

УДК 378.147:62

ИСПОЛЬЗОВАНИЕ SCRUM МЕТОДОЛОГИИ ПРИ ПОДГОТОВКЕ ИНЖЕНЕРОВ-ПРОГРАММИСТОВ



Чорна Алена Витальевна
ассистент кафедры информатики и кибернетики
Мелитопольского государственного педагогического университета
имени Богдана Хмельницкого,
Мелитополь, Украина

USE OF SCRUM METHODOLOGY AT PREPARATION OF ENGINEERING PROGRAMMERS

Chorna Alyona Vitalievna
assistant of the Department of Informatics and Cybernetics
Melitopol State Pedagogical University
name of Bogdan Khmelnytsky,
Melitopol, Ukraine

АННОТАЦИЯ

В статье освещена общая концепция та принципы гибкой методологии разработки программного обеспечения, как платформы при подготовке будущих специалистов. Обоснованные преимущества и недостатки указанной методологии. Приведены основные определения гибких методологий. Проведен сравнительный анализ гибких методологий: экстремальное программирование, Scrum, Kanban, «бережливая разработка», «разработка управляемая функциональностью». Выбранная методология Scrum семейства гибких методологий Agile, как одна из наиболее эффективных методологий при подготовке будущих инженеро-специалистов в процессе изучения профессиональных дисциплин.

Ключевые слова: гибкие методологии, инженер-программист, цикл профессиональных дисциплин, Scrum методология.

ANNOTATION

The article describes the general concept of the principles of a flexible software development methodology, as a platform for the training of future specialists. Substantiated advantages and disadvantages of this methodology. The main definitions of flexible methodologies are given. A comparative analysis of flexible methodologies is conducted: extreme programming, Scrum, Kanban, "lean development", "development driven by functionality". The chosen methodology of the Scrum family of agile flexible methodologies is one of the most effective methodologies for the training of future engineers in the process of studying professional disciplines.

Keywords: flexible methodology, software engineer, a cycle of professional disciplines, Scrum methodology.

Актуальность.

Выпуск продукта, который отвечает современным технологиям разработки программного обеспечения, на сегодня есть одним из показателей успешности ИТ-компании на рынке труда. Но одному разработчику выпустить качественный продукт невозможно. Большинство компаний ориентируется на командную работу разработчиков, привлекая при этом фрилансеров для выполнения небольших задач. Поэтому, на рынке ИТ стали востребованные специалисты не только технологически подкованы, но и так же умеющие работать в команде при этом используя различные методологии программирования.

Командная форма работы на сегодня стала очень востребованной и актуальной, поэтому подготовка будущих инженеро-программистов к командной разработке проектов должна осуществляться у процессе изучения дисциплин профессионального цикла подготовки. Это

позволит выпускать специалистов готовых работать в современной, конкурентной среде.

Постановка задач статьи. Цель статьи – описать возможности Scrum методологии при подготовке будущих инженеро-программистов в процессе изучения профессиональных дисциплин.

Основная часть.

Существующие методологии имеют свои преимущества и недостатки в зависимости от того какая именно задача ставится перед исполнителем работ. Традиционная методология управления проектом является линейной, где все происходит в одном цикле. Все подробно планируется и по результатам выполнения всех этапов жизненного цикла проект сдается полностью. Но при этом возникает проблема оценки, если студент не выполнил какую-то часть в начале, то он не сможет получить необходимые баллы, поэтому целесообразным является использование гибкой методологии разработки.

Гибкая методология разработки базируется на итеративной разработке, то есть работа выполняется серией коротких циклов, состоящих из планирования, реализации, проверки и оценки. В каждой итерации (цикле) группа может проводить самоанализ, осуществлять самостоятельное управление и исправлять ошибки, возникающие при решении задачи [2].

Гибкая методология разработки программного обеспечения базируется на двух компонентах: ценностях и принципах гибкой методологии. Все их положения освещены в двух документах: «Manifesto for Agile Software Development» (Манифест о гибкой разработке программного обеспечения) и «Principles of flexible development» (Принципы гибкой разработки от 11-13 февраля 2001 г.) [1].

Основные принципы Agile следующие:

- разработка ведётся короткими циклами(итерациями) длительностью от одной до четырёх недель;
- в конце каждого цикла заказчик получает определённые ценные результаты;
- команда разработки сотрудничает с заказчиком на протяжении всей разработки;
- изменения проекта приветствуются и включаются в работу максимально быстро [4, с. 227].

