

DOI: 10.37791/2687-0657-2023-17-4-80-89

Оценка эффективности применения технологий моб-программирования в управлении IT-проектами

М. С. Абдалов^{1*}, А. Е. Трубин¹, А. М. Нечаев¹, Д. Н. Чаусов^{1,2}

¹ Университет «Синергия», Москва, Россия

² Федеральный исследовательский центр Институт общей физики им. А. М. Прохорова Российской академии наук, Москва, Россия

* maxim.abdalov@gmail.com

Аннотация. В современном мире компании, занимающиеся разработкой программного обеспечения, сталкиваются с рядом проблем, таких как увеличение объема кода, увеличение числа ошибок, повышение сложности проектов и рост требований к качеству и скорости разработки. Для решения этих проблем необходимо снижать издержки, повышать производительность команды, сокращать время на тестирование и отладку кода, а также повышать качество своей продукции, то есть добиваться конкурентных преимуществ. Целью исследования является поиск решений для достижения конкурентных преимуществ IT-компании с целью повышения эффективности управления проектами разработки и повышения качества продукта посредством применения технологий моб-программирования. Данный подход не был достаточно исследован в контексте Kanban-методологии управления рабочим процессом. Таким образом, исследования в области использования моб-программирования в Kanban позволяют расширить знания о возможностях этой методологии и ее применимости в различных условиях. В статье применялось экспериментальное исследование, а именно проведение экспериментов для сравнения эффективности работы и коммуникации команд, использующих моб-программирование в рамках Kanban, с командами, не использующими эту методологию. Также использовался аналитический метод исследования, сравнительный анализ (для выделения преимуществ и недостатков подхода Mob Programming), табличный метод предоставления информации для большей наглядности, интервьюирование.

Ключевые слова: управление проектами, коммуникация в команде, разработка программного обеспечения, моб-программирование, коллективное программирование, экстремальное программирование, Kanban, Agile-методологии

Для цитирования: Абдалов М. С., Трубин А. Е., Нечаев А. М., Чаусов Д. Н. Оценка эффективности применения технологий моб-программирования в управлении IT-проектами // Современная конкуренция. 2023. Т. 17. № 4. С. 80–89. DOI: 10.37791/2687-0649-2023-17-4-80-89

Performance Evaluation of Application Mob Programming Technologies in IT Project Management

M. Abdalov^{1*}, A. Trubin¹, A. Nechaev¹, D. Chausov^{1,2}

¹ Synergy University, Moscow, Russia

² Prokhorov General Physics Institute of the Russian Academy of Sciences, Moscow, Russia

* maxim.abdalov@gmail.com

Abstract. In the modern world, software development companies face a number of problems, such as an increase in the volume of code, an increase in the number of errors, an increase in the complexity of projects and an increase in requirements for quality and speed of development. To solve these problems, it is necessary to reduce costs, increase team productivity, reduce time for testing and debugging code, as well as improve the quality of their products, i. e. achieve competitive advantages by improving project management systems and communication in the team. The aim of the research is to find solutions to achieve competitive advantages of an IT company to improve the efficiency of project management and improve product quality through the use of mobile programming technologies. This approach has not been sufficiently investigated in the context of the Kanban methodology for workflow management. Thus, research in the field of using mobile programming in Kanban allows us to expand knowledge about the possibilities of this methodology and its applicability in various conditions. The article used experimental research, namely conducting experiments to compare the effectiveness of work and communication of teams using mobile programming within Kanban with teams not using this methodology. The analytical method of research, comparative analysis (to highlight the advantages and disadvantages of the Mob Programming approach), the tabular method of providing information, for greater clarity, interviewing was also used.

