

SCRUM МЕТОДОЛОГІЯ У ПРОФЕСІЙНІЙ ПІДГОТОВЦІ МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ

Чорна А.

Мелітопольський державний педагогічний університет

імені Богдана Хмельницького,

м. Мелітополь

e-mail: alonachorna@gmail.com

Постановка проблеми. Випуск продукту, який відповідає сучасним технологіям розробки програмного забезпечення, на сьогодні є одним з показників успішності ІТ-компанії на ринку праці. Але одному розробнику випустити якісний продукт неможливо. Більшість компаній орієнтується на командну роботу розробників, залучаючи при цьому фрілансерів для виконання невеликих завдань. Тому, на ринку ІТ стали затребувані фахівці не тільки технологічно підковані, але і ті, що вміють працювати в команді, при цьому використовуючи різні методології програмування.

Командна форма роботи на сьогодні стала дуже затребуваною і актуальною, тому підготовка майбутніх інженерів-програмістів до командної розробки проєктів повинна здійснюватися у процесі вивчення дисциплін професійного циклу підготовки. Це дозволить здійснювати підготовку фахівців здатних працювати в сучасному конкурентному середовищі.

Мета статті розкрити можливості Scrum методології у практичній підготовці майбутніх інженерів-програмістів у процесі вивчення фахових дисциплін.

Виклад основного матеріалу. Традиційна методологія управління проєктом є лінійною, де все відбувається в одному циклі. Все детально планується і за результатом виконання всіх етапів життєвого циклу проєкт здається повністю. Але при цьому постає проблема оцінювання, якщо студент не виконав якусь частину на початку, то він не зможе отримати необхідні бали, саме тому доцільним є використання гнучкої методології розробки. Гнучка методологія розробки базується на ітеративній розробці, тобто робота виконується серією коротких циклів, що складаються з планування, реалізації, перевірки і оцінки. У кожній ітерації група може проводити самоаналіз, здійснювати самостійне управління і виправляти помилки, що виникають при вирішенні задачі [2]. Гнучка методологія – це ітеративний процес, який використовує унікальні практики для отримання нового функціоналу програмного забезпечення кожні 1-4 тижні.

Цінності і принципи гнучкої методології закріплені в двох документах: «Маніфест гнучких методологій розробки» (Manifesto for Agile Software

Development) і «Принципи гнучкої розробки» (11-13 лютого 2001 р. США) [1].

До класу гнучких методологій розробки відносять такі методики: екстремальне програмування (XP), Scrum, Kanban, «ощадлива розробка», розробка керована функціональністю (fdd) і ін. В таблиці 1 наведено порівняльний аналіз гнучких методологій.

Таблиця 1.

Порівняльний аналіз гнучких методологій

Назва	Ключові моменти	Унікальні особливості	Недоліки
Екстремальне програмування (XP)	Маленькі команди, щоденні наради, неформальний тип комунікації всередині команди, документації	Постійне коригування проекту для підвищення його ефективності і для адаптації до змін	Більше підходить для індивідуальних практик, ніж для глобального управління, так як в останньому випадку є ризик формування недисциплінованих команд
Scrum	Незалежна, невелика, самоорганізована команда розробників, довжина ітерації - 2-4 тижні, команда сама вирішує скільки часу їй потрібно для виконання завдання	Високий рівень комунікації і взаємодії всередині команди, чітко прописана формальна організація цієї методології	Є ризик збільшення часу проекту за рахунок витрат «Scrum»-заходів
«ощадлива розробка» (lean)	Використання візуалізуючих інструментів, розробка через тестування, короткі ітерації	Головними є ті функції ПЗ, які цінні для замовника, має місце постійне мотивування команди	Рішення приймаються довго, підходить тільки для маленьких проектів
«розробка, керована функціональністю» (fdd)	П'яти кроковий процес, об'єктно-орієнтована розробка, дуже короткі ітерації (можуть досягати за тривалістю кілька годин)	Простота методу, об'єктне моделювання	Дана методологія фокусує свою увагу виключно на проектуванні та впровадженні, дуже мало уваги приділено саме розробці
Kanban	Самоорганізована команда, весь упор на завдання	Немає обмежень за часом виконання, зате є обмеження на число «роботи в даний момент», ефективно, коли невідомо, що може в подальшому знадобитися замовнику від ПЗ	Недолік методології затяжний характер

