

ГИБКИЕ МЕТОДОЛОГИИ В УПРАВЛЕНИИ КОМАНДОЙ РАЗРАБОТЧИКОВ IT-ПРОЕКТА

FLEXIBLE METHODOLOGIES IN MANAGING AN IT PROJECT DEVELOPMENT TEAM

ПОЛУЧЕНО 06.11.2023 ОДОБРЕНО 13.11.2023 ОПУБЛИКОВАНО 25.12.2023

УДК 331 DOI 10.12737/2305-7807-2023-12-6-24-28

**ИЛЬЧЕНКО Е.Н.**

Канд. экон. наук, доцент кафедры «Государственное и муниципальное управление, внешнеэкономическая деятельность и менеджмент», ФГБОУ ВО «Курганский государственный университет», г. Курган

ILCHENKO E.N.

Candidate of Economic Sciences, Associate Professor, Department of "State and Municipal Administration, Foreign Economic Activity and Management", Kurgan State University, Kurgan

e-mail: elnik01@mail.ru

Аннотация

Одним из важных вопросов при реализации стратегии социально — экономического развития страны и регионов является проблема управления интеллектуальными ресурсами компаний. В современных условиях развития экономики интеллектуальные ресурсы компаний приобретают все большее значение. Ключевой задачей предприятий в сфере IT становится повышение эффективности бизнес-процессов. Все большее значение приобретают проблемы, связанные с организацией и управлением командной работой. Выбор оптимального, в определенной ситуации, подхода к управлению командой играет значительную роль в развитии компаний. [1]. Результативность компаний, занимающихся разработкой программного обеспечения, во многом зависит от правильности выбранной методологии разработки программного обеспечения.

Целью исследования является изучение подходов к управлению интеллектуальными ресурсами компаний, а также выбор оптимального подхода к управлению командой разработчиков IT-проектов. В статье представлены результаты исследования сложившихся методологий, а также сделана попытка формирования оптимальной и адекватной модели управления командой разработчиков проекта.

Ключевые слова: команда, управление, командообразование, методология, разработка программного обеспечения.

Abstract

One of the important issues in the implementation of the strategy for the socio-economic development of the country and regions is the problem of managing the intellectual resources of companies. In modern conditions of economic development, the intellectual resources of companies are becoming increasingly important. The key task of enterprises in the IT field is to increase the efficiency of ongoing business processes. Problems related to the organization and management of teamwork are becoming increasingly important. Choosing the optimal approach to team management in a certain situation plays a significant role in the development of companies. [1]. The performance of software development companies largely depends on the correctness of the chosen software development methodology.

The purpose of the study is to study approaches to managing the intellectual resources of companies, as well as choosing the optimal approach to managing a team of IT project developers. The article presents the results of a study of existing methodologies, and also makes an attempt to form an optimal and adequate model for managing a team of IT project developers.

Keywords: team, management, team building, methodology, software development.

ВВЕДЕНИЕ

В современное время в развитии экономики интеллектуальные ресурсы компаний приобретают все большее значение. Основными задачами предприятий в сфере IT являются регулярность предоставления услуг и повышение эффективности протекающих бизнес-процессов. Особенно важными становятся вопросы, связанные с организацией командной работы, так как выбор оптимального, в определенной ситуации, подхода к управлению командой оказывает влияние на эффективность деятельности предприятия [1]. Результативность компаний, занимающихся разработкой программного обеспечения, во многом зависит от правильности выбранной методологии разработки программного обеспечения.

МЕТОДОЛОГИЯ ИССЛЕДОВАНИЯ

Исследование базируется общенаучных методах исследования, таких как поиск, группировка, сравнительный анализ, сопоставление, обобщение подходов и выработка научной гипотезы.

В мировом опыте компаниями применяются два основных подхода к разработке ПО (рис. 1).

