

Ильин Иван Вадимович,

кандидат педагогических наук, доцент, кафедра информационных систем и математических методов в экономике, экономический факультет, Пермский государственный национальный исследовательский университет; 614990, Россия, г. Пермь, ул. Букирева, 15; e-mail: vania_ilin@mail.ru

Кузаев Айдар Файзуллович,

кандидат физико-математических наук, кафедра прикладной информатики, информационных систем и технологий, факультет информатики и экономики, Пермский государственный гуманитарно-педагогический университет, 614990, Россия, г. Пермь, ул. Пушкина, 42; e-mail: kuzaev@pspu.ru

ПРАКТИКА РЕАЛИЗАЦИИ AGILE-МЕТОДОЛОГИЙ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ДИСЦИПЛИНАХ ЦИКЛА «ПРОГРАММИРОВАНИЕ» ПРИ ПОДГОТОВКЕ БАКАЛАВРОВ ИТ-НАПРАВЛЕНИЙ

КЛЮЧЕВЫЕ СЛОВА: разработка программного обеспечения; программное обеспечение; программирование; обучение программированию; учебные дисциплины; информационные технологии; бакалавриат; Agile-методологии; методика Scrum; Kanban-доска; технологии разработки программного обеспечения; системы управления проектами; информатика; методика преподавания информатики; проектная работа; продуктивное обучение

АННОТАЦИЯ. В статье обсуждаются вопросы методики обучения студентов программированию и управление их деятельностью средствами Agile-методологий. Представлено описание проектной деятельности и средств реализации метода Scrum, адаптированного к практике обучения студентов. В данной методике работы выполняются специальными Scrum-командами с тремя ролями: владелец продукта (представитель заказчика продукта), Scrum-мастер (компетентный член команды, курирующий процесс изнутри) и команды разработчиков (студенты группы). Функционал и дизайн продукта с точки зрения запросов пользователей оформляются в «пользовательские истории», на базе которых формулируется «бэклог продукта». Различные виды работ выполняются командами студентов в течение ограниченных по времени рабочих итераций циклов («спринтов»). В результате выполнения заданий студенты получают инкремент продукта как результат работы одного спринта, который оценивается экспертами. В настоящем исследовании, на базе современных веб-технологий, разработано авторское программное обеспечение, реализующее функционал популярных систем управления проектами по методологии Scrum (реализация выполнена средствами фреймворков Laravel, Vue.js и СУБД PostgreSQL). Система имеет базовый функционал, необходимый для небольших компаний, и отвечает современной потребности в импортозамещении. Представлены результаты опытно-экспериментальной работы, которая проводилась в рамках дисциплин «Технология разработки программного обеспечения» и «Объектно-ориентированное программирование». Теоретическая значимость исследования состоит в адаптации и описании методики организации занятий по технологии Scrum и выделении соответствующих критериев результативности обучения. Практическая значимость исследования состоит в авторской разработке специального программного обеспечения для организации занятий по методике Scrum, поддерживающего процесс обучения программированию.

ДЛЯ ЦИТИРОВАНИЯ: Ильин, И. В. Практика реализации Agile-методологий при разработке программного обеспечения в дисциплинах цикла «Программирование» при подготовке бакалавров ИТ-направлений / И. В. Ильин, А. Ф. Кузаев. – Текст : непосредственный // Педагогическое образование в России. – 2022. – № 5. – С. 104-110. – DOI: 10.26170/2079-8717_2022_05_12.

Ilyin Ivan Vadimovich,

Candidate of Pedagogy, Associate Professor, Department of Information Systems and Mathematical Methods in Economics, Perm State National Research University, Perm, Russia

Kuzaev Aidar Faizullovich,

Candidate of Physics and Mathematics, Department of Applied Informatics, Information Systems and Technologies, Perm State Humanitarian Pedagogical University, Faculty of Informatics and Economics, Perm, Russia

THE PRACTICE OF IMPLEMENTING AGILE METHODOLOGIES IN SOFTWARE DEVELOPMENT IN THE DISCIPLINES OF THE CYCLE “PROGRAMMING” IN THE PREPARATION OF BACHELORS IN IT AREAS

