

Научная статья
УДК 004.413
doi: 10.17586/2713-1874-2025-3-59-68

МЕТОДЫ И АЛГОРИТМЫ УПРАВЛЕНИЯ ПРОЦЕССАМИ РАЗРАБОТКИ ПРОГРАММНЫХ ПРОДУКТОВ

*Петр Вениаминович Горшков^{1,4✉}, Игорь Александрович Бессмертный²,
Сергей Викторович Клименков³*

^{1,2,3}Университет ИТМО, Санкт-Петербург, Россия

⁴АО «Универсальная трейдинговая платформа»

^{1,4}vip@consultagency.ru ✉, <https://orcid.org/0009-0008-5173-7981>

²bessmertny@itmo.ru, <https://orcid.org/0000-0001-6711-6399>

³serge.klimenkov@cs.ifmo.ru, <https://orcid.org/0000-0001-5496-6765>

Язык статьи – русский

Аннотация: Нарастающая конкуренция среди разработчиков IT-проектов требует постоянно увеличивать скорость разработки. Это, в свою очередь, негативно сказывается на качестве и стоимости готового продукта. При этом снижение скорости и качества и повышение стоимости связаны не с низкой квалификацией отдельно взятого разработчика, а с плохой организацией процесса разработки. Предметом исследования являются наиболее популярные методологии и подходы и используемые в них методы и инструменты. Цель исследования заключается в систематизации знаний о современных методах и инструментах, определении их совместимости, оценки положительных и отрицательных сторон для понимания целесообразности по их комбинации, направленной на сокращение ограничений. В рамках исследования проведён обзор литературы для выявления методов и инструментов, используемых в наиболее популярных методологиях и подходах. Особое внимание уделено их применимости и ограничениям. Подготовлен и проведён опрос среди представителей отрасли с целью определения используемых на практике методов и инструментов. На основе анализа данных, полученных в процессе обзора литературных источников и опроса, были составлены рекомендации по выбору и использованию методов и инструментов для обеспечения применимости в большем количестве случаев и снижения ограничений.

Ключевые слова: автоматизация процессов, гибкие методологии, гибридные методологии, методологии разработки программного обеспечения, программная инженерия, традиционные методологии, Agile, DevOps, SCRUM

Ссылка для цитирования: Горшков П. В., Бессмертный И. А., Клименков С. В. Методы и алгоритмы управления процессами разработки программных продуктов // Экономика. Право. Инновации. 2025. Т. 13. № 3. С. 59–68. <http://dx.doi.org/10.17586/2713-1874-2025-3-59-68>.

METHODS AND ALGORITHMS FOR MANAGING SOFTWARE DEVELOPMENT PROCESSES

Petr V. Gorshkov^{1,4✉}, Igor A. Bessmertny², Sergey V. Klimenkov³

^{1,2,3}ITMO University, Saint Petersburg, Russia

⁴Universal Trading Platform JSC, Saint Petersburg, Russia

^{1,4}vip@consultagency.ru ✉, <https://orcid.org/0009-0008-5173-7981>

²bessmertny@itmo.ru, <https://orcid.org/0000-0001-6711-6399>

³serge.klimenkov@cs.ifmo.ru, <https://orcid.org/0000-0001-5496-6765>

Article in Russian

Abstract: The growing competition among developers of IT projects requires constantly increasing the speed of development. This, in turn, negatively affects the quality and cost of the finished product. At the same time, the decrease in speed and quality and the increase in cost are not related to the low qualifications of an individual developer, but to the poor organization of the development process. The subject of the research is the most popular methodologies and approaches and the methods and tools used in them. The purpose of the study is to systematize knowledge about modern methods and tools, determine their compatibility, evaluate the positive and negative sides in order to understand the feasibility of their combination aimed at reducing limitations. The study conducted a literature review to identify the methods and tools used in the most popular methodologies and approaches. Special attention is paid to their applicability and limitations. A survey was prepared and conducted among industry representatives in order to identify the methods

and tools used in practice. Based on the analysis of data obtained during the review of literature sources and the survey, recommendations were made on the selection and use of methods and tools to ensure applicability in more cases and reduce restrictions.

Keywords: Agile, agile methodologies, DevOps, hybrid methodologies, software development methodologies, process automation, SCRUM, software engineering, traditional methodologies

For citation: Gorshkov P. V., Bessmertny I. A., Klimentov S. V. Methods and Algorithms for Managing Software Development Processes. *Ekonomika. Pravo. Innovacii*. 2025. Vol. 13. No. 3. pp. 59–68. (In Russ.). <http://dx.doi.org/10.17586/2713-1874-2025-3-59-68>.