Keywords: project management, team communication, software development, mobile programming, collective programming, extreme programming, Kanban, Agile methodologies

For citation: Abdalov M., Trubin A., Nechaev A., Chausov D. Performance Evaluation of Application Mob Programming Technologies in IT Project Management. *Sovremennaya konkurentsya=Journal of Modern Competition*, 2023, vol.17, no.4, pp.80-89 (in Russian). DOI: 10.37791/2687-0649-2023-17-4-80-89

Введение

Современная разработка программного обеспечения требует эффективных методологий, позволяющих обеспечить максимальную скорость, качество и прозрачность работы команды. Одним из наиболее распространенных подходов к управлению командой является Agile, который представляет собой совокупность методов гибкого управления проектами и командами разработчиков, ориентированную

на коллективную работу и быстрое реагирование на изменения в процессе разработки, т. е. коммуникации в команде.

Разработка с использованием методологии Agile в большинстве своем подходит многим командам для управления проектами [6].

Постоянное усиление конкуренции между предприятиями заставляет компании стремиться к изменениям, планировать задачи с использованием современных подходов, таких как Agile, для эффективности работы компании [7].

К этой группе методик относится методология Kanban, представляющая собой системный подход к управлению рабочим процессом, который основывается на визуальном отображении задач и этапов их выполнения.

Однако в ходе работы у Kanban-команд могут возникать проблемы, такие как слабая коммуникация, неоптимальное распределение рабочей нагрузки или необходимость постоянного обучения новых сотрудников [2].

При использовании различных методологий в управлении проектами важно подобрать именно ту, что наиболее полно будет подходить под заданные требования компании, к задачам, которые требуется выполнить [1].

В период экономического кризиса перед компаниями встает вопрос о сохранении денежных средств, сокращении убытков. На смену одним компаниям приходят другие и пытаются занять их нишу. Во время кризиса компаниям важно не только устоять на рынке, но и попробовать новые «тренды» в управлении процессами, для чего необходимо использовать гибкие методологии [3].

Для решения этих проблем была создана методология Mob Programming. Она предлагает процесс разработки программного обеспечения, при котором команда работает над одной задачей вместе за одним компьютером. Каждый член команды активно участвует в процессе разработки и вносит свой вклад в общее решение задачи [10]. Процесс моб-программирования подразумевает наличие ведущего программиста, который руководит командой и координирует работу всех ее участников.

Большое количество положительных моментов несет в себе использование технологий парного программирования для бизнеса, таких как:

1. Меньше остановок. Пара обсуждает все возникающие вопросы между собой, не отвлекаясь от процесса реализации поставленных задач.

2. Быстрее и лучше. Осуществляется обмен знаниями и трудностями во время работы.

3. Меньше ошибок в коде. Опечатки, пробелы с большой долей вероятности будут обнаружены в ходе работы.

4. Повышение своих знаний за счет парного исследования задачи и взаимопомощи, что для бизнеса может быть преимуществом, так как осуществляется экономия средств на обучении сотрудников.

5. Повышение мотивации у сотрудников.

Два разработчика по очереди кодируют или рецензируют работу друг друга по ходу дела. Регулярная ротация ролей помогает сохранять бдительность и вовлеченность обоих разработчиков. В организациях также может быть пара ротационных ролей для работы над разными задачами. Таким образом, они получают опыт работы над различными частями строящейся системы [11].

Цель данной научной статьи – доказать эффективность от применения методологии Mob Programming в Agile-разработке в управлении IT-проектами и выявить ее достоинства и недостатки. В статье будут также рассмотрены примеры успешного применения этой методологии на практике и детально продемонстрированы принципы работы данного метода.

История происхождения подхода

Рабочий процесс в Agile-методиках разбивается на небольшие временные промежутки, их еще называют спринтами (от англ. sprint – бег на короткую дистанцию) или итерациями. Спринт – это основной промежуток времени в Agile-методологии разработки программного обеспечения, в течение которого команда разработки фокусируется на достижении конкретных целей и поставленных задач. Во время каждого спринта команда разработки создает часть продукта, которую можно протестировать и оценить [4].

