

ИСПОЛЬЗОВАНИЯ МЕТОДОЛОГИЙ ПРОГРАММИРОВАНИЯ В СОВРЕМЕННЫХ ПРОЕКТАХ

Вергель И. В., Барсукевич С.Н.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Мигалевич С.А. – магистр техн. наук, ст. преподаватель кафедры информатики

Аннотация. В статье рассматриваются основные методологии разработки программного обеспечения: *Waterfall*, *Agile*, *Scrum* и *Kanban*. Описаны их ключевые особенности, преимущества и недостатки, а также предложены рекомендации по выбору оптимальной методологии для различных типов проектов.

Ключевые слова: методологии программирования, *Agile*, *Scrum*, *Waterfall*, *Kanban*, управление проектами.

Введение. Современная разработка программного обеспечения требует не только высокого уровня технической подготовки, но и грамотного управления процессами. Методологии программирования представляют собой наборы принципов, практик и процессов, которые помогают организовать разработку программного обеспечения. Они определяют, как команда будет взаимодействовать, как будут распределены задачи и как будет контролироваться качество продукта. Однако выбор неподходящей методологии может привести к увеличению сроков разработки, превышению бюджета и снижению конечного продукта.

Основная часть. Одной из самых традиционных методологий является *Waterfall* (каскадная модель). Этот подход предполагает последовательное выполнение этапов разработки: анализ требований, проектирование, разработка, тестирование и внедрение. *Waterfall* хорошо подходит для проектов с четко определенными и неизменными требованиями, например, разработка сложных систем управления или государственных проектов. Однако его главный недостаток — отсутствие гибкости. Если требования изменяются в процессе разработки, это может привести к значительным задержкам и дополнительным затратам [1].

В отличие от *Waterfall*, *Agile* (гибкая методология) предлагает итеративный подход. Разработка ведется небольшими этапами, называемыми спринтами, которые длятся от одной до четырех недель. В конце каждого спринта команда демонстрирует заказчику готовый функционал, что позволяет быстро реагировать на изменения и корректировать требования. *Agile* особенно эффективен для проектов, где требования могут изменяться в процессе разработки, таких как стартапы или мобильные приложения [2].

Одним из наиболее популярных подходов в рамках *Agile* является *Scrum*. *Scrum* предполагает четкое распределение ролей: *Scrum*-мастер, *Product Owner* и команда разработки. *Scrum*-мастер помогает команде следовать *Scrum*-процессу, удаляет препятствия и организует встречи. Он следит за соблюдением принципов *Scrum*, но не участвует в принятии решений по содержанию работы. *Product Owner*, в свою очередь, отвечает за управление продуктом, формирует и приоритизирует список задач, решая, какие должны быть выполнены в следующем спринте. Он работает с командой разработки, чтобы результат соответствовал потребностям пользователей. *Scrum* обеспечивает высокую прозрачность процесса разработки и позволяет команде постоянно улучшать свои процессы [3].

Другим популярным подходом в рамках *Agile* является *Kanban*. В отличие от *Scrum*, он не требует фиксированных спринтов, а фокусируется на визуализации рабочего процесса с помощью доски, разделенной на этапы, такие как «Запланировано», «В работе» и

"Готово". Важным элементом данной методологии является ограничение количества задач в работе, что предотвращает перегрузку команды и способствует равномерному распределению нагрузки.

Для выбора оптимальной методологии программирования необходимо обозначить ряд ключевых моментов:

1 Четко определить цели и задачи проекта, включая ожидаемые результаты и ключевые метрики успеха, а также понять, что является приоритетом: скорость выполнения, качество продукта или его инновационность;

2 Оценить характеристики проекта, такие как его сложность с технической и организационной точек зрения, масштаб и длительность (краткосрочный или долгосрочный);

3 Оценить команду проекта: ее опыт и навыки, размер, уровень автономности и готовность к изменениям и адаптации к новым подходам;

4 Проанализировать ресурсы проекта и их количество, включая временные, финансовые и технические, а также возможность привлечения дополнительных ресурсов;

5 Оценить риски проекта, уровень неопределенности и наличие механизмов управления рисками в рамках методологии.

Компания *Spotify* является одним из наиболее ярких примеров успешного внедрения методологии *Agile*. В начале 2010-х годов компания, разрабатывающая музыкальный стриминговый сервис, столкнулась с необходимостью быстро адаптироваться к изменениям на рынке. Конкуренция в сфере музыкального стриминга была высокой, и требования пользователей постоянно менялись. Чтобы оставаться на плаву, *Spotify* нуждалась в методологии, которая позволила бы ей быстро выпускать новые функции и улучшать продукт [4].

Компания выбрала *Agile*, а именно *Scrum* и *Kanban*, но не остановилась на этом. *Spotify* разработала собственную модель *Agile*, известную как *Spotify Model*. В этой модели команды, называемые *squads*, работают автономно, но координируются через общие цели, объединяемые в *tribes* [4].