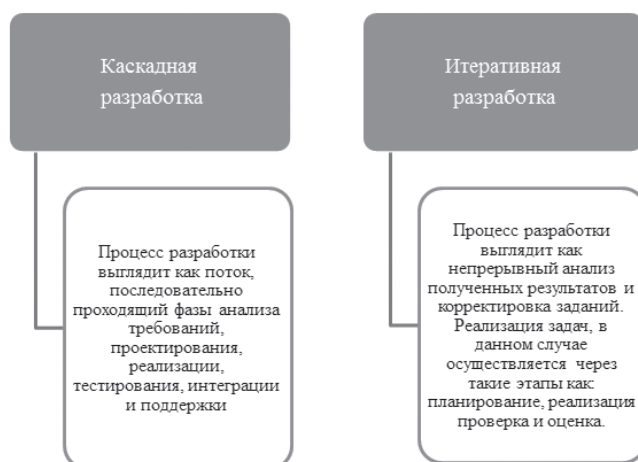


Рис. 1. Сравнительная характеристика подходов к разработке ПО

Таблица 1

Сравнительная характеристика методологий разработки ПО

Методология	Преимущества	Недостатки	Реализация командного подхода
Waterfall model	Хорошо работает в проектах, где требования могут быть четко определены и зафиксированы	1. Игнорирует динамические изменения 2. Сложно управлять рисками	Не обязательно
V-model	1. Планирование на ранних стадиях разработки системы ее тестирования. 2. Простота в использовании. 3. Упрощение отслеживания хода процесса разработки.	1. Сложность поддержки параллельных событий; 2. Не предусмотрены итерации между фазами 3. Поздние сроки тестирования требований Отсутствие в модели действий, направленные на анализ рисков.	Не обязательно
Spiral model	1. Улучшенный анализ рисков. 2. Возможность документации процесса разработки. 3. Раннее создание рабочих прототипов. 4. Возможность внесения изменений на поздних этапах.	1. Достаточно дорого в использовании; 2. Управление рисками требует привлечения высококлассных специалистов; 3. Не подходит для небольших проектов.	Не обязательно
RUP	1. Учитывает изменяющиеся требования. 2. Интеграция функций происходит постепенно. 3. Ранний выпуск продукта. 4. Постоянное обучение. 5. Постоянное улучшение продукта.	1. Недостаточный уровень формализма, приводящий к несогласованности решений, следовательно, к непродуктивным затратам ресурсов	Не обязательно
RAD	1. Использование прототипирования. 2. Обязательное вовлечение пользователей в процесс разработки	1. Модули проектируются изолированно, что приводит к большому количеству ошибок при интеграции. 2. Отсутствие или недостаточность документации.	Не обязательно
Scrum	1. Возможность быстрого запуска проекта с наиболее приоритетными функциями и минимально возможным бюджетом. 2. Ежедневный контроль над ходом работ. 3. Возможность внесения корректив в техническое задание	1. Не подразумевает наличие фиксированного бюджета и фиксированного технического задания, неприменимой для работы с государственными заказами, 2. Не работает при низкой квалификации команды, заниженных сроках работ или бюджете	Обязательно
Kanban	1. Нет четких сроков выполнения задачи. 2. Исключение из производства неэффективных запасов и материалов, за счет этого снижается себестоимость продукции. 3. Высокая гибкость программы	1. Внедрение возможно только в команды с численностью от 5 человек. 2. Не подходит для матричных структур организации предприятия. 3. Не подходит для долгосрочных стратегий.	Обязательно
Lean	1. Высокая организованность процессов позволяет полностью избежать ненужных затрат 2. Быстрый релиз продукта 3. Продукт с наименьшим количеством ошибок	1. Низкая вовлеченность персонала. 2. Работа опытных разработчиков. 3. Все принимаемые решения должны подкрепляться аналитическими данными.	Не обязательно
XP	1. Низкие затраты. 2. Проект может показать исключительную эффективность.	1. Может использоваться только в команде опытных разработчиков. 2. Нельзя развивать команду на несколько частей. 3. Размер команды ограничен числом в 10 – 15 человек.	Обязательно

Каждый подход имеет свои характеристики, которые необходимо учитывать в процессе работы над продуктом. Выбор того или иного подхода зависит от типа проекта (продукта), требований заказчика, сроков выполнения проекта, а также от наличия сформированной команды разработчиков.