Методология Kanban – это одна из методологий семейства Agile, которая представляет собой системный подход к управлению рабочим процессом, который основывается

на визуальном отображении задач и этапов их выполнения на доске Kanban. Каждая задача представлена карточкой, которая перемещается по колонкам доски, отображающей этапы работ, начиная с бэклога (списка задач) и заканчивая окончанием выполнения. Визуальное отображение позволяет лучше контролировать ход работы, определять проблемы и узкие места в процессе. Методология Kanban также может включать в себя принципы ограничения рабочего потока, управления временем выполнения задач, обеспечения непрерывности и постоянного улучшения процесса в целом.

Моб-программирование – это коллективный подход к разработке, который нацелен на повышение эффективности работы команды и улучшение качества программного обеспечения. Этот подход позволяет программистам работать в одном пространстве над одним и тем же кодом, обсуждая и исправляя ошибки в режиме реального времени.

История моб-программирования началась в 2011 году в компании Hunter Industries, где два программиста, Вуддинг и Ларсен, основали группу под названием «Общий балкон». Эта группа объединила в себе несколько разработчиков, которые начали работать вместе над проектом, общаясь и взаимно обмениваясь опытом.

Использование парного программирования в решении задач является одним из принципов экстремального программирования. Основной целью методики использования экстремального программирования является адаптация к постоянному изменению требований, без ухудшения качества разработки. Использование парного программирования позволяет достигать решения поставленных задач без ухудшения качества поставляемого кода, а в большей степени даже способствует его улучшению [8].

Суть моб-программирования заключается в том, чтобы обеспечить максимальный уровень синхронизации и диалога между членами команды. В рамках моб-

программирования все участники проекта работают в одном темпе, что позволяет быстро решать проблемы и устранять ошибки.

Принципы работы моб-программирования заключаются в сотрудничестве, обмене знаниями и навыками, а также уважении мнения каждого участника команды. Для эффективной работы необходимо установить строгое расписание, где каждый из программистов будет занят определенными задачами в течение определенных периодов времени.

Моб-программирование приносит огромную пользу, так как помогает добиваться лучших результатов разработки и повышает слаженность команды. Эта методология дает возможность ускорить работу над проектом и получить более качественный результат, а также позволяет улучшить навыки программирования и обмен опытом с другими участниками команды [9].

Преимущества использования Mob Programming в Agile-разработке

Mob Programming стал популярным подходом в Agile-разработке благодаря нескольким преимуществам, которые он предоставляет командам разработчиков (табл. 1).

Таким образом, использование Mob Programming в Agile-разработке предлагает множество преимуществ по сравнению с традиционным подходом в Scrum-разработке. Он повышает скорость и качество разработки, улучшает коммуникацию и сотрудничество в команде, сокращает время на обучение новых сотрудников и улучшает качество кода, снижает риски и повышает прозрачность в процессе работы благодаря тому, что вся команда работает над одной задачей в конкретный момент времени.

Важно понимать, что Mob Programming не является универсальным решением для всех проектов. В таблице 2 выделены основные типы проектов по применимости в них подхода Mob Programming.

Таблица 1. Преимущества использования подхода Mob Programming в Agile-разработке

Table 1. Benefits of using the Mob Programming approach in Agile development

Показатель <i>Index</i>	Описание <i>Description</i>
Улучшенное качество и скорость разработки <i>Improved quality and development speed</i>	Все члены команды работают вместе, обмениваясь идеями и знаниями, что приводит к лучшему качеству кода и ускоряет процесс разработки <i>All team members work together to share ideas and knowledge, resulting in better code quality and faster development</i>
Уменьшение количества ошибок <i>Reducing the number of errors</i>	Когда много людей работает над одним проектом, больше шансов, что любые ошибки будут выявлены и исправлены раньше, чем они окажутся на production <i>When many people work on the same project, any bugs are more likely to be caught and fixed before they get to production</i>
Распределение знаний <i>Knowledge distribution</i>	Все участники могут учиться друг у друга и получать обратную связь, что помогает распределять знания и увеличивать общую компетенцию команды <i>All participants can learn from each other and receive feedback, which helps to distribute knowledge and increase the overall competence of the team</i>
Улучшенное решение конфликтов <i>Improved conflict resolution</i>	Данный подход помогает членам команды эффективно решать конфликты и споры путем сотрудничества и общения друг с другом <i>The approach helps team members to effectively resolve conflicts and disputes by collaborating and communicating with each other</i>
Равноправный доступ к информации <i>Equitable access to information</i>	Все члены команды имеют равный доступ к информации, что снижает вероятность ошибок и повышает качество работы <i>All team members have equal access to information, which reduces the likelihood of errors and improves the quality of work</i>