К классу гибких методологий разработки относятся следующие методики: экстремальное программирование (XP), Scrum, Kanban, «бережливая разработка» (lean), разработка управляемая функциональностью (fdd) и др. В таблице 1 приведен сравнительный анализ гибких методологий.

Таблица 1.

Сравнительный анализ гибких методологий

Название	Ключевые моменты	Уникальные особенности	Недостатки
Экстремальное программирование (XP)	Работа над проектом выполняется в маленьких командах; каждый день проходят совещания; документация сведена к минимуму; члены команды общаются неформально	В процессе работы над проектом ведётся постоянная корректировка, что способствует росту качества выполнения проекта	Члены команды должны быть очень хорошо самоорганизованы Не подходит для глобального управления проектом
Scrum	Команда с высокой степенью самоорганизации, правильно распределены роли между членами команды (5-7 членов), длина цикла (итерации) составляет 2-4 недели, ведение контроля	Существует много документации по работе с методологией, ведётся высокий уровень совместной работы членов команды	Если происходят задержки выполнения некоторой итерации проекта, то это ведет к невыполнению проекта в указанный срок
«бережливая разработка» (lean)	Проект выполняется при помощи коротких циклов	Методология способствует повышению мотивации команды разработчиков, основной принцип работы исключения потерь (лишняя функциональность продукта, нечеткое ТЗ)	Lean-методологию применяют для выполнения небольших проектов
«разработка, управляемая функциональностью» (fdd)	Работа ведётся при помощи коротких итераций (несколько часов), выполнение проекта происходит в пять этапов	На первом этапе разрабатывается модель, составляются все функции продукта, на последнем этапе происходит тестирование	Основной упор в fdd методологии делается на создании проекта продукта, а не на качестве его реализации
Kanban	Принцип методологии успешно выполнить поставленное задание	Задачи не фиксируются по времени, основная цель методологии выполнить задачу.	Учет времени на разработку составляет среднее арифметическое времени потраченного на проект в целом, не ведётся контроль за разработкой

Особенностью экстремального программирования есть осуществление разработки проекта через его тестирование (для выполнения тестирования разработчику нужно проверить код при помощи выполнения тестов модулю, что был разработан). Главным недостатком есть то что на итерации реализовываются те функции, что нужны в этот момент, то есть отсутствует проектирование будущих функций.

Scrum методология широко известная на сегодняшний день. Перед началом выполнения проекта происходит разбиение на итерации. Команда разработчиков перед выполнением итерации определяет всю функциональность, которая будет реализована. Требования не меняются в течении итерации. В методологии очень развит контроль за выполнение итераций (циклов) [6, с. 5].

Lean-методология эффективная в случае соблюдения основных принципов: проведение рефакторинга, умение сократить время на разработку проекта, сильная мотивация членов команды, решения принимать на основе полного анализа. Методология не эффективная на большие проекты.

FDD-методология строится на пяти видах деятельности: создание целостной модели продукта, составление всех функций продукта, составление подробного плана работы над продуктом, проектирование та разработка функциональности продукта.

Методология Kanban не эффективная для больших команд разработчиков. Эта методология позволяет распределить нагрузку между исполнителями.

На основе анализа гибких методологии при подготовке инженеров-программистов в процессе изучения профессиональных дисциплин была выбрана методология Scrum.

Методология Scrum была впервые представлена в 1990-х годах Кеном Швабером и Джефом Сазерлендом в виде четко задокументированного и формализованного «руководства по «скраму» [5, с. 10].

Представленная методология предусматривает 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 сознательно 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 [3].

Scrum Master (преподаватель) - отвечает за создание рабочей атмосферы, участие в собрании, устранение препятствий, выявление и

демонстрация проблем, контроль за соблюдением практик и процесса в команде.

Во время выполнения задания группе Product Owner выступает студент, который одновременно и является ответственным за конечный продукт (выполненное задание). В его непосредственных обязанностях относят постановка конкретных задач команде, выполняет задание. На при конце итерации он отвечает за правильность выполненного задания.

В методологии Scrum команда является самоорганизованной и самоуправляемой. Она берет на себя обязательства по выполнению объема работ. Как правило, команда состоит из 7 человек (плюс, минус 2). Работа команды оценивается как работа отдельной группы. Члены команды должны обладать различными навыками [7, с. 223].