KEYWORDS: software development; software; programming; programming training; academic disciplines; Information Technology; undergraduate; Agile methodologies; Scrum methodology; Kanban board; software development technologies; project management systems; Informatics; methodology of teaching informatics; project work; productive learning

ABSTRACT. The article discusses the issues of methodology for teaching students programming and managing their activities using Agile methodologies. A description of project activities and means of implementing the Scrum method, adapted to the practice of teaching students, is presented. The work is carried out by dedicated Scrum teams with three roles: Product Owner (representative of the product customer),

Scrum Master (a competent team member who oversees the process from the inside) and Development Teams (group students). The functionality and design of the product from the point of view of user requests are framed in "user stories", on the basis of which the "product backlog" is formulated. Different types of work are performed during short, time-limited working cycle iterations ("sprints"). As a result of the work, we get a product increment as a result of the work of one sprint, which is evaluated by experts. In this study, on the basis of modern web technologies, author's software was developed that implements the functionality of popular project management systems according to the Scrum methodology (implementation was carried out using the Laravel, Vue.js frameworks and the PostgreSQL DBMS). Provides the basic functionality required for small companies and meets the modern need for import substitution. The results of the experimental work, which was carried out within the framework of the disciplines "Technology of software development" and "Object-oriented programming", are presented. The theoretical significance of the study lies in the adaptation and description of the methodology for organizing classes using Scrum technology and the allocation of appropriate criteria for the effectiveness of training. The practical significance of the research lies in the author's development of special software for organizing classes according to the Scrum methodology that supports the process of teaching programming.

FOR CITATION: Ilyin, I. V., Kuzaev, A. F. (2022). The Practice of Implementing Agile Methodologies in Software Development in the Disciplines of the Cycle "Programming" in the Preparation of Bachelors in IT Areas. In *Pedagogical Education in Russia*. No. 5, pp. 104-110. DOI: 10.26170/2079-8717_2022_05_12.

Сегодня в российской ИТ-отрасли обнаруживает себя дефицит отечественных программных продуктов и высококомпетентных специалистов в области проектирования, разработки и тестирования программных продуктов. Решение данной проблемы связано в первую очередь с совершенствованием подготовки студентов ИТ-специальностей. Качественные знания и умения, приобретаемые студентами в рамках учебных дисциплин ИТ-цикла, где программирование считается одним из самых сложных категорий, являются важным основанием для их последующей успешной деятельности в профессиональной ИТ-сфере.

Возникает вопрос: как должен быть организован процесс обучения программированию будущих ИТ-специалистов, чтобы он был максимально приближен к их будущей работе в ИТ-сфере?

Вопросы формирования компетенций в области программирования в процессе профессиональной подготовки будущих ИТ-специалистов рассматриваются в исследованиях Л. А. Кугель [13], В. В. Плещева, Е. В. Осокина [15], Ф. В. Шкарбана и др. Авторами обсуждаются содержательные и методические аспекты обучения программированию, однако предлагаемые подходы к организации учебного процесса носят преимущественно традиционный характер и в явно недостаточной мере соотнесены с современным состоянием и перспективами развития технологий проектной деятельности ИТ-специалистов, занимающихся разработкой программных продуктов.

В образовательные программы вузов по направлениям подготовки УГП 09 (09.03.03 Прикладная информатика, 09.03.02 Информационные системы и технологии и др.) входит немало дисциплин, связанных с изучением дисциплин цикла программирования и управления разработкой программного обеспечения. Возникает вопрос о методах построения учебных дисциплин,

отвечающим современным требованиям работы в ИТ-отрасли.

Традиционно в команде разработчиков программных продуктов использовались такие модели жизненных циклов [4; 5; 7; 9; 14], как каскадная (водопадная) модель, инкрементная модель, V-образная модель, итеративная модель, спиральная модель и др. Большая их часть при сегодняшнем постоянно меняющемся динамическом рынке себя исчерпали.