Методологія Scrum націлена на взаємодію із замовником, і, не дивлячись на те, що команда розробників сама вирішує, які завдання вона буде виконувати протягом однієї ітерації, в зазначеній методології присутній спостерігач (Scrum-майстер), який контролює дотримання Scrum-процесу [4, с. 10].

Scrum методологія передбачає розбиття часу роботи на проекти (спринти), які завжди мають однаковий часовий проміжок. В одному спринті виконується певний командою набір завдань. Важливим правилом роботи по даній методології є виконання всіх завдань у рамках спринту. Для ефективної

роботи команди в методології виділяють наступні ролі учасників: Scrum Master, яку в умовах освітнього процесу здійснює викладач, Product Owner і Team, які можуть виконувати студенти [3].

Scrum Master (викладач) – відповідає за створення робочої атмосфери, участь в зборах, усунення перешкод, виявлення і демонстрація проблем, контроль за дотриманням практик і процесу в команді.

Під час виконання завдання групі Product Owner виступає студент, який одночасно і є відповідальним за кінцевий продукт (виконане завдання). У його безпосередніх обов'язків відносять постановка конкретних завдань команді, виконує завдання. На при кінці ітерації він відповідає за правильність виконаного завдання.

У методології Scrum команда є самоорганізованою і самокерованою. Вона бере на себе зобов'язання щодо виконання обсягу робіт. Як правило, команда складається з 7 осіб (плюс, мінус 2). Робота команди оцінюється як робота окремої групи. Члени команди повинні володіти різними навичками [5, с. 5].

На нашу думку, доцільно у процесі викладання дисциплін «Вступ до спеціальності», «Операційні системи та системне програмування» та «Управління програмними проектами», що відноситься до циклу професійної підготовки інженерів-програмістів освітньо-кваліфікаційного рівня «бакалавр», застосовувати цю методологію для ефективної організації навчальної роботи студентів спеціальності 6.040302 Інформатика та 122 Комп'ютерні науки, для розвитку їх творчого мислення, уяви, вміння працювати в команді, набуття професійних умінь і навичок, вивчення дисциплін циклу професійної підготовки здійснюється за допомогою Scrum методології.

На початку вивчення дисципліни група студентів розбивається на команди, якій необхідно освоїти всі теми, набувши в ході вивчення матеріалу певних компетентностей. Усередині групи члени команди самі розподіляють ролі. Вся інформація про команду і про завдання для вивчення вноситься на Scrum-дошку. Вона в свою чергу складається із стовпців: «Усі завдання», «Необхідно виконати», «В роботі», «Зроблено». Так само на дошку вносяться ще чотири блоки: «Характеристики виконаного»; «Діаграма Ганта» - діаграма, яка показує графік виконання завдань; «Результати» - ця графа містить напрацьований обсяг балів, отриманих на заняттях.

У кожного члена групи є перелік завдань для виконання. Завдання розбиті на спринти, по закінченню якого проводиться перевірка виконаного завдання. Якщо ж завдання не виконане, то вона з розділу «Зроблено» переходить в розділ «Необхідно виконати». У блоці «Діаграма Ганта»

відображається перелік отриманих компетентностей в результаті вивчення матеріалу. Контроль за якістю вивчення дисципліни тепер проводиться не в кінці семестрі, а на протязі всього часу вивчення дисципліни.

