*УДК 37.013:001.895:374-057.875*
***В. Н. Леванцов, Е. В. Рафалова***
*г. Гомель, ГГУ имени Ф. Скорины*

# ИНТЕГРАЦИЯ СОВРЕМЕННЫХ МЕТОДОЛОГИЙ УПРАВЛЕНИЯ ПРОЕКТАМИ В ПОДГОТОВКУ ИТ-СПЕЦИАЛИСТОВ

Современная разработка программного обеспечения (ПО) представляет собой сложный итеративный процесс, успех которого напрямую зависит от выбора и корректного применения методологии управления проектами. В условиях быстро меняющихся технологий и требований рынка труда, подготовка конкурентоспособного ИТ-специалиста требует не только глубоких технических знаний, но и владения инструментами планирования, командного взаимодействия и гибкого реагирования на изменения. Таким образом, включение в образовательные программы дисциплин, посвященных проектным методологиям, становится неотъемлемой частью формирования профессиональных компетенций будущих разработчиков, менеджеров и аналитиков.

Изучение методологий в образовательном контексте преследует такие стратегические цели, как формирование системного мышления, развитие «гибких» навыков (soft skills), что в перспективе поможет повысить навыки коммуникации в команде, приобретение практического опыта работы в процессах, приближенных к современным стандартам, и осознанный выбор инструментария для разработки ПО.

В текущее время широкое применение получили следующие модели:

1. Waterfall (Каскадная модель).
2. Agile (Гибкая разработка).
3. Scrum.
4. Kanban.

Исторически первой и самой строгой методологией является Waterfall. Она пришла из индустриального производства и строительства. Суть метода заключается в последовательном прохождении этапов. Каждый следующий этап начинается только после полного завершения предыдущего. Возврат назад не предусмотрен (или очень дорог), как вода в водопаде не течет вверх.

Основные этапы модели включают сбор и анализ требований, проектирование, разработка, тестирование, внедрение и поддержка.

Преимущества: четкость, предсказуемость сроков и бюджета (при условии грамотно проведённых этапов сбора и анализа требований), понятная документация. Модель подходит для проектов с жесткими, неизменными требованиями (например, ПО для медицинского оборудования или космической отрасли).

Недостатки: низкая гибкость. Если на этапе тестирования выяснится, что архитектура была выбрана неверно, переделывать придется всё с начального этапа. Заказчик получает возможность оценить результат только по завершении всего цикла работ.

Понятие Agile появилось, когда группа разработчиков создала «Agile Manifesto», который предложил новый взгляд на управление проектами. Agile – это не конкретная инструкция, а семейство методологий, объединенных общими ценностями: люди и взаимодействие важнее процессов и инструментов, работающий продукт важнее исчерпывающей документации, сотрудничество с заказчиком важнее согласования условий контракта, готовность к изменениям важнее следования первоначальному плану. На практике Agile реализуется через различные фреймворки, среди которых наибольшее распространение получили Scrum и Kanban.

Scrum – это наиболее популярный фреймворк Agile. Он разбивает процесс разработки на короткие циклы, называемые спринтами, которые обычно длятся 1–4 недели. В рамках каждого спринта команда создает готовый к использованию инкремент продукта.

Процесс регулируется четким набором ролей, артефактов и регулярных мероприятий. Владелец продукта (Product Owner) определяет приоритеты и формирует бэклог продукта. Scrum-мастер отвечает за соблюдение процесса и устранение препятствий. Команда разработки реализует поставленные задачи. К регулярным мероприятиям относятся: ежедневные встречи, чтобы обсудить, что сделано, что будет сделано и какие есть проблемы; демонстрация результатов спринта заказчику и ретроспектива для анализа и улучшения рабочих процессов. Scrum наиболее эффективен для проектов с изменчивыми требованиями, где важна регулярная и инкрементальная поставка ценности заказчику.

Kanban, возникший в производственной системе Toyota, фокусируется на визуализации рабочего процесса и управлении потоком задач. Работа ведется непрерывно, без фиксированных временных итераций. Главная идея – визуализация рабочего процесса и ограничение количества задач, выполняемых одновременно (Work In Progress). Процесс отображается на доске с колонками, соответствующими стадиям выполнения задачи.

В отличие от Scrum, здесь нет жестких спринтов. Работа идет непрерывным потоком. Задача берется из общей очереди, проходит установленные стадии, например, к выполнению, в работе, тестирование, готово, и результат демонстрируется заказчику. Kanban часто используется в технической поддержке или в командах, занимающихся развитием уже готового продукта.