Введение. Существенная конкуренция среди разработчиков IT-проектов привела к необходимости увеличивать скорость разработки. Зачастую следствием этого становится снижение качества и увеличение стоимости. С целью минимизировать негативные последствия придумывается всё больше подходов к разработке программных продуктов, что провоцирует новые проблемы. Наличие большого количества различных подходов говорит об ограниченности отдельно взятого, что приводит к трудностям в принятии решения о выборе оптимального подхода руководителем и менеджерами проектов. Рекомендации не всегда могут решить эту проблему, так как проект может не подходить под рекомендуемые критерии, или, наоборот, будут подходить сразу несколько подходов.

Актуальность темы исследования обусловлена необходимостью повышать скорость разработки с сохранением качества и бюджета. Зачастую невозможность команд разработчиков соответствовать требованиям связана не с отсутствием необходимых компетенций у отдельно взятого члена команды, а с недостаточной организацией процессов в команде. Соответственно, правильный выбор подхода, методов и инструментов, позволяющий грамотно организовать рабочий процесс, является немаловажным фактором успеха разработки программного продукта.

Постановка проблемы и цели исследования. Исследовательская проблема заключается в выявлении методов и инструментов, используемых в современных подходах к управлению разработкой программных продуктов, для понимания их совместимости, применимости и наличия ограничений. Гипотеза, выдвигаемая в рамках исследования, основывается на предположении, что грамотная комбинация методов и инструментов может оказать положительное влияние на процесс

разработки и сократить ограничения для большинства IT-проектов.

Целью исследования является систематизация знаний о современных методах и инструментах, определение их совместимости, оценка положительных и отрицательных сторон и определение целесообразности дальнейшего исследования по их комбинации для сокращения ограничений. В рамках исследования решаются следующие задачи.

1) Анализ существующих подходов к управлению процессами разработки программных продуктов для выявления используемых методов и инструментов, их применимости и ограничений.

2) Исследование использования и комбинации тех или иных методов и инструментов в разработке IT-проектов на основе опроса.

3) Определение совместимости различных методов и инструментов на основе анализа.

Литературный обзор. Изучая различные источники, можно найти не одну сотню подходов, которые берут за основу традиционные или гибкие методы, а также их комбинации.

Согласно последним исследованиям, на данный момент среди традиционных методологий наиболее популярными являются Waterfall, Spiral, V-model [1] и RUP (Rational Unified Process) [2]. Хотя они все не новы, до сих пор активно используются. Это говорит о том, что предлагаемые ими практики в ряде случаев дают хороший результат и есть смысл их включить в исследование.

Но большинство IT-проектов реализуются с использованием гибких методологий. Наиболее популярными среди них являются SCRUM, XP (Extreme Programming), Kanban, Crystal, FDD (Feature-Driven Development), DSDM (Dynamic Systems Development Method) и TDD (Test-Driven Development) [3].

В таблице 1 представлены наиболее популярные подходы, их методы и инструменты, а также преимущества и ограничения.

Если один метод или инструмент встречается в нескольких методологиях, он не дублируется.

Таблица 1

Популярные подходы и методы управления процессами разработки программных продуктов