Таблица 2. Применимость Mob Programming в разных типах проектов

Table 2. Applicability of Mob Programming in different types of projects

Проекты, для которых применима методика Mob Programming <i>Projects for which the Mob Programming methodology is applicable</i>	Проекты, для которых не применима методика Mob Programming <i>Projects for which the Mob Programming methodology is not applicable</i>
Большие и сложные проекты, требующие интенсивного этапа разработки <i>Large and complex projects requiring an intensive development phase</i>	Проекты, требующие индивидуальной работы и независимости каждого члена команды <i>Projects that require individual work and independence of each team member</i>
Проекты, которые требуют высокого уровня безопасности и качества кода <i>Projects that require a high level of security and code quality</i>	Маленькие проекты, для которых непозволительно использование ресурсов на моб-программирование <i>Small projects for which it is not allowed to use resources for mobile programming</i>
Проекты, которые требуют высокой скорости разработки и большого количества функциональности <i>Projects that require high development speed and a lot of functionality</i>	Проекты, требующие разделения команды на группы для выполнения различных задач <i>Projects that require the team to be divided into groups to perform various tasks</i>
Проекты, которые требуют особых профессиональных навыков, которые распределены между участниками команды <i>Projects that require specific professional skills that are shared among team members</i>	Проекты, в которых необходимо быстрое решение проблемы и отладка <i>Projects that require quick problem resolution and debugging</i>

Проведение эксперимента с использованием Mob Programming в команде разработки, работающей по методологии Kanban

Эксперимент проводился в компании «Локалхост», занимающейся разработкой программного обеспечения внутренних потребностей холдинга.

Компания является разработчиком программного обеспечения для своего холдинга, частью которого она является. В состав холдинга входит ряд предприятий различной направленности бизнеса, такие как отельный бизнес, недвижимость и т. д. Основная цель компании «Локалхост» заключается в создании высококачественных программных продуктов, которые полностью удовлетворяют нужды холдинга и помогают его эффективному функционированию. Работа в компании организована в соответствии с методологией Scrum, что позволяет ей процессуализировать и организовывать работу над проектами. Scrum – гибкая и итеративная методология разработки, которая позволяет эффективно управлять проектами с четкими сроками и требованиями. Компания занимается всем жизненным циклом разработки ПО – от составления требований и анализа до проектирования, разработки, тестирования и внедрения готовых продуктов. Специалисты компании обладают глубокими знаниями и опытом в различных областях разработки, таких как веб-приложения, мобильные приложения, базы данных и многое другое.

Одним из ключевых аспектов работы компании «Локалхост» является постоянное взаимодействие с холдингом, чтобы полностью понимать его нужды и задачи. Команда разработчиков находится в постоянном контакте с представителями руководства холдинга, чтобы уточнять требования, согласовывать изменения и обеспечивать эффективную коммуникацию во время разработки.

Для эксперимента был выбран проект, который соответствует критерию применимости, приведенному в таблице 2. Он интенсивно развивается, имеет большое количе-

ство функций и зависимостей, а также является важным с точки зрения безопасности продукта, поскольку данным сервисом пользуются как сотрудники компании, так и внешние контрагенты.

