

Научная статья
УДК 004.413
doi: 10.17586/2713-1874-2024-4-40-49

СРАВНЕНИЕ МЕТОДОВ УПРАВЛЕНИЯ ИТ-ПРОЕКТАМИ

Петр Вениаминович Горшков^{1✉}, Игорь Александрович Бессмертный²

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

¹АО «Универсальная трейдинговая платформа», Санкт-Петербург, Россия

¹vip@consultagency.ru[✉], <http://orcid.org/0009-0008-5173-7981>

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

Язык статьи – русский

Аннотация: В условиях быстрого развития технологий и растущих требований к разработке программного обеспечения традиционные методологии управления проектами все чаще уступают место гибким и гибридным подходам. В статье анализируются современные методы управления ИТ-проектами и их влияние на успех, исходя из характеристик проекта, ключевых требований и особенностей команды. Рассмотрены сильные и слабые стороны традиционных, гибких и гибридных методологий. А также виды, для которых их использование будет наиболее приемлемо. Особое внимание уделено ограничениям при использовании той или иной методологии для возможности дальнейшей компенсации за счет объединения различных методов и подходов. Актуальность данного исследования обусловлена увеличивающимся разрывом между постоянно возрастающими требованиями к разработке программного обеспечения и ограниченными ресурсами и возможностями команд разработчиков. Во многих ситуациях причиной этого несоответствия является не отсутствие профессиональных навыков у программистов или их низкая квалификация, а недостаточно эффективная организации рабочих процессов внутри команд. В работе анализируются факторы, влияющие на выбор методологии, и даются рекомендации по их использованию для различных типов проектов.

Ключевые слова: методологии разработки программного обеспечения, программная инженерия, управление ИТ-проектами, Agile, DevOps, SCRUM

Ссылка для цитирования: Горшков П. В., Бессмертный И. А. Сравнение методов управления ИТ-проектами // Экономика. Право. Инновации. 2024. № 4. С. 40–49. <http://dx.doi.org/10.17586/2713-1874-2024-4-40-49>.

COMPARISON OF IT PROJECT MANAGEMENT METHODS

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

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

¹JSC Universal Trading Platform, Saint Petersburg, Russia

¹vip@consultagency.ru[✉], <http://orcid.org/0009-0008-5173-7981>

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

Article in Russian

Abstract: With the rapid development of technologies and growing demands on software development, traditional project management methodologies are increasingly giving way to agile and hybrid approaches. The article analyzes modern methods of IT project management and their impact on successful project implementation based on project characteristics, key requirements and team features. The strengths and weaknesses of traditional, agile and hybrid methodologies and project types are considered. Much attention is paid to methodology limitations and their compensation by combining different methods and approaches. The study is relevant due to the widening gap between the ever-increasing requirements for software development and the limited resources and capabilities of development teams. In many situations, the reasons for this discrepancy are not the lack of professional skills of programmers or their low qualifications, but the insufficiently effective organization of work processes within teams. The paper analyses the factors influencing the choice of methodology and in recommendations for their optimal use for various types of projects.

Keywords: Agile, DevOps, IT project management, SCRUM, software development methodologies, software engineering

For citation: Gorshkov P. V., Bessmertny I. A. Comparison of IT Project Management Methods. *Ekonomika. Pravo. Innovacii*. 2024. No. 4. pp. 40–49. (In Russ.). <http://dx.doi.org/10.17586/2713-1874-2024-4-40-49>.

Введение. В последнее время наблюдается обострение конкуренции разработчиков программного обеспечения, которая вынуждает их резко сокращать сроки разработки в ущерб качеству с целью как можно быстрее выйти на рынок, постепенно устраняя недостатки, используя механизмы обратной связи с пользователями.

Таким образом, традиционные методологии управления проектами всё чаще уступают место гибким методологиям, но и они имеют свои ограничения. Для устранения недостатков гибких методологий всё больше команд разработчиков прибегают к использованию гибридных подходов, которые также имеют свои преимущества и недостатки.

Актуальность темы исследования обусловлена противоречием между требованием ускорения разработки программного обеспечения и ограниченными возможностями коллективов разработчиков. Во многих случаях этот разрыв связан не с недостаточной квалификацией программистов, а с низким уровнем организации процессов в команде. Таким образом, правильный выбор методологии управления и грамотная организация рабочего процесса являются одними из ключевых факторов успеха IT-проекта.