Источник: составлена авторами

Подходы	Методы и инструменты	Преимущества и ограничения
Waterfall	Каскадный жизненный цикл [4]	Эффективно для небольших и, наоборот, очень больших проектов с заранее определенными требованиями. Не подходит для больших проектов и если есть вероятность изменения требований.
Spiral	Спиральный жизненный цикл [1]	Простое и понятное управление, в результате каждого цикла выпускается новый функционал. Подходит для проектов, в которых на начальном этапе могут быть определены не все требования.
	Встроенный механизм управления рисками через механизм анализа и прототипирования [1]	Акцент на управлении рисками делает данный подход надежным. При этом повышаются затраты, что делает его неприемлемым для небольших проектов или если сильно ограничен бюджет.
V-model	Ранняя верификация и валидация через связь этапов проектирования и тестирования [5]	Позволяет выявлять проблемы на ранних этапах и повысить качество. Снижает применимость для краткосрочных проектов ввиду необходимости проверки на каждом этапе, повышает инвестиции на начальном этапе, так как сразу необходима группа тестирования.
Rational Unified Process (RUP)	Процесс разработки (итеративная модель с фазами и дисциплинами) [2]	Своевременная обратная связь, раннее выявление недочетов, предсказуемость и системность. Высокий порог входа и избыточность, перерасход ресурсов.
	Документирование и управление требованиями [2]	Прозрачность требований, поддержка сложных проектов и возможность внесения изменения. Высокая трудозатрата и снижение скорости.
SCRUM	Самоорганизующаяся скрам-команда [6]	Использование нестандартных ролей и самоорганизация. Стоит иметь в виду, что более привычная компоновка команды и иерархия так же может быть эффективна.
	Спринт и приращение продукта [7]	Итеративная разработка посредством спринтов, подчиняющихся строгим правилам. По окончании спринта должен быть представлен полностью рабочий функционал. Требуется достаточной квалификации разработчиков.
	Мероприятия – ритуалы (оценка пользовательских историй, ежедневные стендап-митинги, ретроспектива) [8]	Являются неотъемлемой частью методологии SCRUM, обеспечивая его функционирование. При недостаточной квалификации команды, отсутствии необходимости или формальном подходе могут занимать время, не принося пользы.

Продолжение таблицы 1

Подходы	Методы и инструменты	Преимущества и ограничения
	Бэклог продукта и диаграмма сгорания задач [7]	Пользовательские истории удобно писать заказчику. Для разработчика написанные заказчиком истории зачастую не несут всей необходимой информации.
Extreme Programming (XP)	Совместная разработка (парное программирование, общая собственность кода) [9]	Позволяет минимизировать ошибки и повысить скорость разработки. Может увеличить стоимость проекта, так как подразумеваются пары программистов одного уровня.
	Простота и малые итерации [9]	Позволяет достичь максимальной скорости разработки и оперативной обратной связи. Высокий риск проблем с архитектурой.
Kanban	Визуализация рабочего процесса [10]	Прозрачность, наглядность и понимание процесса. Не идеальна для долгосрочного планирования, требует зрелости команды.
	Управление потоком и ограничение незавершённой работы [11]	Распределение нагрузки и минимизация простоев. Требует дисциплинированности и могут быть трудности при недостаточной стабильности процессов.
Crystal	Инкрементальная разработка с фокусом на коммуникации [9]	Легкий старт для небольших проектов и быстрая обратная связь. Требует определённых навыков, не эффективна при распределённой команде.
	Приспособление процесса под размер и критичность проекта (цветовой код) [9]	Возможность выбрать используемые артефакты исходя из характеристик проекта. Необходимо обладать достаточными компетенциями.
Feature-Driven Development (FDD)	Разбиение проекта на функции с последующей реализацией через мини-водопад [9]	Прозрачность процесса, регулярные релизы и контроль качества на уровне функций. Неэффективно для небольших проектов, могут быть проблемы с декомпозицией и архитектурой.
Dynamic Systems Development Method (DSDM)	Корректирование объёмов функциональности исходя из ресурсов и времени [12]	Хороший контроль над сроками и бюджетом, высокая адаптивность и учет бизнес-ценностей. Требует постоянного взаимодействия с заказчиком, может усложнять управление зависимостями.
Test-Driven Development (TDD)	Разработка через тестирование [12]	Хорошее качество кода с ранним выявлением ошибок и минимумом дефектов. Проблемы с документацией и необходимы определённые навыки.

Так же можно добавить такие подходы как Behavior-Driven Development (BDD) – понятный бизнес-язык тестов, Disciplined Agile Delivery (DAD) – DevOps и масштабируемость, Lean Software Development (LSD) – фокус на поток создания ценности, Open Source Software Development (OSSD) – открытая разработка [13].

Есть и другие подходы, например, Microsoft Solutions Framework (MSF), Rapid Application Development (RAD), Agile Project Management (APM), Agile Unified Process (AUP), Open Unified Process (OpenUP), Personal Software Process & Team Software Process (PSP&TSP), Adaptive Software Development (ASD), ICONIX Process (ICONIX), Agile

Modeling (AM), Architecture-Driven Modernization (ADM), Participatory Process (PP), Information Systems Development (ISD), Adaptive Software Process (ASP) и Design-Driven Development (D3) [13], но они в меньшей степени

предлагают принципиально новые методы. Отдельное внимание стоит уделить DevOps, так как он содержит в себе необходимые для современной разработки методы и инструменты (таблица 2).