До переваг використання Scrum методології у процесі навчання фахових дисциплін майбутніх інженерів-програмістів слід віднести:

1. Підготовку студентів до роботи на реальних проектах, формування навичок роботи в команді з використанням сучасних гнучких методологій проектування і розробки програмного забезпечення інформаційних систем.

2. Підвищення особистісного професіоналізму.

3. Можливість виконання реальних завдань по автоматизації діяльності освітньої процесу.

Висновки. Використання методології Scrum майбутніми інженерами-програмістами в процесі вивчення фахових дисциплін зменшить розрив між рівнем академічної підготовки і практичними вимогами до ІТ-фахівця. Для ефективної організації гнучкої методології Scrum краще використовувати допоміжні комп'ютерні засоби управління процесом розробки програмного забезпечення.

Література

1. Manifesto for Agile Software Development [Electronic resource]. – <http://agilemanifesto.org/>.

2. Вольфсон Б. Гибкие методологии разработки [Электронный ресурс]. – Режим доступа: <http://adm-lib.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%B5%D0%B2%D0%B0%20%D0%9D_%D0%92.pdf

4. Сазерленд Д. Scrum. Революционный метод управления проектами. – Москва: МИФ. – 2015. – 288 с.

5. 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.

Анотація. У статті висвітлено загальна концепція та принципи гнучкої методології розробки програмного забезпечення, як платформи при підготовці майбутніх фахівців. Обґрунтовані переваги і недоліки зазначеної методології. Проведено порівняльний аналіз гнучких методологій: екстремальне програмування, Scrum, Kanban, «ощадлива розробка», «розробка керована функціональністю». Обрана методологія Scrum сімейства

гнучких методологій Agile, як одна з найбільш ефективних методологій при підготовці майбутніх інженерів-фахівців в процесі вивчення фахових дисциплін.

Ключові слова: гнучкі методології, інженер-програміст, цикл фахових дисциплін, Scrum методологія.

УДК 371.1:801.33:004

ЕЛЕКТРОННИЙ ЗОШИТ НА УРОКАХ ОБРАЗОТВОРЧОГО МИСТЕЦТВА В ПОЧАТКОВІЙ ШКОЛІ

*Шмельова Ю., Постильна О.
Мелітопольський державний педагогічний університет
імені Богдана Хмельницького,
м. Мелітополь
e-mail: postylnaya@mail.ru*

Постановка проблеми. В умовах модернізації системи освіти на перший план виходять завдання формування творчої особистості, створення умов для реалізації можливостей дитини в навколишньому світі.

Дуже важливо для самореалізації дитини створити такі умови за допомогою образотворчого мистецтва, з яких би у дитини виникла потреба до творчості. Сучасна система освіти повинна обладнати дитину універсальними засобами, які допоможуть їй розвиватися.

Застосування на уроках з образотворчого мистецтва інформаційних технологій дає можливість для реалізації творчого потенціалу учнів за рахунок можливостей комп'ютерної техніки, сприяє підвищенню рівня самостійної роботи, підвищення інтересу до навчання та ефективності, розвиває дитину всебічно.

Аналіз останніх досліджень і публікацій. Модернізація начального процесу ставить підвищенні вимоги до дієвих носіїв творчих норм, що знайшло відображення у національних документах, серед яких: Закони України «Про освіту», «Про загальну середню освіту», Національна стратегія розвитку освіти в Україні на період до 2021 року, «Про Концепцію Національної програми інформатизації», Концепція національно-патріотичного виховання дітей і молоді тощо [2, 3].

Проблема використання інформаційних технологій на уроках образотворчого мистецтва на сьогодні перебуває в центрі уваги сучасних дослідників. Теоретичні аспекти використання інформаційних технологій на уроках образотворчого мистецтва висвітлено у роботах таких учених, як А. Хуторського, І. Підласого, І. Якиманської, О. Савченко та ін., проблема творчого розвитку особистості учня не є новою. Значний внесок у теоретико-методологічне та психолого-педагогічне забезпечення навчально-виховного