Первые шаги внедрения Mob Programming в команду были связаны с обучением участников новому способу работы. Разработчики проходили обучение вместе, изучали новые правила и учились эффективно использовать инструменты, которые позволяют работать в режиме тандемного программирования. Это в первую очередь IDE (Integrated Development Environment) – интегрированная среда разработки программного обеспечения, она объединяет несколько инструментов, которые нужны для разработки. IDE позволяет программистам работать над созданием приложений или программ на одной платформе, в режиме парного программирования; GitLab – веб-сервис для управления репозиториями Git и процессами непрерывной интеграции и развертывания, он позволяет командам разработки управлять проектами, создавать и хранить исходный код в репозитории, координировать работу над проектом, отслеживать ошибки, запускать автоматические тесты, настраивать системы сборки и развертывания приложений); Code-review – процесс проверки и анализа исходного кода программы другими членами команды разработки, целью обзора кода является выявление ошибок, уязвимостей, потенциальных проблем и улучшение качества кода в целом; Google Docs для ведения совместной документации в режиме реального времени.

Команда разработки была разделена на 2 группы – контрольную (K1) и экспериментальную (K2). Экспериментальная группа перешла на разработку с использованием Mob Programming, а контрольная продолжила работу по традиционным правилам Kanban-методологии.

Экспериментальная команда начала использовать Mob Programming на регулярной основе. Каждый день, в начале работы, участники собирались вместе и выбирали одну задачу, которую нужно было решить.

Каждый из разработчиков занимал свое место за компьютером, а один человек становился ведущим. Он описывал, что нужно сделать, какие функции будут реализованы, какие тесты будут написаны.

Участники начинали работать в режиме парного программирования, обсуждая решения и находя лучшие варианты. При необходимости к команде могли присоединиться другие разработчики, чтобы поделиться своим мнением и опытом. В результате такой работы команда получила большую отдачу от каждого члена команды.

В процессе эксперимента рассматривались такие показатели, как скорость работы и количество ошибок, выявленных командой тестировщиков в выпускаемом командой разработчики продукте. Время на проведение эксперимента было ограничено 1 месяцем, работа проводилась недельными спринтами, скорость работы измерялась в стори-поинтах.

Стори-поинт – это мера сложности, используемая для оценки задач, которые не могут быть точно оценены временем. Это практика Agile-разработки, которая основывается на оценке работы с помощью относительно значимых единиц измерения. Оценка стори-поинтами основывается на восприятии сложности и затрат времени на выполнение задания. Например, задача может быть оценена в 2 стори-поинта, если она считается более простой, или 8 стори-поинтов, если предполагается, что она более сложная и требует больше времени. Оценка стори-поинтов производится командой разработчиков и определяется после детального обсуждения функционала и технических условий для реализации задачи. Оценка может

быть пересмотрена в ходе разработки, если возникают технические сложности или изменяются требования к проекту.

После завершения второго спринта были подведены промежуточные результаты и проведен анализ того, как проявляет себя каждый из подходов (табл. 3).

Из таблицы 3 видно, что после эксперимента, проводимого в течении двух недель, увеличилась производительность K2 и уменьшилось количество выявленных ошибок данной группы. Для определения причин увеличения производительности и уменьшения ошибок в группе K2 было проведено интервью, по итогам которого удалось выяснить, что к повышению эффективности работы группы привели следующие факторы:

- **Общее понимание задачи:** каждый участник команды имеет возможность высказаться и выразить свою точку зрения на вопрос решения задачи, что позволяет всем членам команды получить общее понимание задачи.
- **Коллективные знания:** в рамках Mob Programming разработчики могут обмениваться знаниями и опытом, что позволяет повысить качество программного обеспечения и производительность команды.
- **Mob Programming** позволяет более эффективно выявлять и устранять ошибки в коде. Большинство ошибок можно обнаружить и устранить раньше, чем они станут серьезной проблемой.
- **Большой по сравнению с работой в команде K1 уровень коммуникаций** между членами команды. Улучшение качества общения и вовлеченности в проект каждого члена команды.