Результаты проведенного исследования свидетельствуют о том, что каждая из изученных моделей подразумевает ее использование в разных ситуациях разработки продукта. Следует обратить внимание, что на современном этапе большее внимание уделяется методологиям, в которых реализуется командный подход. На рис. 2 представлены результаты исследования Agile Survey о популярности гибких методологий [2].

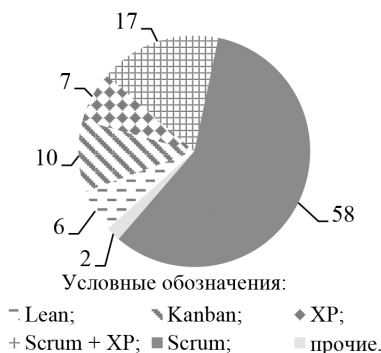


Рис. 2. Результаты исследования Agile Survey о популярности гибких методологий [1]

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

Результаты проведенного исследования свидетельствуют о целесообразности применения на российских пред-

приятиях сферы разработки программного обеспечения методологии Scrum. Гибкая методология Scrum предполагает командную работу разработчиков проекта, сосредоточена на взаимоотношениях внутри команды, а также на улучшении коммуникаций между ними [3]. В методологии Scrum под командой понимается самоорганизующаяся и самоуправляемая структура. В итоге оцениваются результаты работы не отдельного разработчика, а команды в целом, так как оценка индивидуальных результатов разрушает команду. Структура Scrum команды представлена на рис. 3.

Команда в Scrum является многофункциональной, что особо важно при управлении командой. В команду входят участники, обладающие различными навыками: разработчики продукта, аналитики продукта, тестировщики готового продукта, каждый из участников команды выполняет определенные функции.

Модель разработки программного продукта по методологии Scrum изображен на рис. 4.

С учетом особенностей методологии Scrum можно представить модель реализации данного подхода для конкретного предприятия в сфере разработки программного обеспечения (рис. 5).

Внедрение методологии Scrum должно начинаться с создания команды разработчиков (Development Team).

В качестве основных функций Development Team можно выделить:

- разработка необходимого функционала для заказчика;
- отслеживание собственной результативностью (совместно со Scrum Master);
- принятие решений по разработке и дизайну продукта;
- оценивание элементов Product Backlog;
- ответственность Development Team за результат перед Product Owner [3].

