обучения, по которым студент общается и совместно работает над проектами только со студентами своей же специальности. Таким образом, будущий бакалавр или магистр не имеет профессионального взаимодействия со студентами смежных специальностей и, следовательно, не получает необходимых навыков работы в команде разработчиков ИТ-проектов. Кроме того, в современном образовательном процессе присутствует проблема «пустых» курсовых проектов, дипломных работ и диссертационных исследований. Возникает данная проблема в том случае, если результаты научной работы не имеют практического применения, или в ходе работы студенту приходится пропускать или имитировать некоторые этапы в силу отсутствия необходимых знаний или навыков.

С другой стороны, предприятия-потребители ИТ ресурсов требуют уже готовых специалистов, обладающих навыками работы в ИТ-отрасли, где взаимодействие между профессионалами различных специальностей достаточно сильно. Причем немаловажным, а во многих случаях решающим фактором становится наличие навыков взаимодействия с коллективом, работающим над общей проблематикой.

Главная концепция в обучении ИТ-специалистов – формирование сквозного курса («Технология программирования»), который позволит студентам применять приобретаемые ими знания, одновременно обучаясь работать в команде. Имеет смысл включать курс «Технология программирования» в учебную программу начиная со 2-го курса. На 2-м курсе студенты объединяются в команды по 2–3 человека и реализуют в течение семестра достаточно простое приложение, стараясь пройти основные этапы жизненного цикла разработки программного обеспечения: написание требований, разработка кода, тестирование. Это поможет студентам узнать различные области ИТ, а также понять свои предпочтения. Далее, на 3-м курсе студенты объединяются в группы по 5–7 человек и приступают к реализации нового проекта, уже более сложного и требующего определенных навыков. На 4-м курсе группа студентов из 15–20 человек приступает к выполнению нового большого проекта, самостоятельно выбирая его тему. Компании-партнеры вуза могут участвовать в формировании темы проекта, что позволит, в случае успешного завершения проекта, получить сформированную команду разработчиков. Студент же по окончании проекта получит опыт работы над решением актуальных задач, что позволит ему взвешенно выбрать направление специальности магистра, или покажет, что имеющиеся у него знания достаточны для успешной работы в отрасли.

Важной частью работы в проектной группе так же является профессиональное взаимодействие между участниками. При размере группы в 15–20 человек структура группы не является чрезмерно сложной, а, следовательно, отдельные ее члены не изолируются от остального коллектива. В ходе разработки участники группы, обучающиеся по разным специальностям,

будут получать информацию от коллег и осваивать основы взаимодействия. Кроме того, учитывая малый размер коллектива, они получают множество разносторонних знаний об информационных технологиях и смежных областях. Участие в проектах студентов, чья специализация наиболее полно соответствует роли в проекте, позволит уменьшить время дополнительного обучения в рамках выбранной проектной цели, а значит увеличить эффективное время разработки проекта, что, в конечном счете, существенно увеличит число успешно завершенных проектов. В ходе обучения следует применять новые технологии командной разработки программного обеспечения такие, как Agile. Современная методика управления проектами помогает максимально эффективно использовать навыки каждого в команде.

Agile и Scrum зародились в недрах управления IT-проектов, и это не случайно. Трудно представить себе другую область, так сильно зависимую от постоянных изменений: здесь и технические форс-мажоры, и постоянно меняющиеся требования заказчиков, наконец, бурное развитие технологий, требующее постоянного изучения разработчиками новых программ и анализа появляющихся новинок – все это привело к тому, что в конце XX века традиционная модель управления проектами в области разработки программного обеспечения (ПО) начала давать сбои [4]. Речь идет о классической каскадной модели, в основе которой лежит календарное планирование, когда задачи и этапы проекта четко следуют один за другим, и для каждого этапа устанавливаются жесткие сроки. Данная модель перестала отвечать требованиям времени и особенностям реализации проектов в области IT-технологий: изначальные представления о сроках, содержании и продолжительности этапов и операций проектов не соответствовали реальности. «Гладко было на бумаге», но процесс разработки ПО в реальности совершенно не соответствовал плану. Двумерные графики Ганта, связывавшие наглядно сроки с задачами проекта и так любимые проектными менеджерами, приходилось переделывать чуть ли не каждый день. Требовалась более гибкая система управления проектами в области разработки ПО, где наиболее трудно точно рассчитать количество времени, необходимое на решение задач в силу большого количества факторов неопределенности. И как результат труда двух новаторов Д. Сазерленда и К. Швабера (США), в начале 2000-х годов появился Scrum - революционный метод управления проектами.

