ORES

Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

УДК 75.04

СРАВНЕНИЕ МЕТОДОЛОГИЙ РАЗРАБОТКИ ПО

Свищёв Андрей Владимирович,
Старший преподаватель кафедры практической и прикладной информатики
МИРЭА-Российский технологический университет (РТУ МИРЭА), г. Москва.

Беликов Илья Владиславович,
Студент магистратуры, 2 курс
МИРЭА-Российский технологический университет (РТУ МИРЭА), г. Москва

Аннотация

В данной статье сравниваются основные методологии разработки ПО. Проанализированы достоинства и недостатки при работе с ними, которые могут быть использованы в разработке приложений.

Ключевые слова: методология, ПО, разработка, agile, scrum.

COMPARISON OF SOFTWARE DEVELOPMENT METHODOLOGIES

Svishchev Andrey Vladimirovich,
Senior Lecturer at the Department of Practical and Applied Informatics
MIREA - Russian Technological University (RTU MIREA), Moscow.

Belikov Ilya Vladislavovich,
Master's student, 2st year
MIREA - Russian Technological University (RTU MIREA), Moscow

ABSTRACT

This article compares the main software development methodologies. The advantages and disadvantages of working with them, which can be used in application development, are analyzed.

Keywords: methodology, software, development, agile, scrum.

Разработка программного обеспечения требует продуманного подхода, который позволяет минимизировать риски и максимально эффективно использовать ресурсы. Для организации процесса применяются различные методологии, каждая из которых формировалась под влиянием конкретных требований времени, особенностей проектов и доступных технологий.

Сегодня наиболее популярными являются каскадная модель, гибкие подходы, такие как Agile и Scrum, спиральная модель, а также культура DevOps. Каждая из них обладает характерными чертами, которые делают ее эффективной в определенных условиях. Чтобы понять, какая методология лучше подходит для конкретного проекта, необходимо подробно рассмотреть их специфику и области применения.

Каскадная модель, известная также как «водопад», представляет собой линейный подход к разработке. В основе лежит последовательное выполнение этапов: от определения требований до сопровождения готового продукта [1]. Такой подход подходит для проектов, где требования четко определены заранее и не подлежат значительным изменениям.

Чаще всего каскадную модель используют в областях, где крайне важна стабильность продукта, а ошибки на этапе разработки могут стоить слишком дорого. Например, в авиационной и медицинской индустрии, где программные сбои могут привести к серьезным последствиям, требуется предельная строгость выполнения всех этапов [2]. Еще одним примером могут служить государственные проекты, где бюрократические процедуры требуют полной документации на каждом этапе.

Основным преимуществом каскадного подхода является его предсказуемость. Однако он плохо подходит для проектов, где существует вероятность изменения требований. Длительный процесс завершения одного этапа до начала следующего делает эту модель неэффективной в условиях высокой неопределенности или динамичных рынков.

Гибкие методологии, объединенные общими принципами Agile, стали ответом на недостатки каскадного подхода. Основное внимание здесь уделяется взаимодействию между участниками процесса, быстрой адаптации к изменениям и созданию рабочего программного обеспечения в кратчайшие сроки. Agile лучше всего подходит для проектов, где требования заказчика не могут быть полностью определены на старте, а также для среды, где изменения неизбежны [3]. Веб-разработка — яркий пример области, где Agile показывает свою эффективность. При создании сайтов или приложений часто бывает трудно заранее определить все функции и интерфейсы. В процессе работы заказчики могут менять приоритеты, что делает гибкость критически важной.

Гибкость Agile позволяет командам фокусироваться на наиболее важных функциях, создавая прототипы, которые могут быть быстро протестированы и оценены пользователями. Однако для успешного применения требуется хорошо подготовленная команда и постоянное участие заказчика. Отсутствие формальной документации может стать проблемой, особенно если проект передается другой группе разработчиков.

Scrum, будучи одной из реализаций Agile, привносит в гибкие методологии четкие процессы. Он организован вокруг коротких циклов — спринтов, по окончании которых команда представляет готовый результат. Такая структура позволяет сочетать преимущества гибкости с необходимостью регулярной отчетности. Scrum становится особенно эффективным в небольших командах, работающих над проектами с быстро меняющимися требованиями [4]. Например, стартапы, разрабатывающие мобильные приложения, часто сталкиваются с необходимостью тестировать свои идеи на рынке. Каждый спринт позволяет таким командам выпускать минимально жизнеспособный продукт, который можно быстро протестировать на реальных пользователях. Однако Scrum требует высокой дисциплины. Роли внутри команды должны быть четко определены, а ежедневные встречи и регулярные ретроспективы требуют больших затрат времени. В крупных проектах, где участвуют десятки или сотни человек, Scrum может стать трудным для координации. В таких случаях лучше рассмотреть использование специализированных гибких методологий для масштабирования.