Результаты внедрения *Agile* в *Spotify* оказались впечатляющими. Компания смогла быстро адаптироваться к изменениям на рынке, улучшить качество продукта за счет регулярных итераций и обратной связи, а также повысить мотивацию сотрудников благодаря предоставленной им автономии [4].

В то время как *Spotify* успешно внедряла *Agile*, некоторые крупные банки столкнулись с серьезными трудностями при попытке использовать эту методологию для управления проектами разработки программного обеспечения. Банки, стремясь ускорить разработку и конкурировать с финтех-стартапами, решили перейти на *Agile*, включая *Scrum* и *Kanban*.

Однако этот переход оказался неудачным. Основной проблемой стало отсутствие понимания *Agile* среди сотрудников. Многие из них не были готовы к переходу на новую методологию, что привело к хаосу в управлении проектами. Это в свою очередь привело к срыву сроков и бюджетов. Кроме того, банковская сфера требует строгой документации и контроля, что противоречит принципам *Agile*, основанным на гибкости и минимальной бюрократии. Иерархическая структура банков также стала барьером для внедрения *Agile*, который требует высокой степени автономии команд, поэтому многие отказались от этой методологии в пользу более структурированных подходов, таких как *Waterfall*.

Национальное управление по аэронавтике и исследованию космического пространства (*NASA*) является примером успешного использования *Waterfall* для реализации крупных проектов, таких как разработка космических кораблей и спутников.

Компания выбрала методологию *Waterfall*, потому что в космических проектах требования обычно четко определены и редко меняются. Ошибки в таких разработках могут иметь катастрофические последствия, поэтому необходим строгий контроль качества. Кроме того, проекты *NASA* должны соответствовать строгим стандартам и требованиям, что делает *Waterfall* идеальным выбором [5].

Результаты применения методологии *Waterfall* в NASA демонстрируют ее эффективность в реализации сложных инженерных проектов. Компания успешно реализовала сложные проекты, такие как *Mars Rover*, минимизировала риски за счет строгой документации и тестирования и обеспечила высокое качество продукции [5].

В то время как NASA успешно использует *Waterfall*, многие стартапы столкнулись с провалом при попытке применить эту методологию для разработки программного обеспечения. Стартапы, ожидая четкого плана и не имея опыта управления проектами, часто выбирали *Waterfall* как "классический" подход.

Однако *Waterfall* оказался неэффективным в условиях быстро меняющегося рынка стартапов. Основной проблемой стала неспособность адаптироваться к изменениям. На этапе тестирования, когда продукт был почти готов, оказывалось, что требования рынка уже изменились, и продукт больше не соответствовал ожиданиям пользователей.

В результате многие стартапы потерпели неудачу и были вынуждены перейти на *Agile* или другие гибкие методологии.

Заключение. Выбор методологии разработки программного обеспечения играет ключевую роль в успехе проекта, определяя эффективность управления процессами, распределения ресурсов и адаптации к изменениям. Анализ применения различных подходов показывает, что *Waterfall* оказывается наиболее эффективным в проектах с четко определенными требованиями, высокой степенью регуляторного контроля и строгими стандартами качества. В то же время *Agile*, включая *Scrum* и *Kanban*, демонстрирует свою эффективность в динамичных средах, где необходима гибкость и быстрая адаптация к изменениям.

Однако универсального решения не существует. Выбор методологии должен основываться на характеристиках проекта, уровне неопределенности, требованиях к документированию и культуре организации. Успешное управление проектами требует не только знания различных подходов, но и способности адаптировать их к конкретным условиям, комбинируя элементы различных методологий для достижения оптимального результата.

Список литературы

1. A Comparative Study of Agile and Waterfall Software Development Methodologies / Shamsulhuda Khan, Shubhangi Mahadik. Режим доступа: <https://engineering.atspotify.com/2014/03/spotify-engineering-culture-part-1/>. Дата доступа: 11.02.2025.
2. Дизайн Agile-организаций / Павличенко И. Е. СПб.: Питер, 2024. – 208с
3. Scrum Guide: The Definitive Guide to Scrum / K. Schwaber, J. Sutherland. Режим доступа: <https://www.scrumguides.org>. Дата доступа: 11.02.2025.
4. Spotify Engineering Culture [Электронный ресурс]. Режим доступа: <https://engineering.atspotify.com/2014/03/spotify-engineering-culture-part-1/>. Дата доступа: 12.02.2025.
5. SEH 3.0 NASA Program/Project Life Cycle [Электронный ресурс]. Режим доступа: <https://www.nasa.gov/reference/3-0-nasa-program-project-life-cycle/>. Дата доступа: 12.02.2025.

UDC 005

USING PROGRAMMING METHODOLOGIES IN MODERN PROJECTS

Verghel I.V., Barsukevich S.N.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Migalevich S.A. – Master of Tech. Sci, senior lecturer at the department of computer science

Annotation. The article discusses the main software development methodologies: Waterfall, Agile, Scrum and Kanban. Their key features, advantages and disadvantages are described, as well as recommendations for choosing the optimal methodology for various types of projects.

Keywords: programming methodologies, Agile, Scrum, Waterfall, Kanban, Project management.