На сегодняшний день применяется ряд более современных методологий, которые объединяет понятие Agile (от англ. «гибкий»). Agile-методология – это набор методов и практик для гибкого управления ИТ-проектами (в т. ч. на этапе разработки) с целью повышения скорости создания программных продуктов и минимизации рисков за счет итерационного выполнения, интерактивного взаимодействия членов команды и быстрой реакции на изменения со стороны заказчика [8; 16]. Основные принципы методологии (так называемые «12 принципов Agile Manifesto» [21]) разработала исследовательская группа из 17 авторитетных специалистов этой области.

Цель нашего исследования – разработать методику организации учебных занятий по дисциплинам цикла программирования в соответствии с современными практиками Agile-методологии.

Agile-методология включает множество практик, среди которых выделяют [18]: управление проектами Scrum, метод управления Kanban, экстремальное программирование (XP), бережливая разработка (Lean) и др. В нашем исследовании мы выбрали методику Scrum и элементы Kanban.

Scrum – это инфраструктура для эффективной организации работ и управления ими. Проектные работы выполняются специальными Scrum-командами с тремя ролями: владелец продукта (представитель заказчика продукта), Scrum-мастер (компе-

тентный член команды, курирующий процесс изнутри) и команды разработчиков [22]. В нашем случае под владельцем продукта понимается преподаватель дисциплины цикла программирования, а остальные роли распределяются среди студентов.

Описание требований к продукту (функционал и дизайн) с точки зрения запросов пользователей оформляется, по терминологии Scrum, в *пользовательские истории (user story)*. На базе него формулируется так называемый *бэклог продукта (product backlog)* в виде ранжированного набора пользовательских историй (в процессе работы студенческой команды этот список может обновляться). На практике студентам ставится задача (с последующей декомпозицией для бэклога) проектирования и разработки программного продукта (тематику студенческие группы выбирают самостоятельно).

Различные виды работ (анализ предметной области, проектирование структуры, реализация и тестирование программных модулей) выполняются в течение коротких, ограниченных по времени рабочих итераций циклов, которые, в терминологии Scrum,

называются *спринтами (sprint)* и обычно делятся от нескольких дней до нескольких недель (месяца). Спринт состоит из нескольких этапов: планирование, работа, презентация результата и анализ. Команда студентов оценивает свои силы на один спринт и формирует столько задач (*бэклог спринта*), сколько сможет реально выполнить. Под *бэклогом спринта (Sprint Backlog)* понимается список задач и объем работы, которую нужно сделать за один спринт (рис. 1).

Внутри спринтов организуются *ежедневные скрамы (daily scrum)* – так называют каждодневную встречу команды (на 10–15 минут). Если команда работает удаленно, то отчет предоставляется в общем чате или по видеозвонку. На подобных встречах участники команды обсуждают отчет о проделанной работе с момента последней встречи, список задач, которые участник должен выполнить до следующего собрания, и затруднения, возникшие в ходе работы (например, проблемы с отображением UI-компонент веб-приложения, взаимодействия программных модулей и базы данных и др.).

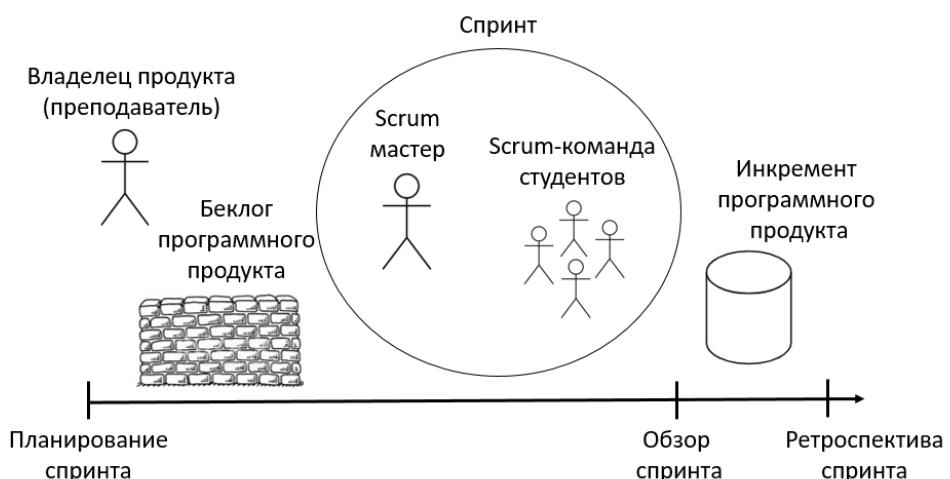


Рис. 1. Модель реализации Scrum-методики при разработке программного обеспечения в дисциплинах цикла «Программирование»

