

# ГИБКИЕ МЕТОДОЛОГИИ КАК МЕТОД ПРАКТИКО-ОРИЕНТИРОВАННОГО ПОДХОДА ПРИ ПОДГОТОВКЕ СТУДЕНТОВ ТЕХНИЧЕСКИХ СПЕЦИАЛЬНОСТЕЙ

Н. В. Деева

---

*Гродненский государственный университет имени Янки Купалы  
Гродно, Беларусь  
E-mail: ndeeva@grsu.by*

В статье рассматривается предложенная автором методика проведения практических занятий в рамках практико-ориентированного подхода для студентов технических специальностей. Основой этой методики является одна из методологий Scrum семейства гибких методологий Agile. Приводятся основные определения гибких методологий. Описывается эксперимент, который был проведен автором в рамках общинженерной практики для одной подгруппы студентов специальности «Программное обеспечение информационных технологий».

*Ключевые слова:* практико-ориентированный подход, гибкие методологии, Agile, Scrum.

## ВВЕДЕНИЕ

Сегодня серьезными преимуществами ИТ-компаний на рынке являются скорость выпуска продукта, следование современным технологиям разработки ПО, а также возможность создания масштабируемых приложений. Однако выпуск крупных проектов в одиночку представляется маловероятным и скорее даже невозможным. Мировые тенденции в этой области таковы, что большая часть компаний ориентируются на работу команд разработчиков, а программисты-одиночки (так называемые *freelancer*) привлекаются для выполнения только узкоспециализированных или небольших задач. Таким образом, ИТ-рынок диктует спрос не только на технологически подкованных специалистов, но и на кадры, готовые работать в команде, используя новые методологии разработки программного продукта.

В свете современных тенденций перехода на командный формат работы актуально начинать обучать средствам командной разработки еще со студенческой скамьи. Такая практика позволит подготовить выпускника к реальной производственной деятельности, получить психо-эмоциональный опыт работы в команде, а также повысить его шансы на трудоустройство в текущей конкурентной среде.

В качестве методики практико-ориентированного подхода в обучении автором предлагается использовать уже зарекомендовавшие себя в производственной деятельности, гибкие методологии разработки программного обеспечения Agile, адаптированные к учебной деятельности.

## AGILE - МЕТОДОЛОГИИ

Agile – это семейство методологий, направленных на решение задач бизнеса. Каждая из методологий, входящих в список Agile, нацелена на быструю или гибкую реакцию на

изменения в требованиях заказчика. Особенно популярны Agile-методологии в сфере создания программного продукта, так как сфера ИТ наиболее подвержена изменениям, а успех компании зачастую зависит от гибкой реакции на них.

Еще в 50-е гг. XX в. американским специалистом в области управления бизнесом Уильямом Демингом были сформулированы 14 принципов менеджмента. Применяв эти принципы, компания Toyota получила значительный прирост производительности при выпуске автомобилей и сформировала новое направление управления производством, которое в будущем получило название Lean. Сегодня Lean – это одна из методологий, включенных в список Agile. Lean делает упор на организационную эффективность, оптимизацию и сокращение «отходов».

Все методологии, включенные в Agile, разделяют на 2 категории: сосредоточенные на процессе производства и на управлении и выпуске продукции. Возможность работы с использованием Agile регламентируется так называемым Agile Манифестом.

Точкой отсчета активного использования Agile в промышленном программировании можно считать 1985 год, именно тогда была опубликована статья «Спиральная модель разработки программного обеспечения» Барри Бозма.

В 1999 г. была опубликована работа по новой методологии Scrum, которая уделяет повышенное внимание практикам по самоорганизации команд, именно эту методологию предлагается использовать в качестве базовой при обучении студентов командному программированию.

В списке Agile также большой популярностью пользуются XP (Extreme Programming, в переводе с английского – экстремальное программирование), Kanban и другие. Автор полагает, что и эти методологии могут успешно использоваться в практике обучения командному программированию в рамках практико-ориентированного подхода.

Гибкие методологии лучше всего адаптированы к постоянно ускоряющемуся развитию технологий, появлению новых средств разработки и все более переменчивым требованиям заказчика. Применение Agile позволяет максимально быстро начать выполнение проекта, детали же могут быть уточнены в процессе его реализации.

## **МЕТОДОЛОГИЯ SCRUM**

В качестве методологии для проведения эксперимента была выбрана методология Scrum, которая является одной из самых популярных методологий гибкой разработки, позволяет участникам команды выступать единым целым и отменяет персонализацию. Именно этот аспект наиболее важен для ровного вовлечения в процесс обучения всех студентов, участвующих в эксперименте.

Эта методология предполагает разбиение времени работы на проекте на так называемые спринты, которые всегда имеют одинаковый временной промежуток. В одном спринте выполняется определенный командой набор задач. Важным правилом работы по данной методологии является выполнение всех задач в рамках спринта. Однако может показаться, что команда будет заведомо снижать количество задач в спринте и завышать оценки отдельной задачи, таким образом для эффективной работы команды в методологии выделяют следующие роли участников: Scrum Master, Product Owner и Team.