Спиральная модель предлагает уникальный подход к разработке, совмещая элементы последовательности и итеративности. Каждый виток спирали включает этапы

анализа, проектирования, реализации и оценки рисков. Это делает методологию особенно подходящей для проектов с высокой степенью неопределенности. Корпоративные системы и сложные программные комплексы — основные примеры, где спиральная модель демонстрирует свои преимущества. Например, в банковской сфере, где требуется разработка надежных систем с высокой степенью безопасности, этот подход позволяет минимизировать риски за счет тщательного тестирования и анализа [5]. Тем не менее, спиральная модель требует значительных ресурсов. Её реализация возможна только при наличии опытной команды и достаточного финансирования. Это делает её менее подходящей для небольших проектов или стартапов, где ограниченные ресурсы не позволяют провести полноценный анализ рисков.

DevOps — это не просто методология, а скорее культура, направленная на объединение процессов разработки и эксплуатации. Основное внимание уделяется автоматизации и ускорению релизов, что делает этот подход идеальным для проектов, где требуется регулярное обновление программного обеспечения. Облачные сервисы и платформы с высокой нагрузкой, такие как стриминговые сервисы или интернет-магазины, требуют быстрой доставки новых функций и поддержания высокой надежности. DevOps позволяет сократить время между релизами, внедряя автоматические тестирования и развертывание. Однако внедрение DevOps требует значительных усилий. Необходимость в автоматизации процессов, обучении сотрудников и изменении устоявшихся подходов может стать серьезным барьером. Этот подход не оправдывает себя в небольших проектах, где объем работ не оправдывает затраты на автоматизацию.

В реальных условиях выбор методологии определяется рядом факторов. Для небольших проектов, где требования четко определены заранее, каскадная модель может быть самым простым и удобным решением. Если же требования подвержены изменениям, лучшим выбором станет Agile или Scrum, которые позволяют быстро адаптироваться и работать в условиях неопределенности.

Для крупных проектов с высокой степенью сложности и неопределенности спиральная модель обеспечивает надежность за счет управления рисками. В случаях, когда важна скорость поставки обновлений и высокая степень автоматизации, DevOps становится практически незаменимым.

В большинстве современных компаний применяется комбинированный подход. Например, на начальных этапах проектирования может использоваться каскадная модель для определения требований, затем — Agile для разработки, а на стадии эксплуатации — DevOps.

Каждая из методологий имеет свои сильные и слабые стороны. Успешная разработка программного обеспечения требует не только знаний о существующих подходах, но и умения адаптировать их к конкретным задачам. Понимание контекста проекта, его масштабов и специфики помогает выбрать наиболее эффективный метод работы. В условиях быстроменяющегося мира важно не ограничиваться одной методологией, а использовать гибридные подходы, которые обеспечат максимальную ценность для конечного продукта.

Список литературы:

1. Токмаков, Г. П. Информационное и лингвистическое обеспечение локальных и распределенных автоматизированных систем: учебное пособие / Г. П. Токмаков. — Ульяновск: УлГТУ, 2022. — 333 с. — ISBN 978-5-9795-2230-2.
2. Инструментальное программное обеспечение разработки и проектирования информационных систем: учебное пособие / А. А. Куликов, В. Т. Матчин, А. В. Синицын, В. В. Литвинов. — Москва: РТУ МИРЭА, 2022.

3. Волков, М. Ю. Разработка серверных частей интернет-ресурсов: учебное пособие / М. Ю. Волков, В. В. Литвинов, А. А. Лобанов. — Москва: РТУ МИРЭА, 2021. — 188 с.
4. Архитектурные решения информационных систем / А. И. Водяхо, Л. С. Выговский, В. А. Дубенецкий, В. В. Цехановский. — 3-е изд., стер. — Санкт-Петербург: Лань, 2023. — 356 с. — ISBN 978-5-507-46063-2.
5. Токмаков, Г. П. Основы XML-технологий: учебное пособие / Г. П. Токмаков. — Ульяновск: УлГТУ, 2017. — 229 с. — ISBN 978-5-9795-1701-8.

References:

1. Tokmakov, G. P. Information and linguistic support of local and distributed automated systems: a tutorial / G. P. Tokmakov. - Ulyanovsk: UlSTU, 2022. - 333 p. - ISBN 978-5-9795-2230-2.
2. Instrumental software for the development and design of information systems: a tutorial / A. A. Kulikov, V. T. Matchin, A. V. Sinitsyn, V. V. Litvinov. - Moscow: RTU MIREA, 2022.
3. Volkov, M. Yu. Development of server parts of Internet resources: a tutorial / M. Yu. Volkov, V. V. Litvinov, A. A. Lobanov. - Moscow: RTU MIREA, 2021. - 188 p.
4. Architectural solutions for information systems / A. I. Vodyakh, L. S. Vygovsky, V. A. Dubenetsky, V. V. Tsekhanovsky. - 3rd ed., reprinted. - St. Petersburg: Lan, 2023. - 356 p. - ISBN 978-5-507-46063-2.
5. Tokmakov, G. P. Basics of XML technologies: a tutorial / G. P. Tokmakov. - Ulyanovsk: UlSTU, 2017. - 229 p. - ISBN 978-5-9795-1701-8.