В результате выполнения одной итерации создается *инкремент продукта (product increment)* как результат работы одного спринта. После завершения спринта реализуется так называемая *ретроспектива спринта (sprint retrospective)*, на котором команда, «скрам-мастер» и «владелец продукта» оценивают текущее состояние программного продукта и дают рекомендации по дальнейшей работе.

Одновременно с выполнением основных задач студенты готовят свои вопросы преподавателю по теоретическому блоку программы курса (видеозаписи лекций преподавателя находятся в открытом доступе в дистанционной оболочке обучения вуза).

Вне зависимости от выбора конкретной реализации Agile-методологии применяются программные инструменты, автоматизирующие командную работу над проектом – *системы управления проектами (СУП)* [12; 17] (например, Jira, Trello, Мегаплан и др.). Внутри них для визуального отображения рабочего процесса проекта используется *Scrum-доска*. Это современное интерактивное веб-приложение (рис. 2), как правило, со столбцами: «to do», «in progress», «code review» и «done». По мере работы над проектом на доске перемещаются задачи в другие столбцы.

В настоящем исследовании разработана авторская СУП (рабочее название «Yeskela»

под авторством И. В. Ильина и В. Л. Наго-вицина), которая предоставляет базовый функционал, необходимый для небольших ИТ-компаний, отвечая потребности в импортозамещении в современных реалиях санкций. В ее функционал входят отсутствие количественных ограничений на участников или задач, учетные записи сотрудников, ролевая модель сотрудников, Kanban-доска (как расширенный вариант scrum-доски), механизм управления задачами и установка их приоритета, подключение внешних источников данных (текстовые файлы, электронные таблицы, некото-

рые базы данных), комментирование задач, создание структур подразделений и проектов, формирование отчетной статистики. Программная реализация СУП выполнена средствами фреймворков Laravel, Vue.js и СУБД PostgreSQL.

Используя общую интерактивную доску (рис. 2), студенты видят текущий статус реализации общего проекта [6]. На нее можно свободно добавлять информацию, прикреплять файлы, писать комментарии и др. Внутри описываются задачи: указываются проект, автор, исполнители, приоритет, текущий статус и др.

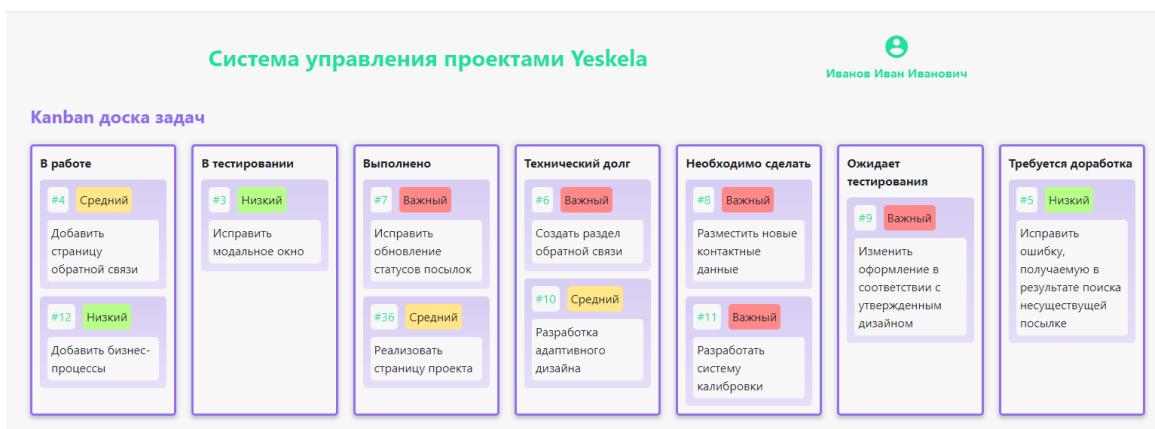


Рис. 2. «Kanban-доска» разработанной СУП

Безусловно, применение подобных программных продуктов в практике работы ИТ-специалистов всех категорий определяет необходимость целенаправленного их внедрения в процесс обучения, что и реализовывалось в нашем исследовании.