Таблица 2

Методы DevOps

Источник: составлено авторами

Методы и инструменты	Описание
CI/CD (Continuous Integration / Continuous Deployment) [14]	Автоматизация сборки, тестирования и развертывания.
Автоматизация тестирования [15]	Используются различные инструменты, и специализированные DevOps-тестовые фреймворки для обеспечения качества.
Контейнеризация и оркестрация [16]	Инструменты для упаковки приложений и управления их развертыванием в распределённой среде.
Мониторинг и логирование [16]	Инструменты для мониторинга и анализа состояния системы и приложений.
Управление конфигурациями и инфраструктурой как код [16]	Позволяет автоматизировать управление инфраструктурой и развертыванием.
Сентимент-анализ и метрики [15]	Внедрение инструментов для анализа тональности коммуникаций, KPI, метрик зрелости и прогресса проекта.
Dashboards и отчёты [15]	Различные BI-панели и дашборды, интеграция с DevOps pipeline для визуализации прогресса, качества и зрелости проекта.

Компоновка методов и инструментов в рамках отдельно взятого подхода позволяет сделать его более адаптированным под конкретные ситуации, но снижает универсальность, что приводит к такому большому количеству различных подходов «на все случаи жизни» и провоцирует связанные с этим трудности. Различные надстройки, как бы делающие подход более универсальным, зачастую его усложняют, но не дают должного результата.

Материалы и методы. Для достижения поставленной цели и решения задач был проведен обзор литературы для выявления методов и инструментов, используемых в наиболее популярных подходах к управлению IT-проектами. Источники подбирались по ключевым словам и запросам в базах данных Scopus, Web of Science и Google Scholar исходя из авторитетности издания,

признанности автора, актуальности ссылок, содержания статьи и даты публикации.

Далее был подготовлен и проведён опрос, чтобы понять, какие методы и инструменты используются, как они между собой комбинируются, и к каким это приводит результатам. Опрос состоял из 13 вопросов с готовыми вариантами ответов, а также текстовым полем, куда можно было внести свои варианты или добавить комментарии. В опросе приняли участие 102 человека, среди которых – 64 разработчика, 8 тестировщиков, 7 руководителей проекта, 5 DevOps-инженеров, по 4 архитектора и аналитика и другие, с опытом работы 1–3 года – 27 человек, 3–5 лет – 36 человек, 5–10 лет – 31 человек, более 10 лет – 2 человека и менее 1 года – 6 человек. Опрос включал в себя следующие вопросы:

- 1) Какова Ваша роль в IT-проекте?
- 2) Какой у Вас опыт работы в сфере IT?

- 3) Какую методологию разработки Вы используете в качестве основной в проектах?
- 4) Насколько полно Вы реализуете выбранную основную методологию в проектах?
- 5) Какие другие методологии Вы используете в проектах дополнительно к основной?
- 6) Какие методы разработки Вы применяете в своих проектах?
- 7) Какие инструменты DevOps Вы используете в своих проектах?
- 8) На Ваш взгляд, чего не хватает в используемых Вами методологиях для повышения эффективности разработки?
- 9) Какие методы или инструменты, на Ваш взгляд, являются избыточными или усложняющими процесс разработки?
- 10) Как часто Вы совмещаете методы и инструменты из разных методологий в одном проекте?
- 11) Насколько Вы удовлетворены процессом разработки в своих проектах?

12) Насколько Вы удовлетворены качеством и результатами своих проектов?

13) Какие основные проблемы Вы видите при использовании методологий разработки в своих проектах?

На основе анализа данных, полученных из литературного обзора и опроса, была определена совместимость различных методов и инструментов.

Результаты исследования. На основе анализа выбранных источников были выявлены методы и инструменты, используемые в наиболее популярных подходах к разработке программных продуктов. Далее был проведен опрос среди представителей IT-отрасли. Наиболее популярные подходы, используемые в качестве основного, представлены на рисунке 1, полнота реализации выбранного основного подхода – на рисунке 2.