Постановка проблемы и цели исследования. Исследовательская проблема заключается в определении и сравнении современных методологий управления IT-проектами для понимания их применимости и ограничений. Гипотеза, выдвигаемая в рамках исследования, основывается на предположении, что грамотная комбинация методов из различных подходов может существенно сократить ограничения для большинства IT-проектов.

Целью исследования является систематизация знаний о современных методах управления IT-проектами, оценка их положительных и отрицательных сторон и разработка рекомендаций по их комбинации для сокращения ограничений. В рамках исследования решаются следующие задачи.

1) Анализ существующих методологий управления IT-проектами на основе публикаций.

2) Выявление факторов, влияющих на эффективность, и разработка рекомендаций

по комбинации методов для сокращения ограничений.

Литературный обзор. В настоящий момент существуют сотни подходов, которые состоят из традиционных и гибких методов и практик. Стоит отметить, что до сих пор остаются вопросы относительно того, что такое «Agile», и лишь немногие проекты придерживаются чистого «Agile», предпочитая собственные гибридные методы разработки [1]. Хотя многие компании используют какую-то конкретную методологию, преимущественно гибкую, более 50% проектов можно классифицировать как гибридные [2]. Среди традиционных методологий наиболее часто встречаются Waterfall, Spiral, V-model, Rational Unified Process (RUP), Incremental/Iterative, Synchronize and Stabilize, Rapid Prototyping, OpenUP (Open Unified Process), MSF (Microsoft Solutions Framework), PMBOK (Project Management Body of Knowledge) и PRINCE2 (Projects in Controlled Environments Version 2), каждая из которых имеет свои плюсы и минусы [3–5]. Если взять, например, V-model, существует большое количество вариаций, но они несовершенны, и их необходимо дорабатывать [6]. Среди гибких и гибридных методологий чаще всего используются Scrum, XP (Extreme Programming), Kanban, Crystal, FDD (Feature-Driven Development), DSDM (Dynamic Systems Development Method), TDD (Test-Driven Development), ASD (Adaptive Software Development), D3 (Design-Driven Development), Scrumban, AUP (Agile Unified Process), BDD (Behavior-Driven Development) и DevOps [4].

Хотя в последнее время наблюдается тренд на переход к гибким методологиям даже в компаниях, не связанных с разработкой [7], и гибкие методологии становятся общепризнанным стандартом [8], остаются случаи, когда традиционные методологии более эффективны, например:

- при невозможности разделения проекта на небольшие части;
- когда регулярные изменения или поэтапный подход невозможны с юридической, технической или экономической точки зрения;
- если операционные риски не позволяют использовать гибкий подход;

- при создании решения в области управления процессами или приложений реального времени и систем, критически важных для безопасности [9].

Более 70% организаций за основу выбирают гибкие методологии из-за сложностей при внесении изменений, исправлений и корректировок по мере развертывания проекта и его фаз традиционными методами [10].

Традиционные методы, наиболее популярные из которых представлены в таблице 1, являются наилучшими в стабильной среде, когда определённый результат должен быть представлен за фиксированный бюджет в обозначенные сроки [11] и предъявляются особые требования к безопасности, например ВПК, космическая отрасль и другие [12].

Таблица 1

Область применения и ограничения наиболее популярных традиционных методологий

Источник: составлена авторами на основе [3, 13, 14]