Не существует единственно верной методологии для разработки, выбор зависит от контекста. Каскадная модель сохраняет актуальность для проектов с консервативными, четко определенными требованиями. В условиях неопределенности и высокой динамики изменений приоритет переходит к гибким подходам. Scrum предлагает структурированную циклическую модель для командной работы над созданием нового продукта, в то время как Kanban обеспечивает гибкое управление непрерывным потоком задач. Эффективность методологии определяется ее соответствием специфике проекта, характеру требований и организационной культуре команды.

Таким образом, спектр методологий управления проектами предоставляет будущим специалистам разнообразный инструментарий. Однако для их осмысленного применения недостаточно теоретического ознакомления. Ключевой педагогической задачей становится создание образовательной среды, в которой студенты смогут не только узнать о Waterfall, Scrum или Kanban, но и «прожить» их, столкнувшись с типичными задачами командной разработки.

Эффективное усвоение принципов гибких методологий требует отхода от чисто лекционного формата в сторону активного, практико-ориентированного обучения. Интеграция методологий в учебный процесс должна происходить на трех уровнях: концептуальном (понимание ценностей и принципов), инструментальном (владение инструментами и практиками) и поведенческом (развитие соответствующего мышления и soft skills).

Наиболее эффективным методом является включение в учебный план сквозного группового проекта или серии мини-проектов, которые реализуются с применением изучаемых методологий. Например, начальный этап – формализация требований к программному продукту и создание архитектурного макета может быть выполнен по принципам Waterfall, что подчеркивает важность этапности. Затем проект переводится в Agile-режим для итеративной разработки и тестирования функциональности.

Моделирование рабочего процесса через учебные проекты позволит погрузить студентов в условия, близкие к реальным. Например, студенческие команды могут разрабатывать прототип веб-приложения или мобильного сервиса. Ключевой задачей преподавателя становится трансформация из лектора в роль Product Owner и Scrum Master. Он формирует и приоритизирует учебный бэклог продукта, содержащий функциональные требования, а также помогает команде организовать свою работу и устраняет организационные препятствия.

Для визуализации и управления учебным проектом целесообразно внедрить цифровые аналоги профессиональных инструментов. Создание учебного бэклога продукта

происходит в виде структурированного списка в Google Sheets или Trello. Далее задачи распределяются по спринтам длительностью 2-3 учебные недели. Студенты проводят краткие ежедневные встречи в начале лабораторных занятий, фиксируют прогресс на Scrum-доске (например, в Miro) и по итогам спринта представляют работающий инкремент на демонстрации в рамках практических занятий.

Организуя работу над менее структурированными задачами, такими как выполнение этапа разработки проекта, подготовка документации или исправление ошибок, студенты создают общую Kanban-доску с колонками «Бэклог», «В работу», «В процессе», «Ревью/Тестирование», «Готово». Устанавливается лимит незавершенной работы (WIP Limit), например, не более двух задач на одного участника в колонке «В процессе». Это наглядно учит управлять потоком, выявлять «узкие места» и бороться с многозадачностью.

Для формирования целостной картины развития методологий от Waterfall к Agile, рекомендуется поступательное погружение. В начале курса студенты могут выполнить небольшой проект по водопадной модели, тщательно проработав техническое задание, архитектурный проект и план тестирования. Это дает понимание важности этапности и документирования процессов. Затем осуществляется переход к гибким методологиям. Такой опыт позволяет студентам на практике оценить достоинства и ограничения каждой модели и осознанно выбирать подходы в будущей профессиональной деятельности.

Ключевым образовательным элементом, завершающим каждый учебный спринт или проектную фазу, должна стать ретроспективная встреча. Её цель не просто поставить оценку, а проанализировать процесс – что из запланированного удалось реализовать, какие возникли трудности в коммуникации или технической реализации, какие методы работы стоит сохранить, а какие изменить.

Таким образом, интеграция методологий управления проектами в обучение превращает теоретический курс в динамичную учебную среду, максимально приближенную к реальности ИТ-индустрии. Студенты не только пассивно изучают технологии, но и активно используют их, приобретая незаменимый опыт самоорганизации, командного взаимодействия и итеративной разработки, что напрямую повышает их конкурентоспособность на рынке труда.

## Литература

1. Бек, К. Экстремальное программирование: разработка через тестирование / Кент Бек; [пер. с англ. С. А. Слинкина, М. В. Иванова]. – Санкт-Петербург: Питер, 2018. – 219 с.

2. Сазерленд, Д. Scrum: революционный метод управления проектами / Джефф Сазерленд; [пер. с англ. П. Миронова]. – Москва : Манн, Иванов и Фербер, 2016. – 267 с.