Дисциплины «Введение в специальность», «Операционные системы и системное программирование» та «Управление программными проектами» относится к циклу профессиональной и практической подготовки будущего инженера-программиста образовательного уровня «бакалавр». Для эффективной организации учебной работы студентов специальности 6.040302 Информатика та 122 Компьютерные науки, для развития их творческого мышления, воображения, умения работать в команде, приобретения профессиональных умений и навыков, изучение дисциплин циклу профессиональной и практической подготовки осуществляется при помощи Scrum методологии.

В начале изучения дисциплины группа студентов разбивается на команды, которой необходимо освоить все темы предмета, приобрести определенные компетентности. Внутри группы команды студентов сами распределяют роли. Вся информация о команде и о заданиях для изучения вносится на Scrum-доску. Она в свою очередь состоит из столбцов: «Все задачи», «Необходимо выполнить», «В работе», «Сделано». Так же на доску вносятся еще четыре блока: «Характеристики выполненного»; «Диаграмма Ганта» - диаграмма, которая показывает график исполнения заданий; «Результаты» - эта графа содержит наработанный объем баллов, полученных на занятиях.

У каждого члена группы есть перечень задач для выполнения. Задачи разбиты на спринты, по истечению которого проводится проверка выполненного задания. Если же задача не выполнена, то она из раздела «Сделано» переходит в раздел «Необходимо выполнить». В блоке «Диаграмма Ганта» отображается перечень полученных компетентностей в результате изучения материала. Контроль за качеством изучения дисциплины теперь проводится не в конце семестре, а на протяжении всего времени изучения дисциплины.

До преимуществ использования Scrum методологии следует отнести

1. Формирование у инженеров-программистов управленческих умений: ведения
2. принятия и реализации управленческих решений с целью оптимизации деятельности, умение управления временем, делегирование полномочий, работа в группах при разработке сложных программ
3. Повышение личностного профессионализма.
4. Возможность выполнения реальных задач по автоматизации деятельности образовательной процесса.

Выводы и перспективы дальнейших исследований.

Использование методологии Scrum будущими инженерами-программистами в процессе изучения профессиональных дисциплин уменьшит разрыв между уровнем академической подготовки и практическими требованиями к IT – специалистам.

Для эффективной организации гибкой методологии Scrum лучше использовать вспомогательные компьютерные средства управления процессом разработки программного обеспечения. К основным функциям указанных средств можно отнести: способность к планированию задач, возможность составления расписания, ведение контроля за выполнением задания, правильное распределение ресурсов, возможность совместной работы (делать команды, группы), быстрое общение, управление системой, ее документацией. В дальнейших исследованиях

процесса разработки программного продукта, планируется провести анализ компьютерных средств обеспечения.

СПИСОК ЛИТЕРАТУРЫ:

1. Manifesto for Agile Software Development [Electronic resource]. – <http://agilemanifesto.org/>.
2. Вольфсон Б. Гибкие методологии разработки [Электронный ресурс]. – Режим доступа: <http://admlib.ru/books/10/Gibkie-metodologii.pdf>
3. Деева Н.В. Гибкие методологии как метод практико-ориентированного подхода при подготовке студентов технических специальностей / Н.В. Деева// [Электронный ресурс]. – Режим доступа: http://www.elib.bsu.by/bitstream/123456789/22180/1/%D0%94%D0%B5%D0%B2%D0%B0%20%D0%9D_%D0%92.pdf
4. Карпов Д.В. Гибкая методология разработки программного обеспечения / Д.В. Карпов // Информационные технологии. Вестник Нижегородского университета им. Н.И. Лобачевского, 2011, No 3(2), с. 227–230.
5. Сазерленд Д. Scrum. Революционный метод управления проектами. – Москва: МИФ. – 2015. – 288 с.
6. Тронин В. Г. Возможности применения гибких методологий управления проектами при обучении в вузе по техническим специальностям / В. Г. Тронин // Вестник Ульяновского государственного технического университета (Вестник УлГТУ). – 2016. – С. 4-6.
7. Чорна А.В. Можливості використання комп'ютерних засобів управління процесом розробки програмного забезпечення під час вивчення дисципліни «Операційні системи та системне програмування» / А.В. Чорна // Проблеми інженерно-педагогічної освіти. Збірник наукових праць. – Харків, Українська інженерно-педагогічна академія (УІПА), 2015. – Випуск 48. – С. 221-227.