Применяя методику Scrum в организации учебных занятий, рационально использовать элементы традиционных педагогических технологий проектного (М. Кнолл) и продуктивного обучения (И. Бем, Й. Шнайдер [2]). Важно отметить, что продуктивная и проектная технологии обучения содержательно пересекаются. Главной отличительной особенностью продуктивного обучения является создание продукта с потребительскими свойствами, допускающего прямое использование и тиражирование. Программные продукты, создаваемые студентами (web-сайты, desktop-программы, мобильные приложения) в рамках предложенной методики, в большей степени соответствуют потребительскому рынку.

Опытно-экспериментальная работа проводилась в рамках дисциплин «Технология разработки программного обеспечения» и «Объектно-ориентированное про-

граммирование». Определяющим критерием результативности обучения в нашем исследовании является достигнутый студентами в результате обучения уровень профессиональной компетентности (ПК) в области программирования (конкретное название компетенции по конкретному учебному плану не так важно). Анализ работ по продуктивному обучению [1; 2; 10] и опыта проведения подобных занятий по методике Scrum показал, что уровень компетентности студента проявляется: а) в качестве освоения теоретической части курса (оцениваются глубина понимания учебного материала и причинно-следственные связи); б) в степени готовности отчетов на Scrum-собраниях (научность и доступность изложения материала); в) в качестве выполнения задач из «бэклога» проекта (объем и качество разработки программного кода модулей); г) в степени своевременности и правильности внесения изменений на Kanban-доску; д) в степени готовности итогового программного продукта (суммарная оценка разработанных студентами подсистем и модулей).

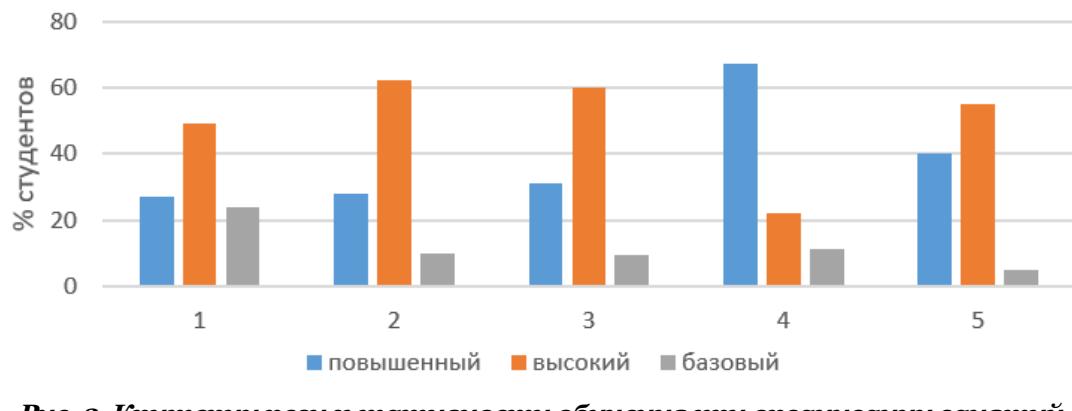


Рис. 3. Критерии результативности обучения при организации занятий по методике Scrum: 1 – качество освоения теоретической части курса; 2 – степень готовности отчетов на Scrum-собраниях; 3 – качество выполнения задач из «бэклога» проекта; 4 – степень своевременности и правильности внесения изменений на Канбан-доску; 5 – степень готовности итогового программного продукта

Подавляющая часть студентов демонстрирует высокий и повышенный уровень сформированности компетенций в решении поставленных профессиональных задач. Для экспертной оценки в балльно-рейтинговой системе использовались требования европейской системы зачетных единиц ECTS [11] (уровень усвоения в 60% использовался для оценки промежуточных этапов работ) и теория В. П. Беспалько [3] относительно степени усвоения учебного материала, характеризующей завершенность обучения (значение в 70% использовалось для оценки итогового проекта).