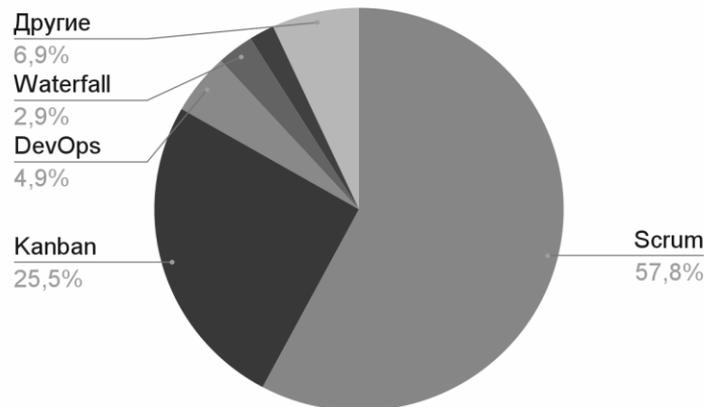


Рисунок 1 – Основной подход
Источник: составлен авторами

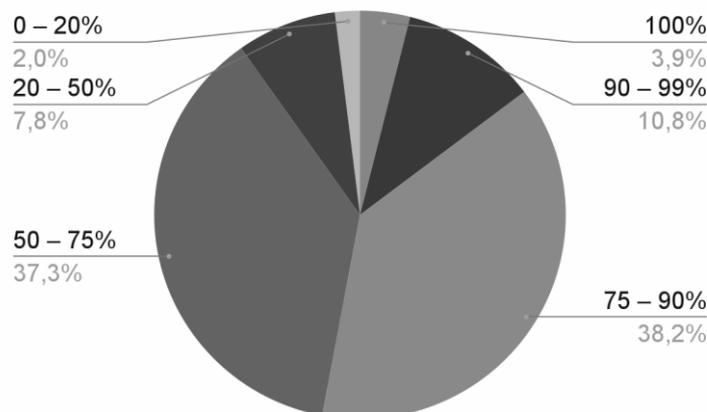


Рисунок 2 – Полнота реализации основного подхода
Источник: составлен авторами

При этом 47% респондентов отметили, что совмещают методы и инструменты из разных подходов в одном проекте более, чем в 50% случаев, 26% отметили, что во всех проектах, 15% – иногда (реже, чем в 25% случаев) и 12% – редко или никогда.

В дополнение к основному подходу чаще всего берут элементы из подходов, перечисленных на рисунке 3. При этом в своих проектах респонденты наиболее часто используют методы и инструменты, указанные на рисунке 4.

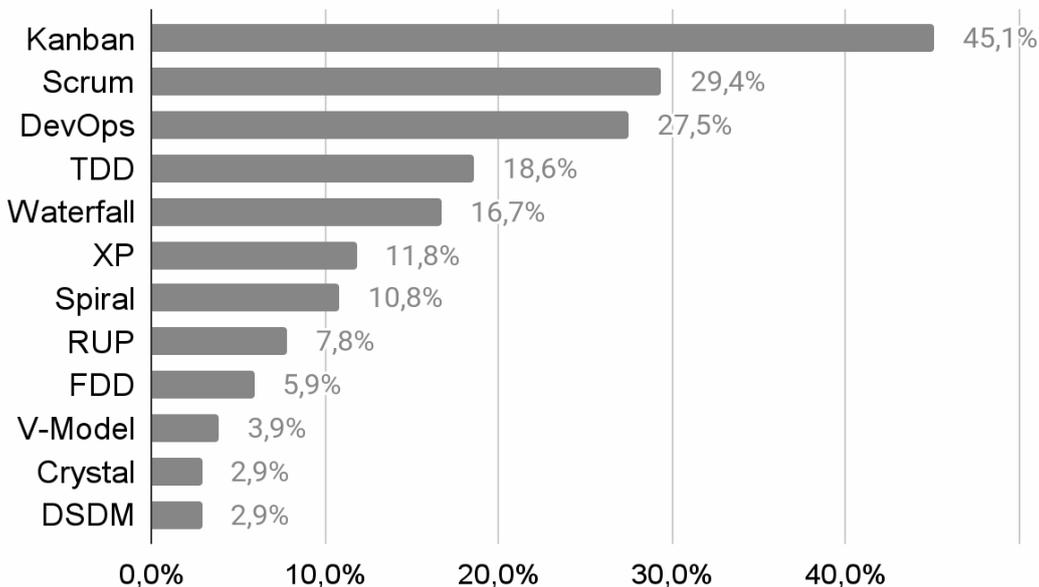


Рисунок 3 – Дополнительные подходы

Источник: составлен авторами



Рисунок 4 – Методы и инструменты

Источник: составлен авторами

Различные DevOps инструменты используются более чем 78% команд разработчиков, наиболее популярные из которых – Docker (78%) и GitLab CI (73%). Инструменты для мониторинга, логирования и отчетности используются минимум в 62% случаев.