Скрам Мастер (Scrum Master) отвечает за успех Scrum в проекте и, по сути, является интерфейсом между менеджментом и командой. Как правило, эту роль в проекте играет менеджер проекта или лидер команды (teamleader). Важно подчеркнуть, что Скрам Мастер не раздает задачи членам команды. Команда является самоорганизующейся и самоуправляемой. Скрам Мастер ведет Daily Scrum Meeting (регулярные совещания) и отслеживает прогресс команды при помощи Sprint Backlog (Резерв проекта), отмечает статус всех задач в спринте, а также может помогать Product Owner создавать Backlog для команды.

Основными обязанностями Скрам Мастера являются: создание атмосферы доверия, участие в собраниях (митингах) в качестве фасилитатора, устранение препятствий, обнаружение и демонстрация проблем, контроль за соблюдением практик и процесса в команде.

Product Owner – это человек, отвечающий за разработку продукта и ставящий задачи команде, при этом он не вправе ставить задачи конкретному члену проектной команды в течение спринта. Являясь единой точкой принятия окончательных решений для команды в проекте, Product Owner – это всегда один человек, а не группа. Именно он взаимодействует с командой и заказчиком, управляет ожиданиями заказчиков, предоставляет понятные требования команде, приоритизирует подзадачи, а также отвечает за приемку кода в конце каждой итерации.

В методологии Scrum команда является самоорганизующейся и самоуправляемой. Она берет на себя обязательства по выполнению объема работ на спринт. Как правило, команда состоит из 7 человек (плюс, минус 2). Члены команды должны уметь эффективно взаимодействовать друг с другом. Команда в Scrum кроссфункциональна, в которой нет заранее определенных ролей. Члены команды должны обладать различными навыками, так чтобы любое звено, в случае необходимости, могло быть безболезненно заменено. Вклад же отдельных членов команды не оценивается, чтобы не пошатнуть принцип самоорганизации. Работа каждого оценивается как работа единой группы. Желательно, чтобы команда работала в одной комнате (возможно, виртуальной), а члены команды всегда должны видеть текущее положение дел, которое отображается на специальной доске White board. Оценку задач выполняют с помощью специальных карт Planning Poker, которые помогают исключить влияние авторитетов и максимально вовлечь в процесс всех участников команды.

### **ИСПОЛЬЗОВАНИЕ МЕТОДОЛОГИИ SCRUM В УЧЕБНОМ ПРОЦЕССЕ**

Эксперимент по внедрению методологии Scrum в учебный процесс проходил в мае 2012 г. в рамках общеинженерной практики для одной подгруппы студентов специальности «Программное обеспечение информационных технологий» на факультете математики и информатики Гродненского государственного университета как итоговое занятие для закрепления навыков и знаний основ алгоритмизации и языка программирования C++. Временной диапазон – 8 академических часов.

Подгруппа была разделена на 2 команды по 7 человек, в качестве Скрам Мастера выступали студенты специальности «Управление информационными ресурсами». Автор статьи выступала в роли Product Owner, формировала задания для команд, требования, а также отвечала на все возникающие по заданиям вопросы.

Команды самостоятельно обсудили задание, определили список подзадач, а также технологии для их решения, сформировали спринт и оценили каждую из подзадач с помощью Planning Poker. Всего за время эксперимента каждой командой было выполнено по два спринта.

Так как команда самостоятельно выполняла дифференциацию задания и оценки подзадач, то после завершения первого спринта, как и ожидалось, были выполнены не все поставленные задачи, а также выявлено, что оценки не всегда были верными. Скрам Мастер поднял эти вопросы в конце спринта, а также прокомментировал свои наблюдения по работе команды во время спринта, выявив слабые места и возможные ошибки. В результате финального обсуждения выработались методы оценки заданий, а также были перераспределены подзадачи и определены их приоритеты.

В целом эксперимент показал заинтересованность студентов в таком виде учебной деятельности, в ходе работы команды были выявлены повышение мотивации к формированию знаний в области программирования, а также коммуникационных качеств, что бесспорно важно в командной работе.

Так как в качестве Скрам Мастеров выступали также студенты, они получили опыт управления командой в приближенном к реальным условиям. Основными метриками их работы для оценки команды были: соответствие оценок задач спринта реальному времени выполнения, эффективность формирования спринта (нет ли простоев или завалов), эмоциональный фон в команде (наличие споров, разногласий), эффективность проведения митингов-встреч (определение уровня компетенций команды в разности оценок planning poker).

### **ЗАКЛЮЧЕНИЕ**

Описанный эксперимент показал, что гибкие методологии разработки программного обеспечения могут быть адаптированы и применяться в учебном процессе для студентов разных специальностей ИТ-направления. Внедрение новых методик не только обусловлено интересом студентов и повышением их мотивации, но и приобретением ими опыта работы в команде, столь актуальной сегодня. Автору представляется, что в случае организации практических (лабораторных) занятий в полном объеме часов или частично на разных курсах специальности возможно даже выйти на выполнение объемных проектов студентами и формирование у них хорошего портфолио для будущего успешного трудоустройства. Предложенный метод, бесспорно, может быть применен только в рамках практико-ориентированного подхода к обучению студентов.