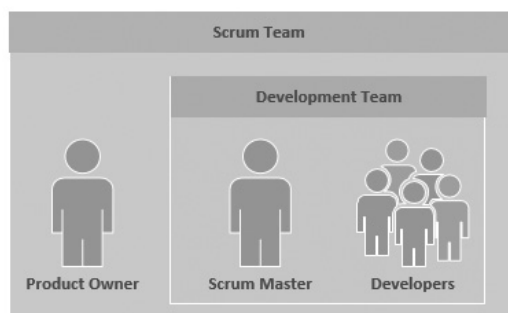


Рис. 3. Структура Scrum команды

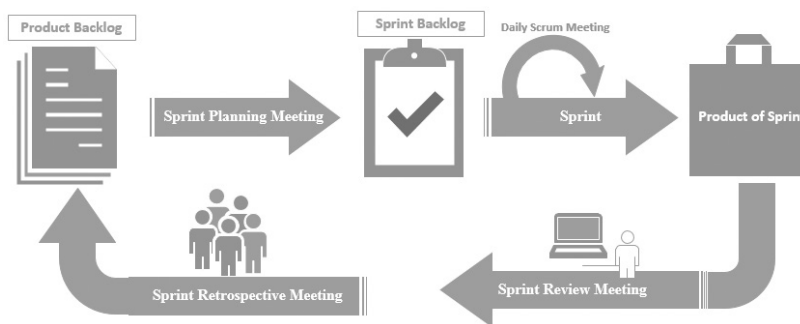


Рис. 4. Модель разработки программного продукта по методологии Scrum



Рис. 5. Модель Scrum для предприятия в сфере разработки

При формировании команды разработчиков важно отойти от привычного понимания должностей. Важно понимать, что каждый участник команды разработчиков должен быть наделен определенными качествами, так как каждый член команды играет определенную роль.

Методология Scrum требует особого внимания к организации единого рабочего пространства для Development Team. При внедрении методологии Scrum необходимо использовать практику открытого пространства. Единое пространство позволит участникам команды постоянно взаимодействовать между собой, представляя единый коммуникационный процесс, осуществляя полный цикл обратной связи.

На следующем этапе реализации методологии Scrum происходит назначение владельца продукта (product owner). Product Owner является владельцем продукта. Это не заказчик, а полноправный член команды, который имеет не только большую ответственность, но также и большие ограничения. Окончательные решения по разработке требуемого функционала принимает владелец продукта. Самая главная ответственность Product Owner — это создание и контроль Product Backlog. Все свои решения владелец продукта отражает в Product Backlog.

Основными задачами Product Owner при управлении Product Backlog являются:

- набор элементов Product Backlog;
- оптимальное расположение элементов для достижения цели;
- обеспечение доступности Product Backlog;
- формирование ценности работы Development Team;
- ответственность за Product Backlog [4].

Для того чтобы назначить Product Owner необходимо определить перечень личностных качеств, которыми он должен обладать.

Product Owner должен обладать высокой степенью вовлеченности и ответственности, должен принимать принципы Agile. Владелец продукта должен понимать суть продукта и его назначение [5].

Вопрос о том, должен ли Product Owner обладать навыками программирования достаточно спорный. Одни авторы утверждают, что владелец продукта должен быть выбран среди опытных разработчиков, другие отмечают, что Product Owner — это руководящая должность, а значит лучший вариант — это сотрудник из отрасли менеджмента.

Следующий этап внедрения методологии Scrum — это назначение Scrum Master, он не ставит задачи, а решает проблемы, возникающие в процессе работы команды.

В качестве основных функций Scrum Master можно обозначить:

- решение проблем, возникающих в команде;
- формирование благоприятного социально-психологического климата в команде;
- контроль заработной команды;
- изменение, в случае необходимости, статуса задач в спринте;
- реализация Daily Scrum Meeting;
- проведение встреч перед спринтами;
- организация помощи Product Owner с Backlog [6].

Следует отметить, что Scrum Master при этом должен взаимодействовать не только с командой разработчиков, но и с Product Owner. Он помогает понимать владельцу продукта, как можно создать Backlog с наивысшей ценностью. Scrum Master осуществляет поиск наиболее эффективных методов разработки и сопровождения Backlog.

Следующим этапом реализации Scrum является планирование спринта. Но данный процесс невозможно начать без двух составляющих Product Backlog и Sprint Backlog, которые в Scrum называются артефактами.

Бэклог продукта (product backlog) представляет собой определенный набор элементов с очередностью задач и перечнем необходимых функций, в которых заинтересован клиент [3]. Этот список содержит краткие описания всех желаемых возможностей продукта.

Product Owner должен обеспечить понимание Product Backlog. Прозрачность Product Backlog поможет разобраться в нем как команде, так и заказчику. Как правило, заказчик и специалистам сложно понять друг друга, так как говорят совершенно на разных профессиональных языках, что создает некий «языковой барьер», который оказывает отрицательное влияние на процесс работы над продуктом. Если где-то изначально возникло недопонимание и интерпретация желаний заказчика была не верна, то после спринта будет готовый продукт, который сильно ушел в сторону от желаний заказчика.