Обсуждение. Хотя есть множество различных гибких подходов, 7% респондентов отметили, что за основу в их командах используются традиционные методологии, а элементы традиционных методологий используют более 18% респондентов. Это говорит о том, что классические подходы по-прежнему остаются актуальными.

Большинство команд разработчиков используют гибкие подходы, при этом все равно отмечаются сложные механизмы контроля качества (35%), сложные процессы управления требованиями (28%), избыточное документирование (23%), чрезмерно детализированные тестовые сценарии (17%) и жесткая архитектурная модель (14%), что противоречит основным принципам Agile. А некоторые респонденты, наоборот, жалуются на избыточную гибкость (23%), автоматизацию (9%), и «ритуалы ради ритуалов».

Хотя больше половины респондентов отметили, что используют различные инструменты для мониторинга, логирования и отчетности, отвечая на вопрос, что является избыточным и мешает разработке, 25% отметили, что слишком много Dashboards и отчетов.

Опрос показал, что в большинстве случаев используются гибридные подходы, при этом далеко не всегда из каждого отдельно взятого подхода берется только то, что способствует лучшему опыту, подтягиваются так же и ограничения. В то же время попытки строго следовать определенному выбранному подходу приводят к излишнему вниманию на внешних атрибутах, которые зачастую мешают разработке.

Выводы и рекомендации. На основании проведенного исследования сделан вывод о том, что в реальной практике практически не встречается методологий в чистом виде, и в большинстве случаев каждая команда подстраивает под себя одну из методологий (почти всегда это одна из наиболее популярных

методологий, самой популярной является Scrum), добавляя туда элементы другой (других) методологии (чаще всего Kanban-доску).

Согласно результатам опроса нет особых ограничений между комбинациями различных методов и практик, есть различные примеры комбинаций, при этом удовлетворенность процессом разработки, а также качеством и результатами в большинстве случаев (более 61%) оценена на 4 балла из 5. Некоторые методы и инструменты (например, CI/CD, контейнеризация, парное программирование, автоматическое тестирование) стали фактически универсальными и применяются вне зависимости от используемой методологии.

При комбинации методов и практик из разных подходов есть смысл адаптировать их к основной используемой методологии, чтобы не добавлять сопутствующие ограничения. Также, особенно при использовании гибких подходов, необходимо определить, какие артефакты и ритуалы будут способствовать повышению эффективности разработки, а какие, наоборот, будут занимать время, особо не принося пользы. Рекомендуется убрать излишнюю документацию, перегруженные процессы согласования требований и дублирующие отчеты.

Исходя из всего вышесказанного можно сформировать следующие рекомендации.

1) Выбираемый подход необходимо адаптировать под проект.

2) Сокращать ненужную бюрократию, оставляя (или добавляя) необходимую документацию.

3) Усиливать слабые места методами и инструментами из других подходов, но стараться не добавлять сопутствующие ограничения.

4) Осуществлять выбор, опираясь на вводные проекта, так как не всегда популярность метода говорит о его пользе в конкретной ситуации.

Дальнейшее исследование будет посвящено определению комбинации конкретных методов и практик в рамках одного подхода для минимизации ограничений и повышению применимости в различных ситуациях.