Методология	Область применения	Ограничения
Waterfall	Непродолжительные проекты с постоянными и четкими требованиями, стабильной технологией, подразумевающих высокий контроль качества	Ограниченная гибкость, недостаточное участие клиентов и отсутствие своевременной обратной связи, сложность изменения требований после начала разработки, позднее тестирование (высокий риск выявления ошибок на поздних стадиях), длительный цикл разработки из-за возможных задержек на отдельных этапах [13]
Spiral	Сложные проекты со средним и высоким уровнем риска, где требуется оценка затрат, возможны корректировки, неопределённые требования заказчика	Слишком тяжелое управление и необоснованные затраты для простых проектов, высокая зависимость от экспертной оценки, излишняя документация, сложности соблюдения бюджета и графика [3]
V-model	Небольшие и средние проекты с определенными спецификациями и доступом к необходимым технологиям	Высокие изначальные затраты (поскольку тестирование начинается с первого этапа), значительная доработка документации (благодаря изменению требований), отсутствие гибкости [3]
Rational Unified Process (RUP)	Крупные проекты с высокой сложностью и необходимостью в формальном управлении, стабильности и точной документации	Риск превышения бюджета и отклонения от графика (особенно при изменении требований), большое количество документации (усложняет проект и снижает гибкость), ограниченная коммуникация между участниками (приводит к изоляции команд) [14]

Традиционные методы подразумевают трудоёмкий процесс, включающий в себя составление плана проекта, подробное описание требований до начала разработки и

дальнейшее следование им на этапах проектирования, кодирования и тестирования, практически без возможности вносить изменения [15]. Также традиционные подходы

зачастую обречены на провал при реализации инновационных проектов, так как подробное предварительное планирование оказывает только негативное влияние, а недостаточная автономия и коммуникация не обеспечивают

необходимой для получения результата вовлечённости [16]. Всё это способствует переходу к более гибким методологиям, наиболее популярные из которых представлены в таблице 2.

Таблица 2

Область применения и ограничения наиболее популярных гибких методологии

Источник: составлена авторами на основе [17–19]

Методология	Область применения	Ограничения
SCRUM	Небольшие или средние команды, проекты с меняющимися требованиями и приоритетами, где важна гибкость и быстрая адаптация	В чистом виде не подходит для крупных или распределенных команд, слабая системная архитектура, отсутствие гибкости в спринтах, упор на планировании и выполнении спринтов, а не на качестве продукта, требуются опытные члены команды, недостаточно внимания уделяется инженерии и тестированию [17]
Extreme Programming (XP)	Малые проекты с низким уровнем риска, где требуется гибкость, нет необходимости в ведении документации и приветствуется быстрое реагирование на изменения в требованиях	Минимум документации и недостаточное внимание к архитектуре и дизайну ограничивают масштабируемость XP, делая его подходящим только для совсем небольших проектов [18]
Kanban	Проекты, где важно последовательное улучшение и управление рабочей нагрузкой, с акцентом на гибкость и адаптацию к существующим процессам	Сложно установить лимиты незавершенной работы, недостаточная визуализация процесса может усложнить мониторинг проекта, трудности с внедрением из-за привязанности сотрудников к привычным методам, перебои в процессе могут замедлить весь поток работы, требуя дополнительных усилий на восстановление [19]
Crystal	Небольшие, локальные команды с проектами, ограниченными по времени и ресурсам, если важна гибкость и адаптивность	Отсутствие системной верификации дизайна и кода, недостаточная документация и ограниченное описание практик, зависимость плана проекта от размера команды, а не от требований [17]
Feature-Driven Development (FDD)	Большие проекты, где важны качественное проектирование и независимость модулей, например, для проектов, требующих высококачественной модели	Не выполняется сбор и анализ требований, отсутствует управление изменениями и командная работа, системный дизайн и диаграммы считаются лишь рабочими документами, процесс зависит от главного разработчика (что ограничивает коллективное владение кодом) [17]

Методология	Область применения	Ограничения
Dynamic Systems Development Method (DSDM)	Небольшие проекты с четкими сроками и ресурсами, особенно если важны гибкость и строгое соблюдение сроков	Не подходит для жизненно важных проектов и управления большими командами, слабо поддерживает средние и сложные проекты, неэффективен для научных и инженерных проектов [17]
Test-Driven Development (TDD)	Проекты с частыми изменениями кода, где важно поддержание высокого качества через автоматическое тестирование и улучшение архитектуры	Требуются навыки тестирования у разработчиков, плохо подходит для проектов с высокими зависимостями, мало документации, трудности при множественных ошибках тестов [17]

Хотя SCRUM является наиболее распространённой методологией [15, 20, 21], разработчики зачастую не верят, что организация соблюдает принципы и ценности SCRUM [22]. Всё больше команд выбирают бережливые подходы [23], ставя под сомнение эффективность ежедневных стендап-встреч, использование пользовательских историй вместо технического задания и других артефактов [21]. Даже приверженцы SCRUM говорят о необходимости использования дополнительных инструментов [24].