Таблица 3. Средние значения за период 2 недели

Table 3. Average values for a period of 2 weeks

Показатель <i>Index</i>	K1 <i>K1</i>	K2 <i>K2</i>	Эффективность, % <i>Efficiency, %</i>
Скорость разработки <i>Development speed</i>	64 стори-поинта <i>64 story points</i>	89 стори-поинтов <i>89 story points</i>	+28
Количество выявленных ошибок за спринт <i>Number of identified bugs per sprint</i>	15	8	+47

Промежуточные итоги эксперимента показали значительное влияние на результаты работы K2, эксперимент был продолжен. Итоговые результаты представлены в таблице 4. После подведения итогов эксперимента удалось установить, что в экспериментальной команде скорость разработки увеличилась на 23%, а количество ошибок уменьшилось на 63%.

В результате внедрения Mob Programming удалось достичь:

- ускорения работы и повышения качества кода;
- улучшения связи внутри команды и принятия лучших решений;

- более эффективного использования ресурсов и улучшения процесса обучения;
- повышения качества приложения и улучшения процесса разработки.

Недостатки и минусы использования подхода Mob Programming

Mob Programming как подход может иметь ряд недостатков (табл. 5) [5].

Кроме того, использование Mob Programming требует больших затрат на команду разработчиков, что может оказаться проблемой для небольших и средних компа-

Таблица 4. Средние значения за период 1 месяц

Table 4. Average values for a period of 1 month

Показатель <i>Index</i>	K1 <i>K1</i>	K2 <i>K2</i>	Эффективность, % <i>Efficiency, %</i>
Скорость разработки <i>Development speed</i>	75 стори-поинтов <i>75 story points</i>	92 стори-поинта <i>92 story points</i>	+23
Количество выявленных ошибок за спринт <i>Number of identified bugs per sprint</i>	27	10	+63

Таблица 5. Недостатки подхода Mob Programming

Table 5. Disadvantages of the Mob Programming approach

Показатель <i>Index</i>	Описание <i>Description</i>
Сложности с организацией рабочего пространства <i>Difficulties with the organization of the workspace</i>	Подход требует хорошо организованного рабочего пространства, оборудованного соответствующими инструментами и программными средствами, чтобы обеспечить эффективную работу команды. Некоторые команды могут испытывать трудности с обеспечением такого пространства, что может оказаться дополнительной проблемой при использовании Mob Programming <i>The approach requires a well-organized workspace equipped with the right tools and software to keep the team working efficiently. Some teams may have difficulty securing this space, which can be an additional challenge when using Mob Programming</i>
Применимость к проектам и задачам <i>Applicability to projects and tasks</i>	Этот метод не обязательно подходит для всех типов проектов и задач. Например, если проект требует разделения задач на частные подзадачи, которые лучше выполнять индивидуально, то в таком случае использование Mob Programming может не привести к ожидаемым результатам <i>This method is not necessarily suitable for all types of projects and tasks. For example, if the project requires the division of tasks into separate subtasks that are best performed individually, then the use of Mob Programming may not lead to the expected results</i>
Высокий уровень коммуникаций <i>High level of communications</i>	Mob Programming требует высокого уровня коммуникаций, и если в команде с этим есть сложности, то он не подойдет <i>Mob Programming requires a high level of communication, and if the team has difficulty with this, then it will not work</i>

ний. Также методология требует дисциплины (чтобы вся команда находилась на «одной волне») и перерывов перед сменами программистов. Это может оказаться неудобно для некоторых программистов, которые предпочитают работать в одиночку.

Наконец, Mob Programming может быть менее эффективен при работе с разработчиками, работающими в разных часовых поясах, так как требует включения в работу всей команды непосредственно в офисе или посредством таких инструментов, как Skype, Zoom.

Заключение

В процессе проведения исследования, целью которого было нахождение решения для улучшения процесса разработки программного обеспечения и повышения качества продукта путем коллективной работы разработчиков в рамках методологии Kanban, было установлено, что в случае использования методики Mob Programming удастся достичь определенных улучшений в производительности, а также уменьшения количества ошибок в некоторых типах проектов.