Список источников

1. Akinsola J.E.T., Ogunbanwo A.S., Okesola O.J., Odun-Ayo I.J., Ayegbusi F.D., Adebisi A.A. Comparative Analysis of Software Development Life Cycle Models (SDLC) // *Advances in Intelligent Systems and Computing*, Springer, Cham. – 2020. – Т. 1224. – С. 310–322. – DOI: 10.1007/978-3-030-51965-0_27. – Текст: электронный. (In Eng.).
2. Shafiee S., Wautelet Y., Hvam L., Sandrin E., Forza C. Scrum Versus Rational Unified Process in Facing the Main Challenges of Product Configuration Systems Development // *Journal of Systems and Software*. – 2020. – Т. 170. – DOI: 10.1016/j.jss.2020.110732. – Текст: электронный. (In Eng.).
3. Ibrahim M., Aftab S., Bakhtawar B., Ahmad M., Iqbal A., Aziz N., Javeid M., Ihnaini B. Exploring the Agile Family: A Survey // *International Journal of Computer Science and Network Security*. – 2020. – Т. 20 (10). – С. 163–179. – DOI: 10.22937/IJCSNS.2020.20.10.22. – Текст: электронный. (In Eng.).
4. Saravanos A., Curinga M. Simulating the Software Development Lifecycle: The Waterfall Model // *Applied System Innovation*. – 2023. – Т. 6 (6). – С. 108. – DOI: 10.3390/asi6060108. – Текст: электронный. (In Eng.).
5. Graessler I., Hentze J. The New V-Model of VDI 2206 and its Validation // *Automatisierungstechnik*. – 2020. – Т. 68 (5). – С. 312–324. – DOI: 10.1515/auto-2020-0015. – Текст: электронный. (In Eng.).
6. Žužek T., Kušar J., Rihar L., Berlec T. Agile-Concurrent hybrid: A Framework for Concurrent Product Development Using Scrum // *Concurrent Engineering: Research and Applications*. – 2020. – Т. 28 (1). – С. 1–10. – DOI: 10.1177/1063293X20958541. – Текст: электронный. (In Eng.).
7. Grebic B., Stojanović A. Application of the Scrum Framework on Projects in IT Sector // *European Project Management Journal*. – 2021. – Т. 11 (2). – С. 37–46. – DOI: 10.18485/epmj.2021.11.2.4. – Текст: электронный. (In Eng.).
8. Morandini M., Coletti T.A., Oliveira E., Corrêa P.L.P. Considerations About the Efficiency and Sufficiency of the Utilization of the Scrum Methodology: A Survey for Analyzing Results for Development Teams // *Computer Science Review*. – 2021. – Т. 39. – DOI: 10.1016/j.cosrev.2020.100314. – Текст: электронный. (In Eng.).
9. Saeedi K., Visvizi A. Software Development Methodologies, HEIs, and the Digital Economy // *Educ. Sci.* – 2021. – Т. 11 (2). – С. 73. – DOI: 10.3390/educsci11020073. – Текст: электронный. (In Eng.).
10. Orlov E. V., Rogulenko T. M., Smolyakov O. M., Oshovskaya N. V., Zvorykina T. I., Rostanets V. G., Dyundik E. P. Comparative Analysis of the Use of Kanban and Scrum Methodologies in IT Projects // *Universal Journal of Accounting and Finance*. –

References

1. Akinsola J.E.T., Ogunbanwo A.S., Okesola O.J., Odun-Ayo I.J., Ayegbusi F.D., Adebisi A.A. Comparative Analysis of Software Development Life Cycle Models (SDLC). *Advances in Intelligent Systems and Computing*, Springer, Cham. 2020. Vol. 1224. pp. 310–322. DOI: 10.1007/978-3-030-51965-0_27.
2. Shafiee S., Wautelet Y., Hvam L., Sandrin E., Forza C. Scrum Versus Rational Unified Process in Facing the Main Challenges of Product Configuration Systems Development. *Journal of Systems and Software*. 2020. Vol. 170. DOI: 10.1016/j.jss.2020.110732.
3. Ibrahim M., Aftab S., Bakhtawar B., Ahmad M., Iqbal A., Aziz N., Javeid M., Ihnaini B. Exploring the Agile Family: A Survey. *International Journal of Computer Science and Network Security*. 2020. Vol. 20 (10). pp. 163–179. DOI: 10.22937/IJCSNS.2020.20.10.22.
4. Saravanos A., Curinga M. Simulating the Software Development Lifecycle: The Waterfall Model. *Applied System Innovation*. 2023. Vol. 6 (6). P. 108. DOI: 10.3390/asi6060108.
5. Graessler I., Hentze J. The New V-Model of VDI 2206 and its Validation. *Automatisierungstechnik*. 2020. Vol. 68 (5). pp. 312–324. DOI: 10.1515/auto-2020-0015.
6. Žužek T., Kušar J., Rihar L., Berlec T. Agile-Concurrent hybrid: A Framework for Concurrent Product Development Using Scrum. *Concurrent Engineering: Research and Applications*. 2020. Vol. 28 (1). pp. 1–10. DOI: 10.1177/1063293X20958541.
7. Grebic B., Stojanović A. Application of the Scrum Framework on Projects in IT Sector. *European Project Management Journal*. 2021. Vol. 11 (2). pp. 37–46. DOI: 10.18485/epmj.2021.11.2.4.
8. Morandini M., Coletti T.A., Oliveira E., Corrêa P.L.P. Considerations About the Efficiency and Sufficiency of the Utilization of the Scrum Methodology: A Survey for Analyzing Results for Development Teams. *Computer Science Review*. 2021. Vol. 39. DOI: 10.1016/j.cosrev.2020.100314.
9. Saeedi K., Visvizi A. Software Development Methodologies, HEIs, and the Digital Economy. *Educ. Sci.* 2021. Vol. 11 (2). P. 73. DOI: 10.3390/educsci11020073.
10. Orlov E. V., Rogulenko T. M., Smolyakov O. M., Oshovskaya N. V., Zvorykina T. I., Rostanets V. G., Dyundik E. P. Comparative Analysis of the Use of Kanban and Scrum Methodologies in IT Projects. *Universal Journal of Accounting and Finance*. 2021.