При прочих равных условиях использование SCRUM по сравнению с Kanban может позволить увеличить прибыль на 8,5% и сократить расходы на 15,3% [25]. Но SCRUM лучше применять только тогда, когда команды прошли квалификацию. При использовании SCRUM есть более высокий риск недооценки сроков и бюджета. Поэтому выбор должен осуществляться исходя из условий работы и срочности, частоты изменения требований и неопределённости в будущем, размера и сложности, и уровня зрелости команды [26].

DevOps – подход, объединяющий группы разработчиков и эксплуатации для обеспечения непрерывной разработки, который еще не доведён до необходимого уровня [27]. Он действительно сокращает сроки, повышает производительность, позволяет управлять быстрыми циклами [28]. Однако есть большое количество факторов, влияющих на

успех, новые роли и обязанности, наличие рисков [29, 30]. В сочетании с Agile, например, расширение BizDevOps, нацелено на внедрение гибкости в классическую архитектуру [31]. Позволяет увеличить частоту релизов, но требует более широкий набор разносторонних навыков и зависит от уровня зрелости [32]. Внедрение может быть затруднено из-за недостатка структурированных руководств [33].

При всей популярности гибких методологий, каждый подход имеет массу ограничений, связанных с заинтересованными сторонами и командой, подходом к процессу и планированию, проектной работой, доставкой и изменениями [34]. Немаловажным фактором является зрелость команды [35] и наличие организационных навыков у всех членов команды [36]. Даже авторы Agile манифеста признают, что Agile стал сложнее и его необходимо перезагрузить, часто встречается фальшивая гибкость, и переход на Agile не приносит заложенных в него преимуществ [37]. Именно поэтому в последнее время всё больше фокус внимания направляется на гибридные методологии.

Гибридные методологии. Наиболее часто встречается объединение традиционных или гибких подходов с методологией SCRUM.

Объединение Waterfall и SCRUM позволяет уменьшить ограничения каждого из подходов [38]. Фазы инициирования и

планирования реализуются с использованием Waterfall, а сама деятельность осуществляется с помощью SCRUM или Kanban [39]. Объединение традиционных и гибких методологий обеспечивает расширение прав и возможностей членов команды, что дает им чувство причастности и повышает мотивацию [40].

Объединение SCRUM и RUP улучшает адаптивность проектов к изменениям, сохраняя преимущества традиционных методов. Однако при этом остаётся сложность интеграции гибких подходов в крупные организации, привыкших к жестким регламентам, и необходима дополнительная подготовка команд [41].

Scrumban – ещё одна популярная гибридная методология, в которой основа берётся из SCRUM, а доска заимствуется из Kanban [42].

Использование гибридных методологий требует наличия высокой квалификации руководителей проектов, так как отсутствие должного опыта может привести к ухудшению результатов. Вместо того, чтобы убрать ограничения одного метода, будут наложены ограничения нескольких методов [43], например, при использовании гибридного подхода, объединяющего UCD / User-centered design, Lean Startup и XP [44] вместо преимуществ могут появиться новые ограничения.

Материалы и методы. Для достижения поставленной цели и решения задач были использованы различные методы сбора и обработки информации. В основе исследования лежит систематический анализ литературы по методологиям управления IT-проектами.

По информации, содержащейся в открытых источниках, был составлен список общепризнанных авторов, внёсших наибольший вклад в развитие программной инженерии. После чего был произведён поиск их свежих публикаций в базах данных Scopus, Web of Science и Google Scholar.

Следующим шагом было составление набора ключевых слов и запросов, на основе которых производился поиск. При поиске особое внимание уделялось компетентности авторов, авторитетности издания и содержанию самой статьи.

При проведении исследования использовались сравнительный анализ для определения сильных и слабых сторон различных

подходов и определения области их применения и систематический анализ для обобщения и классификации полученных данных.

Для статистической оценки были использованы данные о применении различных подходов в проектах, отличающихся масштабом, уровнем сложности и критически значимыми факторами. Для структурирования и формулировки выводов использовались методы анализа и синтеза.