Agile и её фреймворк Scrum подразумевают разработку программного обеспечения короткими циклами. Каждый этап – это уменьшенный вариант IT-проекта: проводится планирование, анализ требований, проектирование, кодирование, тестирование и документирование. По окончании этапа получается работающая версия IT-системы, а если требуется – пересматриваются дальнейшие приоритеты проекта, и цикл разработки запускается снова. В итоге создается решение, которое на 100 % соответствует ожиданиям. Преимущества Agile как гибкой разработки программного обеспечения:

- Быстрая и постоянная обратная связь команды разработчиков с заказчиком – изменения в требованиях пользователей оперативно учитываются.

- Гибкий график реализации функциональности – при использовании Agile и Scrum процесс разработки легко перестраивается для лучшей реакции на потребности бизнеса.

- Акцент на эргономичность ИТ-системы – требования заказчика к эргономике системы имеют равный приоритет с требованиями к ее функциональности.

- Отсутствие затрат на формализацию процессов и документации – Agile позволяет избежать затрат.

На практике методология Agile может использоваться в нескольких интерпретациях: Scrum, Extreme Programming, Lean Software Development (LSD), Dynamic Systems Development Method (DSDM), Open Unified Process (OpenUP), Agile Project Management (APM), Microsoft Solutions Framework для Agile (MSF).

Сейчас многие компании частично или полностью применяют концепции и инструменты Agile. Применение данных технологий позволит студентам:

- Познакомиться с полной концепцией Agile подхода, разобраться в Agile как способе мышления (mindset) и общем подходе к организации процессов разработки, помимо набора конкретных практик. А также они смогут научиться определять, достаточен ли в их проекте уровень взаимодействия между людьми.

- Систематизировать знания по основным принципам гибкой разработки, используемым терминам, знаниям о возможных подводных камнях и дополнительных требованиях, которые необходимы для эффективного участия в agile-проектах, распределения ролей, поиска и анализа проблем совместно с другими студентами.

- Ознакомиться сразу с несколькими основными подходами современной разработки программного обеспечения: Agile, Scrum, Kanban от начала работы над проектом до выпуска и поддержки работающего продукта.

Между тем, данная работа требует значительных временных затрат преподавателей, что нередко конфликтует с их аудиторной нагрузкой, созданием/актуализацией рабочих программ дисциплин и требованиями, предъявляемыми в отношении публикационной активности. Поэтому на создание курса, учебника или учебно-методического пособия уходит столько времени, что к моменту завершения часть информации уже теряет актуальность.

В основе Agile лежит философия, базирующаяся на 4-х основных ценностях и 12-ти принципах, представленных в Манифесте Agile, опубликованном группой ведущих специалистов-практиков в области программного обеспечения как ответ на растущее беспокойство по поводу увеличения количества неудач в области реализации ИТ-проектов. Эксперты констатировали не возможность их преодоления с помощью традиционных методов управления проектами, базирующихся на больших объемах документации. И провозгласили альтернативный подход, основанный на следующих ценностях:

1. Прямое общение людей важнее инструментов.
2. Продукт, который реально функционирует, важнее детальной спецификации.
3. Прямое общение с клиентами важнее утряски условий контракта.

4. Следовать изначальному плану бессмысленно, необходимо всем быть готовым к изменению хода событий и требований [2].