Теоретическая значимость исследования состоит в адаптации и описании методики организации занятий по технологии Scrum и выделении соответствующих критериев результативности обучения. Практическая значимость исследования состоит в авторской разработке специального программного обеспечения для организации занятий по методике Scrum, поддерживаю-

щего процесс обучения программированию.

Разработка коллективного проекта (в т. ч. реализуются элементы коллективного способа обучения), следствием которого является конечный программный продукт для конкретного потребителя, вызывает у студентов большую заинтересованность и способствует росту их ответственности за качество полученного программного продукта [19; 20].

Итак, предложенная модель обучения будущих ИТ-специалистов по современной методологии управления проектами Scrum, адаптированной под работу на учебных заданиях, обеспечивает нужный уровень мотивации и результативности студентов в разработке коллективного программного продукта. Разработанная в рамках настоящего исследования авторская СУП может использоваться не только для процесса обучения, но и в небольших ИТ-компаниях, отвечаая потребности в импортозамещении в современных реалиях санкций.

ЛИТЕРАТУРА

1. Башмаков, М. И. Что такое продуктивное обучение? / М. И. Башмаков // Теория и практика продуктивного обучения : коллективная монография. – М. : Народное образование, 2000. – С. 6-14.
2. Бем, И. Продуктивное обучение: слагаемые системы / И. Бем, И. Шнайдер // Совместный выпуск журналов «Школьные технологии». – 1999. – № 4; «Новые ценности образования». – 1999. – № 9. – С. 59-70.
3. Беспалько, В. П. Слагаемые педагогической технологии / В. П. Беспалько. – М. : Педагогика, 1989. – 192 с.
4. Битти, Д. Разработка требований к программному обеспечению : пер. с англ. / Д. Битти, К. Вигерс. – М. : Издательство «Русская редакция» ; СПб. : БХВ-Петербург, 2014. – 736 с.
5. Брауде, Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004. – 655 с.
6. Васильева, Е. Н. Интерактивное обучение как модель продуктивного учебного процесса в условиях введения ФГОС / Е. Н. Васильева // Актуальные проблемы качества математической подготовки школьников и студентов: методологический, теоретический и технологический аспекты : матер. IV всерос. науч.-метод. конф. междунар. науч.-образ. форума «Человек, семья, общество: история и перспективы развития» / КГПУ им. В. П. Астафьева. – Красноярск, 2016. – С. 70-75.
7. Винтерс, Т. Делай как в Google. Разработка программного обеспечения / Т. Винтерс, Т. Маншрек, Х. Райт. – СПб. : Питер, 2021. – 544 с.
8. Докукина, И. А. Цифровые платформы в публичном управлении на основе Agile менеджмента / И. А. Докукина, А. В. Полянин // Вестник Воронежского государственного университета. Серия: Экономика и управление. – 2020. – № 1. – С. 126-131.