Результаты исследования. На основе анализа выбранных источников были выявлены особенности применения различных методологий.

Гибкие методологии лучше использовать при работе небольшими командами, когда контроль качества не является критически важным и есть большая вероятность внесения изменений после начала проекта. В большинстве случаев рекомендуется использовать не отдельно взятую методологию, а сочетать Scrum в качестве основы и Kanban доску для визуализации. Для небольших проектов в рамках спринта можно добавить парное программирование из XP. Если команда распределенная или часть задач выполняется дистанционно, стоит использовать онлайн инструменты для синхронизации работы, в первую очередь, это относится к Kanban доске.

Традиционные методологии больше подходят для крупных проектов, в которых критически важно обеспечение контроля качества и есть возможность принудительно ограничивать число вносимых изменений. Тем не менее, здесь целесообразно добавлять элементы гибкости, промежуточные итерации с обратной связью. Это позволит минимизировать риск поздней обратной связи и необходимости переделывать весь проект. Даже совсем небольшие проекты, вероятность расширения которых близка к нулю, хотя это сейчас может быть не популярно, рекомендуется реализовывать с помощью Waterfall и использовать монолитную архитектуру.

Гибридные методологии, особенно при объединении традиционных и гибких методов, позволяют обеспечить баланс между строгостью и гибкостью и могут быть адаптированы под самые разные проекты. Рекомендуется начать с Waterfall для четкого планирования и составления документации, а

непосредственно разработку осуществлять, опираясь на рекомендации по использованию гибких методологий. Добавление практики DevOps тоже имеет место быть, но только при условии достаточной зрелости команды.

Исследование подтверждает, что проблемы часто связаны не столько с уровнем компетенции разработчиков, сколько с низким уровнем организации процессов в команде. Анализ ключевых факторов подтвердил, что распределение ролей, обратная связь и наличие структурированных процессов являются определяющими условиями для успешной реализации IT-проектов.

Результаты исследования имеют практическую значимость для менеджеров проектов, предоставляя краткую информацию по области применения и ограничениям наиболее популярных методологий.

Неправильный выбор подхода к проектированию может привести не только к низкому качеству, но и к полному провалу проекта. В качестве наиболее характерного примера такого провала можно привести проект модернизации инфраструктуры информационных технологий ФБР США «Виртуальное досье» (Virtual Case File), который обошелся бюджету США в сумму \$ 170 млн и был закрыт. Причины провала – нарушение принципов разработки программного обеспечения [45]. Еще несколько примеров, таких как провал ресурса Healthcare.gov или миссии NASA Mars Climate Orbiter, свидетельствуют о том, что практически неограниченные бюджеты, ресурсы, возможность использовать лучших разработчиков не имеют значения, если неправильно выстроено управление.

Выводы и рекомендации. На основании проведенного анализа можно сделать вывод, что наличие множества самых различных технологий разработки программного обеспечения свидетельствует об отсутствии универсальной методологии. Выбирать методологию следует исходя из специфики и характеристик проекта и команды разработчиков. При наличии должного уровня компетентности можно комбинировать методы и использовать гибридные подходы для компенсации ограничений отдельно взятой методологии. Тем не менее, на данный момент не существует общеизвестной комбинации методов, которая бы позволила избавиться от

большинства ограничений и была бы применима при различных спецификах и характеристиках проекта и команды разработчиков.

Перед стартом проекта необходимо оценить зрелость команды и опыт работы с выбранной методологией. При недостатке опыта стоит внедрить фреймворки поддержки. Экспериментировать с объединением подходов и использовать гибридный подход стоит только после обучения команды.

При выборе методологии стоит обратить внимание и на не самые очевидные моменты.

1) Scrum пропагандирует гибкость, является самой популярной Agile методологией, но её гибкость не следует понимать буквально. Строгая привязка к артефактам и спринтам может затруднить адаптацию под реальные потребности команды.

2) Традиционный Waterfall – это не абсолютно «вчерашний день». Несмотря на свою репутацию устаревшего метода, Waterfall остаётся стандартом для проектов с фиксированными требованиями, например в космической отрасли или ВПК.