Поскольку Scrum предполагает командную деятельность, для удобства обзора информации и мониторинга хода работ над проектом используется такой методический инструмент, как Scrum-доска. Она реализуется либо в виде реальной доски, структурированной определенным образом, либо в электронном виде, например, с помощью сервиса Trello [1]. Суть метода заключается в следующем: работа над проектом строится из серии коротких «перебежек» - спринтов. Каждый спринт (определенный отрезок времени, в течение которого команды работали над задачами проекта, поставленными в бэклоге) завершается встречей команды и презентацией готового продукта или его части заказчику. Обычно временные промежутки берутся одинаковые, в 2-4 недели. Но возможны и более короткие спринты. Основное условие – работа команды в тесном взаимодействии. Для этого есть и встречи в конце спринтов, и ежедневные стендапы – короткие встречи команды в начале дня, где в течение 5-15 минут каждый участник команды дает обратную связь по ситуации: что удалось сделать вчера, что планирует делать сегодня, с какими трудностями столкнулся, какая помощь нужна и т.д. Это позволяет Скрам-мастеру держать ситуацию под контролем и гибко реагировать на возникающие непредвиденные ситуации и изменения. В зависимости от активности команд в спринте, Владелец продукта вносит изменения в бэклог, может добавлять новые задачи, которые появляются по ходу работы команд в спринте.

Достоинства гибких подходов к управлению проектами привели к росту популярности Agile-методологии и способствовали распространению методов на различные сферы бизнеса, производства, на сферу образования.

Таблица 1 – Результаты применения

№	Индикаторы эффективности	До внедрения модели	После внедрения модели
1	Доля студентов, успешно (4 или 5 баллов по 5-балльной шкале) сдавших экзамены по профильным дисциплинам	69%	79%
2	Уровень удовлетворенности студентов учебным процессом по 100-балльной шкале, измеряемой через анкетирование в разрезе дисциплин	62%	79%
3	Количество предприятий (потенциальных работодателей) в базе для устройства на практику	7	38
4	Доля студентов, которые в период обучения в университете начинают работать по специальности	12%	32%
5	Количество работодателей в год, обращающихся с просьбой подбора студента для трудоустройства	3	10
6	Доля студентов, привлекаемых для реализации коммерческого проекта, заказчиком которого является предприятие - потенциальный работодатель	19%	27%

В целом все индикаторы демонстрируют три классических ключевых положительных эффекта, получаемых от внедрения agile-технологий: повышение качества, ускорение процессов, снижение затрат. Большинство индикаторов сочетают в себе сразу два или три положительных эффекта. Стоит отметить, что эффект, связанный со снижением затрат, имеет лишь косвенное проявление в приведенных индикаторах. Однако бесспорно положительный экономический эффект присутствует, когда мы видим: что выпускники, обладая компетенциями на должном уровне, не требуют существенных инвестиций от работодателя в процессы адаптации и повышения квалификации; что студенты, привлекаемые для реализации коммерческих проектов, приносят выгоду кафедре; что для привлечения абитуриентов не нужны большие вложения, так как привлекательный бренд кафедры и выпускника является лучшим маркетинговым средством продвижения образовательных услуг на рынке. Таким образом, приведенные статистические данные с положительной динамикой доказывают состоятельность внедренной управленческой инновации и демонстрируют значительный прирост по параметрам качества и скорости подготовки студентов [3].

Список литературы

- [1] *Лукашенко М.А., Телегина Т.В.* Научить студента думать: Scrum как метод продуктивного обучения в учебном заведении. 2019. №1. URL: <https://cyberleninka.ru/article/n/nauchit-studenta-dumat-scrum-kak-metod-produktivnogo-obucheniya-v-uchebnom-zavedenii> (дата обращения: 10.10.2020).
- [2] *Лукашенко М.А., Телегина Т.В.* Управление созданием образовательных продуктов с помощью метода Scrum. 2019. №1. URL: <https://cyberleninka.ru/article/n/upravlenie-sozdaniem-obrazovatelnyh-produktov-s-pomoschyu-metoda-scrum> (дата обращения: 11.10.2020).
- [3] *Молодчик Н.А., Нагибина Н.И.* Формирование и внедрение модели практико-ориентированного обучения в университете на основе agile-принципов. 2019. №1. URL: <https://cyberleninka.ru/article/n/formirovanie-i-vnedrenie-modeli-praktiko-orientirovannogo-obucheniya-v-universitete-na-osnove-agile-printsipov> (дата обращения: 12.10.2020).
- [4] *Швабер К., Сазерленд Д.* Руководство по Скраму. 2017. №1. URL: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Russian.pdf> (дата обращения: 12.10.2020).