9. Дэвис, Дж. Философия DevOps. Искусство управления ИТ / Дж. Дэвис, К. Дэниелс. – СПб. : Питер, 2017. – 416 с.
10. Калмыкова, З. И. Продуктивное мышление как основа обучаемости / З. И. Калмыкова. – М. : Педагогика, 1981. – 200 с.
11. Кэрран, Т. Общеевропейские шкалы оценок: опыт национальных систем и ECTS / Т. Кэрран // Высшее образование в Европе. – 2005. – Т. XXIX, № 1. – С. 48.
12. Ким, Д. Руководство по DevOps. Как добиться гибкости, надежности и безопасности мирового уровня в технологических компаниях / Дж. Ким, П. Дебуа, Дж. Уиллес, Дж. Хамбл ; пер. с англ. И. Лейко и И. Васильева ; науч. ред. Н. Корытко. – М. : Манн, Иванов и Фербер, 2018. – 512 с.
13. Кутель, Л. А. Обучение студентов алгоритмизации и программированию на основе структурно-алгоритмического подхода к постановке и реализации задач: на примере направления подготовки бакалавров «прикладная информатика» : автореф. дис. ... канд. пед. наук : 13.00.02 / Кутель Л. А. – Москва, 2015. – 22 с.
14. Мартин, Р. Чистая архитектура. Искусство разработки программного обеспечения / Р. Мартин. – СПб. : Питер, 2018. – 352 с.
15. Осокина, Е. В. Использование метода коллективного проектирования при обучении будущих специалистов в области информационных технологий разработке информационных систем : дис. ... канд. пед. наук : 13.00.02 / Осокина Е. В. – Шадринск, 2011. – 172 с.
16. Рубин, С. Основы Scrum: практическое руководство по гибкой разработке ПО : пер. с англ. / С. Рубин. – СПб. : ООО «Диалектика», 2020. – 544 с.
17. Херинг, М. DevOps для современного предприятия / М. Херинг ; пер. с англ. М. А. Райтмана. – М. : ДМК Пресс, 2020. – 232 с.
18. Atlassian official site. – URL: <https://www.atlassian.com/ru/agile/kanban> (mode of access: 10.09.2022). – Text : electronic.
19. Georgiou, H. Characterising Communication of Scientific Concepts in Student-Generated Digital Products / H. Georgiou // Educ. Sci. – 2020. – No. 10. – P. 18.
20. Knoll, M. The project method: Its vocational education origin and international development / M. Knoll // Journal of Industrial Teacher Education. – 1997. – No. 34 (3). – P. 59-80.
21. Manifesto for Agile Software Development. – URL: <https://agilemanifesto.org/> (mode of access: 10.09.2022). – Text : electronic.
22. Walls, M. Building a DevOps Culture / M. Walls. – Sebastopol, CA : O'Reilly Media, 2013. – 26 p.

РЕФЕРЕНС

1. Bashmakov, M. I. (2000). *Chto takoe produktivnoe obuchenie?* [What Is Productive Learning?]. In *Teoriya i praktika produktivnogo obucheniya: kollektivnaya monografiya*. Moscow, Narodnoe obrazovanie, pp. 6-14.
2. Bem, I., Schneider, I. (1999). Produktivnoe obuchenie: slagaemye sistemy [Productive Learning: Components of the System]. In *Sovmestnyi vypusk zhurnalov «Shkol'nye tekhnologii»*. No. 4; «*Novye tsennosti obrazovaniya*». No. 9, pp. 59-70.
3. Bespalko, V. P. (1989). *Slagaemye pedagogicheskoi tekhnologii* [Components of Pedagogical Technology]. Moscow, Pedagogika. 192 p.
4. Beatty, D., Vigers, K. (2014). *Razrabotka trebovaniy k programmnomu obespecheniyu* [Development of Software Requirements]. Moscow, Izdatel'stvo «Russkaya redaktsiya», Saint Petersburg, BKhV-Peterburg. 736 p.
5. Braude, E. (2004). *Tekhnologiya razrabotki programmnogo obespecheniya* [Technology of Software Development]. Saint Petersburg, Piter. 655 p.
6. Vasilyeva, E. N. (2016). *Interaktivnoe obuchenie kak model' produktivnogo uchebnogo protsessa v usloviyakh vvedeniya FGOS* [Interactive Learning as a Model of a Productive Educational Process in the Context of the Introduction of the Federal State Educational Standard]. In *Aktual'nye problemy kachestva matematicheskoi podgotovki shkol'nikov i studentov: metodologicheskii, teoreticheskii i tekhnologicheskii aspekty: mater. IV vseros. nauch.-metod. konf. mezhdunar. nauch.-obraz. foruma «Chelovek, sem'ya, obshchestvo: istoriya i perspektivy razvitiya»*. Krasnoyarsk, pp. 70-75.
7. Winters, T., Manshrek, T., Wright, H. (2021). *Delai kak v Google. Razrabotka programmnogo obespecheniya* [Do as in Google. Software Development]. Saint Petersburg, Piter. 544 p.
8. Dokukina, I. A., Polyanin, A. V. (2020). *Tsifrovye platformy v publichnom upravlenii na osnove Agile menedzhmenta* [Digital Platforms in Public Administration Based On Agile Management]. In *Vestnik Voronezhskogo gosudarstvennogo universiteta. Seriya: Ekonomika i upravlenie*. No. 1, pp. 126-131.
9. Davis J., Daniels K. (2017). *Filosofiya DevOps. Iskusstvo upravleniya IT* [Philosophy of DevOps. The Art of IT Management]. Saint Petersburg, Piter. 416 p.
10. Kalmykova, Z. I. (1981). *Produktivnoe myshlenie kak osnova obuchaemosti* [Productive Thinking as a Basis for Learning]. Moscow, Pedagogika. 200 p.
11. Curran, T. (2005). *Obshcheevropeiskie shkaly otsenok: opyt natsional'nykh sistem i ECTS* [Pan-European Grading Scales: The Experience of National Systems and ECTS]. In *Vysshee obrazovanie v Europe*. Vol. XXIX. No. 1, p. 48.
12. Kim, J., Debois P., Willis J., Humble J. (2018). *Rukovodstvo po DevOps. Kak dobit'sya gibkosti, nadezhnosti i bezopasnosti mirovogo urovnya v tekhnologicheskikh kompaniyakh* [DevOps Guide. How to Achieve World-Class Flexibility, Reliability and Security in Technology Companies] / transl. by I. Leiko, I. Vasilyev, ed. by N. Korytko. Moscow, Mann, Ivanov i Ferber. 512 p.
13. Kugel, L. A. (2015). *Obuchenie studentov algoritmizatsii i programmirovaniyu na osnove strukturno-algoritmicheskogo podkhoda k postanovke i realizatsii zadach: na primere napravleniya podgotovki bakalavrov «prikladnaya informatika»* [Teaching Students Algorithmization and Programming Based on a Struc-