3) Гибридные подходы не решают за вас все проблемы. При отсутствии опыта внедрение гибридных моделей может усугубить существующие ограничения, вместо того чтобы компенсировать их.

Отдельная рекомендация касается владельцев бизнеса. В условиях ограниченного бюджета может возникнуть желание сэкономить на дорогостоящей позиции руководителя проекта и самостоятельно управлять разработкой. Скорее всего, если нет должной компетенции, это приведёт к провалу. Лучше использовать меньше программистов, но поставить квалифицированного менеджера проекта. Хороший менеджер принесёт больше ценности, чем несколько разработчиков.

В дальнейшем планируется изучение отдельно взятых методов в рамках большинства общеизвестных методологий для определения их совместимости и комбинации методов. Основной целью дальнейших исследований является минимизация ограничений и применимость в большинстве проектов для обеспечения должного качества и скорости разработки без увеличения бюд-

жета и необходимости выбирать между сотней методологий или самостоятельной

комбинацией методов в рамках гибридного подхода.

Список источников

1. Kuhrmann M. et al. What Makes Agile Software Development Agile? *IEEE Transactions on Software Engineering*. 2021. Vol. 48. No. 9. pp. 3523–3539. (In Eng.). DOI: 10.1109/TSE.2021.3099532.
2. Andrew G., Blaize R., Pedro S. Agile, Traditional, and Hybrid Approaches to Project Success: Is Hybrid a Poor Second Choice? *Project Management Journal*. 2020. No. 52 (1). (In Eng.). DOI: 10.1177/8756972820973082.
3. 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). *Silhavy, R. (eds) Intelligent Algorithms in Software Engineering. CSOC 2020. Advances in Intelligent Systems and Computing, Springer, Cham*. 2020. Vol. 1224. pp. 310–322. (In Eng.). DOI: 10.1007/978-3-030-51965-0_27.
4. 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. (In Eng.). DOI: 10.15439/2020F46.
5. Lecomber A. Value of Teaching PRINCE2 Project Management Methodology. *Tatnall A. (eds) Encyclopedia of Education and Information Technologies. Springer, Cham*. 2020. pp. 1758–1761. (In Eng.). DOI: 10.1007/978-3-030-10576-1_199.
6. Graessler I., Hentze J. The new V-Model of VDI 2206 and its validation. *Automatisierungstechnik*. 2020. Vol. 68 (5). pp. 312–324. (In Eng.). DOI: 10.1515/auto-2020-0015.
7. Patrucco A. S., Canterino F., Minelgaite I., How do Scrum Methodologies Influence the Team's Cultural Values? A Multiple Case Study on Agile Teams in Nonsoftware Industries. *IEEE Transactions on Engineering Management*. 2022. Vol. 1. No. 69 (6). pp. 1–11. (In Eng.). DOI: 10.1109/TEM.2022.3146717.
8. Saravanas A., Curinga M. Simulating the Software Development Lifecycle: The Waterfall Model. *Applied System Innovation*. 2023. Vol. 6 (6). P. 108. (In Eng.). DOI: 10.3390/asi6060108.
9. Theo Thesing T., Feldmann C., Burchardt M. Agile versus Waterfall Project Management: Decision Model for Selecting the Appropriate Approach to a Project. *Procedia Computer Science*. 2021. Vol. 181. pp. 746–756. (In Eng.). DOI: 10.1016/j.procs.2021.01.227.
10. Saeedi K., Visvizi A. Software Development Methodologies, HEIs, and the Digital Economy. *Education Sciences*. 2021. Vol. 11 (2). No. 73. 21 p. (In Eng.). DOI: 10.3390/educsci11020073.
24. Jacobson I., Sutherland J., Kerr B., Buhnova B. Better Scrum through Essence. *Software: Practice and Experience*. 2022. Vol. 52. No. 6. pp. 1531–1540. (In Eng.). DOI: 10.1002/spe.3070.
25. 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. Vol. 9. No. 4. pp. 693–700. (In Eng.). DOI: 10.13189/ujaf.2021.090415.
26. Zayat W., Senvar, O. Framework Study for Agile Software Development Via Scrum and Kanban. *International Journal of Innovation and Technology Management*. 2020. Vol. 17. No. 4. (In Eng.). DOI: 10.1142/S0219877020300025.
27. Jayakody J. A. V. M. K., Wijayanayake J. Challenges for Adopting DevOps in Information Technology Projects. *2021 International Research Conference on Smart Computing and Systems Engineering (SCSE), Colombo, Sri Lanka*. 2021. pp. 203–210. (In Eng.). DOI: 10.1109/SCSE53661.2021.9568348.
28. Angara J., Prasad S., Gutta S. DevOps Project Management Tools for Sprint Planning, Estimation and Execution Maturity. *Cybernetics and Information Technologies*. 2020. Vol. 20. No. 2. pp. 79–92. (In Eng.). DOI: 10.2478/cait-2020-0018.
29. 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. No. 2. pp. 257–280. (In Eng.). DOI: 10.1002/spe.3269.
30. Azad N., Hyrynsalmi S. Mäntymäki M. Understanding DevOps Critical Success Factors: Insights from Professionals. *New Sustainable Horizons in Artificial Intelligence and Digital Solutions*. 2023. pp. 78–90. (In Eng.). DOI: 10.1007/978-3-031-50040-4_7.
31. Fuentes-Quijada G., Ruiz-González F., Caro A. Challenges as Regards Aligning IT and Business with Agility when Applying the BizDevOps Approach. *Enterprise Information Systems, 25th International Conference, ICEIS 2023*. 2023. pp. 201–219. (In Eng.). DOI: 10.1007/978-3-031-64755-0_10.
32. Aymeric H., Barbara L., Frantz R., Brian F. From Agile to DevOps: Smart Skills and Collaborations. *Information Systems Frontiers*. 2020. Vol. 22. No. 3. pp. 927–945. (In Eng.). DOI: 10.1007/s10796-019-09905-1.
33. Almeida F., Simões J., Lopes S. Exploring the Benefits of Combining DevOps and Agile. *Future Internet*. 2022. Vol. 14 (2). No. 63. (In Eng.). DOI: 10.3390/fi14020063.