Элементы Product Backlog рассматриваются как user stories, они упорядочены в зависимости от их важности в проекте. Чем выше в бэклоге размещается элемент, тем быстрее разработчики приступят к работе с ним.

Под бэклог спринтом (Sprint Backlog) понимается список задач для функционирования бэклога продукта, которыми команда должна заняться в ближайший спринт. Необходимо обеспечить понимание участниками команды поставленных задач. Каждый спринт команда создает новый бэклог спринта на основе бэклога продукта. Планирование спринта (Sprint Planning Meeting) условно делится на две части, что отображено на рис. 6.

В первой части Product Owner проводит исследование элементов по Product Backlog и обсуждает их на встрече. Владелец продукта формулирует, что он хочет получить. Для Product Owner хорошим ориентиром будет обсуждение задач, которые будут распределены на спринты.

Целью спринта является формирование предложений, которые команда разработчиков планирует достичь по спринтам. Цель описывается совместно с командой и владельцем продукта.

Во второй части команда формирует Sprint Backlog. На данном этапе вмешательство Product Owner недопустимо. Будет проведена оценка продолжительности работы над той или иной задачей. Гибкость методологии Scrum необходима для длительности планирования спринта, которая зависит от длительности будущего спринта. Формула длительности планирования следующая: 1 неделя Sprint = 2 часа Sprint Planning Meeting [6].

После проведения планирования спринта задачи, которые вошли в Sprint Backlog, должны быть размещены на scrum-доске. Она является неотъемлемым артефактом методологии Scrum и служит центром информации о проекте и задачах на спринт.

Использование скрам-доски имеет следующие преимущества:

- процессы разработки становятся более прозрачными, так как участники видят статус каждой задачи

и соответственно могут своевременно влиять на процессы;

- появляется последовательность выполнения задач в рамках проекта, что позволяет обеспечивать стабильный уровень качества продукта на всех стадиях разработки ПО;
- использование онлайн-доски позволит руководству компании в любой момент узнать на какой стадии находится разработка проекта без отвлечения сотрудников от разработки программного обеспечения;
- ведется документация разработки продукта, которая должна храниться и распространяться централизованным образом.

Следующий этап после проведения планирования спринта — этап разработки требуемого функционала. В процессе Scrum Sprint каждый день следует проводить специальные встречи по 15 минут — Daily Scrum Meeting. Они нужны для того, чтобы открыто показать проблемы, возникающие в процессе работы команды, чтобы вовремя их устранить. Также эти встречи показывают, как идет работа по проекту в целом. Таким образом, можно сформулировать цель такого совещания следующим образом: скорректировать и понять работу команды, узнать, какие у нее текущие проблемы, и предложить варианты решения [7].

Участие в Daily Scrum Meeting обязательно для всей Development Team, также участвуют Scrum Master и Product Owner. Иные лица (заказчики, маркетологи и т.д.) могут присутствовать на встрече, однако не имеют права говорить.

Успешно реализовав все задачи из бэклога спринта, команда переходит к следующему этапу внедрения методологии скрам — демонстрации сделанного продукта (Sprint Reviews Meeting). Sprint Review Meeting должно проводиться в конце каждого спринта и несет обзорный характер. На встрече команда сможет оценить, что она сделала, чаще всего это будет представлено в виде демонстрации новых возможностей продукта [8].

В ходе Sprint Review Meeting проект соотносится с целью спринта, определенной во время планирования. Команда разработки программного обеспечения должна выполнить все задачи из Backlog Sprint, но это не является главным показателем продуктивности. Важно достигать цель спринта.