tural-Algorithmic Approach to Setting and Implementing Tasks: On the Example of the Direction of Training Bachelors "Applied Informatics"]. Avtoref. dis. ... kand. ped. nauk. Moscow. 22 p.

14. Martin, R. (2018). *Chistaya arkhitektura. Iskusstvo razrabotki programmnogo obespecheniya* [Pure Architecture. The Art of Software Development]. Saint Petersburg, Piter. 352 p.

15. Osokina, E. V. (2011). *Ispol'zovanie metoda kollektivnogo proektirovaniya pri obuchenii budushchikh spetsialistov v oblasti informatsionnykh tekhnologii razrabotke informatsionnykh system* [Using the Method of Collective Design in Teaching Future Specialists in the Field of Information Technology to Develop Information Systems]. Dis. ... kand. ped. nauk. Shadrinsk. 172 p.

16. Rubin, S. (2020). *Osnovy Scrum: prakticheskoe rukovodstvo po gibkoi razrabotke PO* [Fundamentals of Scrum: A Practical Guide to Agile Software Development]. Saint Petersburg, OOO «Dialektika». 544 p.

17. Hering, M. (2020). *DevOps dlya sovremennoego predpriyatiya* [DevOps for the Modern Enterprise] / transl. by M. A. Raitman. Moscow, DMK Press. 232 p.

18. *Atlassian official site*. URL: <https://www.atlassian.com/ru/agile/kanban> (mode of access: 10.09.2022).

19. Georgiou, H. (2020). Characterising Communication of Scientific Concepts in Student-Generated Digital Products. In *Educ. Sci.* No. 10, pp. 18.

20. Knoll, M. (1997). The Project Method: Its Vocational Education Origin and International Development. In *Journal of Industrial Teacher Education*. No. 34 (3), pp. 59-80.

21. *Manifesto for Agile Software Development*. URL: <https://agilemanifesto.org/> (mode of access: 10.09.2022).

22. Walls, M. (2013). *Building a DevOps Culture*. Sebastopol, CA, O'Reilly Media. 26 p.