11. Ciric Lalic D., Lalic B., Delić M., Gracanin D., Stefanovic D. How Project Management Approach Impact Project Success? From Traditional to Agile. *International Journal of Managing Projects in Business*. 2022. Vol. 15 (3). pp. 494–521. (In Eng.). DOI: 10.1108/IJMPB-04-2021-0108.
12. Islam A. K. M., Ferworn A. A Comparison between Agile and Traditional Software Development Methodologies. *Global Journal of Computer Science and Technology*. 2020. Vol. 20 (2). pp. 7–42. (In Eng.). DOI: 10.34257/GJCSTCVOL20IS2PG7.
13. Pargaonkar S. A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. *International Journal of Scientific and Research Publications*. 2023. Vol. 13 (8). pp. 120–124. (In Eng.). DOI: 10.29322/IJSRP.13.08.2023.p14015.
14. 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. (In Eng.). DOI: 10.1016/j.jss.2020.110732.
15. 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. (In Eng.). DOI: 10.3991/ijim.v14i11.13269.
16. Malik M., Sarwar S., Orr S. Agile Practices and Performance: Examining the Role of Psychological Empowerment. *International Journal of Project Management*. 2021. Vol. 39 (1). pp. 10–20. (In Eng.). DOI: 10.1016/j.ijproman.2020.09.002.
17. 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. (In Eng.). DOI: 10.22937/IJCSNS.2020.20.10.22.
18. Muhammad I., Shabib A., Munir A., Ahmed I., Bilal S.K., Muhammad I., Baha N.S.I., Nouh S.E. Presenting and Evaluating Scaled Extreme Programming Process Model. *International Journal of Advanced Computer Science and Applications*. 2020. Vol. 11 (11). pp. 163–171. (In Eng.). DOI: 10.14569/IJACSA.2020.0111121.
19. 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. (In Eng.). DOI: 10.14807/ijmp.v12i8.1482.
20. Sutherland J., Jacobson, I., Kerr B. Scrum Essentials Cards. *Queue*. 2020. Vol. 18. No. 3. pp. 83–106. (In Eng.). DOI: 10.1145/3411757.3418775.
21. Putri P., Raharjo T., Hardian B., Simanungkalit T. Challenges and Best Practices Solution of Agile Project Management in Public Sector: A Systematic Literature Review. *JOIV International Journal on Informatics Visualization*. 2023. Vol 7 (2). pp. 606–614. (In Eng.). DOI: 10.30630/joiv.7.2.1098.
22. Fernando L., Paula M., Vanessa B.B., Graciano, Paola G., Guilherme T. Does Maturity Level Influence the Use of Agile UX Methods by Digital Startups? Evaluating Design Thinking, Lean Startup, and Lean User Experience. *Information and Software Technology*. 2023. Vol. 154. (In Eng.). DOI: 10.1016/j.infsof.2022.107107.
23. Venkatesh V., Thong J.Y.L., Chan F. K. Y., Hoehle H., Spohrer K. How Agile Software Development Methods Reduce Work Exhaustion: Insights on Role Perceptions and Organizational skills. *Info Systems J*. 2020. Vol. 30 (4). pp. 733–761. (In Eng.). DOI: 10.1111/isj.12282.
24. Cockburn A., Kerievsky J., Hunt A. Rebooting Agile. *Cutter Business Technology Journal*. 2019. Vol. 32 (3). pp. 6–12. (In Eng.).
25. Yahya N., Maidin S. The Waterfall Model with Agile Scrum as the Hybrid Agile Model for the Software Engineering Team. *10th International Conference on Cyber and IT Service Management (CITSM)*. 2022. pp. 1–5. (In Eng.). DOI: 10.1109/CITSM56380.2022.9936036.
26. Făgărășan C., Popa O., Pisla A., Cristea C. Agile, Waterfall and Iterative Approach in Information Technology Projects. *IOP Conference Series: Materials Science and Engineering*. 2021. Vol. 1169 (012025). (In Eng.). DOI: 10.1088/1757-899X/1169/1/012025.
27. 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. (In Eng.). DOI: 10.1177/1063293X20958541.
28. Chandra P., Reddy S., Reddy P.C., Nachiyappan S., Ramakrishna V., Senthil R., Anwer S., Rk K. Hybrid Model Using Scrum Methodology for Software Development System. *Journal of Nuclear Energy Science & Power Generation Technology*. 2021. Vol. 10 (9). pp. 1–6. (In Eng.).
29. Fuentes-Del-Burgo J., Sebastián M. Comparative Analysis of the Board Tool in the Agile Methodologies Scrum, Kanban and Scrumban in Software Projects. *26th International Congress on Project Management and Engineering, Terrassa, Spain*. 2022. pp. 1660–1671. (In Eng.).
30. Gunawan F., Budiardjo E.K. A Quest of Software Process Improvements in DevOps and Kanban: A Case Study in Small Software Company. *In Proceedings of the 2021 4th International Conference on Software Engineering and Information Management*