Завершающим этапом в ходе спринта является проведение ретроспективы (Sprint Retrospective Meeting), которая также должна проводиться в последний день спринта. Если Sprint Review Meeting ориентирован на результат продукта, то ретроспектива дает возможность видеть результат команды. Scrum-команда всегда должна иметь возможность

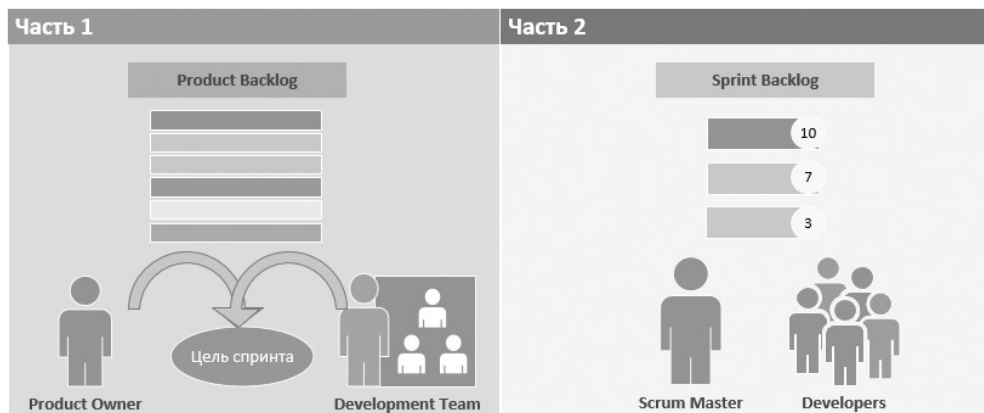


Рис. 6. Процесс планирования спринта

работы над совершенствованием, поэтому в методологии Scrum предусматривается время, позволяющее обдумать способы оптимизации работы над проектом.

ЗАКЛЮЧЕНИЕ

Применение гибкой методологии Scrum в командной работе принесет следующие преимущества:

- возможность краткосрочного планирования в виде бэклога спринта, который команда всегда обязуется выполнить в установленный срок, позволит повысить удовлетворенность заказчика;
- возможность лучшего понимания поставленных задач, что позволит сократить время работы с проблемами и возникающими ошибками. Как следствие, уменьшение объемов инцидентов со стороны технической поддержки;
- уменьшается время обслуживания клиента, так как будет минимизировано количество сбоев в программе, которые непосредственно влияют на время обслуживания клиента. Работа без спешки и неправильно поставленных сроков позволит проводить необходимое тестирование продукта, что положительным образом скажется на конечном потребителе;
- равномерность распределения нагрузки внутри команды разработчиков. Команды будут брать такое количество задач, которое смогут выполнить, а в выделенное дополнительное время смогут работать над дополнительными или срочными задачами;
- повышение эффективности коммуникаций между командой, руководством и заказчиком. Каждому члену команды необходимо понимать свою роль, это позволяет сделать планирование и распределение работ более прозрачными.

Следовательно, применение методологии Scrum позволит обеспечить системный и комплексный подход к разработке программного продукта.

Принимая во внимание тренд расширения инновационной активности компаний, направлениями дальнейших исследований могут стать вопросы персональных коммуникаций при решении задач по разработке программного обеспечения [9].

ЛИТЕРАТУРА

1. *Nissen V., Lezina T., Saltan A.* The Role of IT-Management in the Digital Transformation of Russian Companies. *Foresight and STI Governance*, 2018, vol. 12, no 3, pp. 53–61. DOI: 10.17323/2500-2597.2018.3.53.61
2. *Вольфсон Б.* Гибкие методологии разработки. СПб.: Питер, 2017.
3. *Schwaber K., Sutherland J.* The Scrum Guide. The definitive Guide to Scrum: The Rules of the Game, 2016.
4. *Cohn M.* Scaling Agile to Work with Distributed Teams. URL: <https://www.mountaingoatsoftware.com/presentations/scaling-agile-and-working-with-a-distributed-team> (дата обращения: 3 ноября 2023)
5. *Стеллман Э., Грин Д.* Постигая Agile. Ценности, принципы, методологии. М.: Манн, Иванов и Фербер, 2017.
6. *Деннинг С.* Эпоха Agile. М.: Манн, Иванов и Фербер, 2019.
7. *Сазерленд Д.* Scrum. Революционный метод управления проектами. М.: Манн, Иванов и Фербер, 2017.