Был проведен эксперимент, который наглядно показал, что Mob Programming

действительно положительно влияет на результаты работы команды, однако он требует увеличения ресурсов команды, увеличения коммуникаций внутри команды, поэтому было установлено, что данный подход не может использоваться во всех типах проектов.

Рекомендации по применению методологии Mob Programming зависят от типа проекта, его особенностей и требований заказчика [5]. Она может быть использована как для разработки сложных программных продуктов, так и для решения бизнес-задач в других областях. Рекомендуется внедрять ее в командах сильных и опытных специалистов, готовых работать в условиях высокой командной ответственности.

Перспективы развития и доработки методологии связаны с углублением ее интеграции с другими Agile-практиками, внедрением новых инструментов и технических решений для повышения комфорта командного взаимодействия и улучшения качества процессов разработки [4]. Также возможна дальнейшая стандартизация и формализация подхода для широкого использования в различных областях бизнеса.

Список литературы

1. *Алексеева Н. В., Горчаков М. В.* Цифровизация управления проектами на основе гибких методологий // *Горизонты экономики.* 2021. №6 (65). С. 6–12.
2. *Гаязова Е. Э., Стригунов В. В.* О применении методологии Kanban в разработке программного обеспечения // *Дневник науки.* 2019. №6 (30). С. 1–8.
3. *Горчаков М. В.* Управление удаленными командами при реализации проектов в сфере информационных технологий // *Горизонты экономики.* 2020. №2 (55). С. 55–58.
4. Как реализовать стратегию цифровой трансформации [Электронный ресурс] // Skillfactory media. URL: <https://blog.skillfactory.ru/chto-takoe-strategiya-cifrovoj-transformaczi/> (дата обращения: 12.07.2023).
5. *Колесников А. М., Будагов А. С., Мухин К. Ю.* Гибкое управление проектами Agile: анализ ключевых положительных сторон, недостатков, требований, статистики реализации // *Экономический вектор.* 2018. №3 (14). С. 53–57.
6. *Куликова С. В., Круглов А. А.* Разработка метода непрерывного совершенствования услуг с использованием инструментов Business Intelligence // *Открытое образование.* 2018. Т. 22. №1. С. 83–92.
7. *Локтионов Д. А., Масловский В. П.* Критерии применения Agile-методологии для управления проектом // *Креативная экономика.* 2018. Т. 12. №6. С. 839–854. DOI: 10.18334/ce.12.6.39179.
8. *Маношин Д. А.* Особенности методологии организации разработки программного обеспечения extreme programming (экстремальное программирование) // *Colloquium-Journal.* 2019. №12-2 (36). С. 24–25.
9. *Сазерленд Дж.* Scrum. Революционный метод управления проектами / пер. с англ. М. Гескиной. – М.: Манн, Иванов и Фербер, 2016. – 288 с.
10. *Brooks F. P.* The Mythical Man-Month: Essays on Software Engineering. – 1st ed. – Addison-Wesley, 1975. – 200 p.
11. *Gillis A. S.* What is pair programming? // TechTarget. URL: <https://www.techtarget.com/searchsoftwarequality/definition/Pair-programming#:~:text=Pair%20programming%20is%20an%20Agile,code%20and%20test%20user%20stories> (дата обращения: 12.07.2023).

Сведения об авторах

Абдалов Максим Сергеевич, ORCID 0000-0003-4128-994X, аспирант, кафедра цифровой экономики, Университет «Синергия», Москва, Россия, maxim.abdalov@gmail.com

Трубин Александр Евгеньевич, ORCID 0000-0002-7189-5679, канд. экон. наук, доцент, заведующий кафедрой цифровой экономики, Университет «Синергия», Москва, Россия, niburt@yandex.ru

Нечаев Андрей Михайлович, ORCID 0000-0003-1917-3975, канд. воен. наук, доцент, кафедра цифровой экономики, Университет «Синергия», Москва, Россия, Nechaev_a@inbox.ru