21. Morandini M., Coleti 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. (In Eng.). DOI: 10.1016/j.cosrev.2020.100314.
22. Grebic B., Stojanović A. Application of the Scrum Framework on Projects in IT Sector. *European Project Management Journal*. 2021. Vol. 11. No. 2. pp. 37–46. (In Eng.). DOI: 10.18485/epmj.2021.11.2.4.
23. Hassanein E. E., Hassanien S. A. Cost Efficient Scrum Process Methodology to Improve Agile Software Development. *International Journal of Computer Science and Information Security*. 2020. Vol. 18. No. 4. pp. 123–131.
- (*ICSIM '21*). Association for Computing Machinery, New York, NY, USA. 2021. pp. 39–45. (In Eng.). DOI: 10.1145/3451471.3451478.
44. Zorzetti M., Signoretti I., Salerno L., Marczak S., Bastos R. Improving Agile Software Development using User-Centered Design and Lean Startup. *Information and Software Technology*. 2022. Vol. 121. (In Eng.). DOI: 10.1016/j.infsof.2021.106718.
45. Goldstain H. Who Killed the Virtual Case File? How the FBI Blew More than \$100 Million on Case-Management Software it Will Never Use. *IEEE Spectrum*. Sept., 01, 2005 [Электронный ресурс]. – Режим доступа: <https://spectrum.ieee.org/who-killed-the-virtual-case-file> (In Eng.).