8. *Копылова О.* Гибкие методологии для организации управления разработки программного продукта // *Вестник современных исследований*. 2019. № 1.3 (28). С. 90–91. (In Russian) URL: <https://elibrary.ru/item.asp?id=36885211> (дата обращения: 2 ноября 2023)
9. *Кудрявцева Е.И., Волкова Н.В.* Роль сетевой коммуникативной активности в формировании проектной команды // *Российский журнал менеджмента*. 2019. № 17 (1). С. 47–70. (In Russian). URL: <https://www.semanticscholar.org/paper> (дата обращения: 3 ноября 2023)
10. *Митрофанова А.Е.* Методические подходы к мотивации и стимулированию проектной команды / А.Е. Митрофанова, И.С. Брикошина // *Управление персоналом и интеллектуальными ресурсами в России*. — 2023. — Т. 12, № 4. — С. 54–61. (In Russian). DOI 10.12737/2305-7807-2023-12-4-54-61 (дата обращения: 3 ноября 2023)
11. *Митрофанова Е.А.* Особенности управления персоналом в командах / Е.А. Митрофанова, А.Е. Митрофанова // *Управление персоналом и интеллектуальными ресурсами в России*. — 2022. — Т. 11, № 4. — С. 10–15. (In Russian). DOI 10.12737/2305-7807-2022-11-4-10-15 (дата обращения: 3 ноября 2023)

REFERENCES

1. Nissen V., Lezina T., Saltan A. The Role of IT-Management in the Digital Transformation of Russian Companies. *Foresight and STI Governance*, 2018, vol. 12, no 3, pp. 53–61. DOI: 10.17323/2500-2597.2018.3.53.61
2. Wolfson B. Flexible development methodologies. St. Petersburg: Peter, 2017.
3. Schwaber K., Sutherland J. The Scrum Guide. The definitive Guide to Scrum: The Rules of the Game, 2016.
4. Cohn M. Scaling Agile to Work with Distributed Teams. URL: <https://www.mountaingoatsoftware.com/presentations/scaling-agile-and-working-with-a-distributed-team> (accessed November 3, 2023)
5. Stellman E., Green D. Understanding Agile. Values, principles, methodologies. M.: “Mann, Ivanov and Ferber”, 2017.
6. Denning S. The Age of Agile. M.: “Mann, Ivanov and Ferber”, 2019.
7. Sutherland D. Scrum. A revolutionary project management method. M.: “Mann, Ivanov and Ferber”, 2017.
8. Kopylova O. Flexible methodologies for organizing management of software product development. *Bulletin of modern research*. 2019. No. 1.3 (28). pp. 90–91. (In Russian) URL: <https://elibrary.ru/item.asp?id=36885211> (access date: November 2, 2023)
9. Kudryavtseva E.I., Volkova N.V. The role of network communication activity in the formation of a project team. *Russian Journal of Management*. 2019. No. 17 (1). pp. 47–70. (In Russian) URL: <https://www.semanticscholar.org/paper> (accessed November 3, 2023)
10. Mitrofanova A.E. Methodological approaches to motivating and stimulating a project team / A.E. Mitrofanova, I.S. Briko-shina // *Personnel and intellectual resources management in Russia*. — 2023. — Т. 12, No. 4. — P. 54–61. (In Russian) DOI 10.12737/2305-7807-2023-12-4-54-61 (accessed November 3, 2023)
11. Mitrofanova E.A. Features of personnel management in teams / E.A. Mitrofanova, A.E. Mitrofanova // *Personnel and intellectual resources management in Russia*. — 2022. — Т. 11, No. 4. — P. 10–15. (In Russian) DOI 10.12737/2305-7807-2022-11-4-10-15 (accessed November 3, 2023)