2021. – Т. 9 (4). – С. 693–700. – DOI: 10.13189/ujaf.2021.090415. – Текст: электронный. (In Eng.).
11. Hamzah A., Mazni O., Rohaida R. The State of the Art of Agile Kanban Method: Challenges and Opportunities // *Independent Journal of Management & Production*. – 2021. – Т. 12 (8). – С. 2535–2550. – DOI: 10.14807/ijmp.v12i8.1482. – Текст: электронный. (In Eng.).
12. Alsaqqa S., Sawalha S., Abdel-Nabi H. Agile Software Development: Methodologies and Trends // *International Journal of Interactive Mobile Technologies*. – 2020. – Т. 14 (11). – С. 246–270. – DOI: 10.3991/ijim.v14i11.13269. – Текст: электронный. (In Eng.).
13. Necmettin O., Busra O.K., Mehmet G. Towards a Better Understanding of Agile Mindset by Using Principles of Agile Methods // *Proceedings of the Federated Conference on Computer Science and Information Systems*. – 2020. – Т. 21. – С. 721–730. – DOI: 10.15439/2020F46. – Текст: электронный. (In Eng.).
14. Jayakody J.A.V.M.K., Wijayanayake J. Challenges for Adopting Devops in Information Technology Projects // *International Research Conference on Smart Computing and Systems Engineering (SCSE), Colombo, Sri Lanka*. – 2021. – С. 203–210. – DOI: 10.1109/SCSE53661.2021.9568348. – Текст: электронный. (In Eng.).
15. Angara J., Prasad S., Gutta S. DevOps Project Management Tools for Sprint Planning, Estimation and Execution Maturity // *Cybernetics and Information Technologies*. – 2020. – Т. 20 (2). – С. 79–92. – DOI: 10.2478/cait-2020-0018. – Текст: электронный. (In Eng.).
16. Akbar M.A., Khan A.A., Islam N., Mahmood S. DevOps Project Management Success Factors: A Decision-Making Framework // *Softw: Pract Exper*. 2024. – Т. 54 (2). – С. 257–280. – DOI: 10.1002/spe.3269. – Текст: электронный. (In Eng.).
- Vol. 9 (4). pp. 693–700. DOI: 10.13189/ujaf.2021.090415.
11. Hamzah A., Mazni O., Rohaida R. The State of the Art of Agile Kanban Method: Challenges and Opportunities. *Independent Journal of Management & Production*. 2021. Vol. 12 (8). pp. 2535–2550. DOI: 10.14807/ijmp.v12i8.1482.
12. Alsaqqa S., Sawalha S., Abdel-Nabi H. Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies*. 2020. Vol. 14(11). pp. 246–270. DOI: 10.3991/ijim.v14i11.13269.
13. Necmettin O., Busra O.K., Mehmet G. Towards a Better Understanding of Agile Mindset by Using Principles of Agile Methods. *Proceedings of the Federated Conference on Computer Science and Information Systems*. 2020. Vol. 21, pp. 721–730. DOI: 10.15439/2020F46.
14. Jayakody J.A.V.M.K., Wijayanayake J. Challenges for Adopting Devops in Information Technology Projects. *International Research Conference on Smart Computing and Systems Engineering (SCSE), Colombo, Sri Lanka*. 2021. pp. 203–210. DOI: 10.1109/SCSE53661.2021.9568348.
15. Angara J., Prasad S., Gutta S. DevOps Project Management Tools for Sprint Planning, Estimation and Execution Maturity. *Cybernetics and Information Technologies*. 2020. Vol. 20 (2). pp. 79–92. DOI: 10.2478/cait-2020-0018.
16. Akbar M.A., Khan A.A., Islam N., Mahmood S. DevOps Project Management Success Factors: A Decision-Making Framework. *Softw: Pract Exper*. 2024. Vol. 54 (2). pp. 257–280. DOI: 10.1002/spe.3269.