Чausов Денис Николаевич, ORCID 0000-0002-1287-6427, док. физ.-мат. наук, профессор, кафедра цифровой экономики, Университет «Синергия»; заведующий Лабораторией фотоники и органической электроники, Федеральный исследовательский центр Институт общей физики им. А. М. Прохорова Российской академии наук (ИОФ РАН), Москва, Россия, d.chausov@yandex.ru

Статья поступила 13.07.2023, рассмотрена 26.07.2023, принята 07.08.2023

References

1. Alekseeva N. V., Gorchakov M. V. Digitalization of project management based on flexible methodologies. *Gorizonty ekonomiki*, 2021, no.6(65), pp.6-12 (in Russian).
2. Gayazova E. E., Strigunov V. V. On the application of Kanban methodology in the development of software. *Dnevnik nauki*, 2019, no.6(30), pp.1-8 (in Russian).
3. Gorchakov M. V. Management of remote teams in the implementation of projects in the field of information technologies. *Gorizonty ekonomiki*, 2020, no.2(55), pp.55-58 (in Russian).
4. *Kak realizovat' strategiyu cifrovoj transformacii* [How to implement a digital transformation strategy]. Skillfactory media. Available at: <https://blog.skillfactory.ru/chto-takoe-strategiya-cifrovoj-transformacii/> (accessed 12.07.2023).
5. Kolesnikov A. M., Budagov A. S., Mukhin K. Yu. Flexible project management Agile: analysis of key positive aspects, shortcomings, requirements, implementation statistics. *Ekonomicheskii vektor*=Economic Vector, 2018, no.3(14), pp.53-57 (in Russian).
6. Kulikova S. V., Kruglov A. A. Development of a method of continuous improvement of services using the business intelligence tools. *Otkrytoe obrazovanie*=Open Education, 2018, vol.22, no.1, pp.83-92 (in Russian).
7. Loktionov D. A., Maslovskiy V. P. Criteria for applying the Agile methodology for project management. *Kreativnaya ekonomika*, 2018, vol.12, no.6, pp.839-854 (in Russian). DOI: 10.18334/ce.12.6.39179.
8. Manoshin D. A. The specificity of the methodology of software development in extreme programming. *Colloquium-Journal*, 2019, no.12-2(36), pp.24-25 (in Russian).
9. Sutherland J. *Scrum. Revolyutsionnyi metod upravleniya proektami* [Scrum. Revolutionary method of project management]. Transl. from English M. Geskina. Moscow, *Mann, Ivanov and Ferber Publ.*, 2016, 288 p.
10. Brooks F. P. *The Mythical Man-Month: Essays on Software Engineering*. 1st ed. Addison-Wesley, 1975, 200 p.
11. Gillis A. S. What is pair programming? TechTarget. Available at: <https://www.techtarget.com/searchsoftwarequality/definition/Pair-programming#:~:text=Pair%20programming%20is%20an%20Agile,code%20and%20test%20user%20stories> (accessed 12.07.2023).

About the authors

Maxim S. Abdalov, ORCID 0000-0003-4128-994X, Postgraduate, Digital Economy Department, Synergy University, Moscow, Russia, maxim.abdalov@gmail.com

Alexander E. Trubin, ORCID 0000-0002-7189-5679, Cand. Sci. (Econ.), Associate Professor, Head of Digital Economy Department, Synergy University, Moscow, Russia, niburt@yandex.ru

Andrey M. Nechaev, ORCID 0000-0003-1917-3975, Cand. Sci. (Military), Associate Professor, Digital Economy Department, Synergy University, Moscow, Russia, Nechaev_a@inbox.ru

Denis N. Chausov, ORCID 0000-0002-1287-6427, Dr. Sci. (Phys.-Math.), Professor, Digital Economy Department, Synergy University; Head of Laboratories of Photonics and Organic Electronics, Prokhorov General Physics Institute of the Russian Academy of Sciences, Moscow, Russia, d.chausov@yandex.ru

Received 13.07.2023, reviewed 26.07.2023, accepted 07.08.2023