



Современные методологии разработки ПО: обзор





Оглавление

CONTENTS

01. От Waterfall до Agile

02. Scrum

03. Kanban

04. DevOps



От Waterfall до Agile

1

Классическая последовательность Waterfall

Каскадная модель предполагает линейную последовательность этапов: требования, проектирование, реализация, тестирование и сопровождение. Этот подход прост в понимании и планировании, но не подходит для современных проектов с высокой неопределенностью и частыми изменениями.

2

Проблемы Waterfall

Основные проблемы Waterfall: отсутствие возможности вернуться на предыдущий этап, позднее обнаружение дефектов и ошибок, а также размытая ответственность за риски. Это приводит к задержкам и увеличению бюджета проектов.

3

Рождение Agile

В 2001 году был опубликован манифест Agile, который предложил альтернативный подход к разработке ПО. Agile фокусируется на гибкости, быстром реагировании на изменения и постоянном взаимодействии с клиентом, что стало ответом на недостатки Waterfall.

Манифест Agile : четыре ценности

Люди и взаимодействие

В Agile люди и взаимодействие важнее процессов и инструментов, что требует сплоченности и эффективной коммуникации команды для быстрой адаптации к изменениям и решения проблем.

Сотрудничество с заказчиком

Сотрудничество с заказчиком важнее условий контракта, так как его активное участие позволяет быстро вносить изменения в требования.

Работающий продукт

Работающий продукт важнее исчерпывающей документации, потому что он является функциональным и может быть улучшен.

Готовность к изменениям

Готовность к изменениям важнее следования плану — это позволяет адаптироваться и повысить успех проекта.



Scrum

Наиболее популярные и широко применяемые в настоящее время фреймворки гибкой разработки

Scrum — это облегченная структура, предназначенная для помощи командам в совместной поставке ценных продуктов посредством решения сложных адаптивных проблем. Она делает акцент на эмпирическом контроле процесса, и её три столпа заключаются в следующем:

Прозрачность: Ключевые аспекты процесса должны быть видимыми для тех, кто отвечает за результат.

Проверка: Артефакты Scrum и прогресс должны часто проверяться для выявления нежелательных отклонений.

Адаптация: Если процесс отклоняется за допустимые пределы, его необходимо скорректировать.



Scrum:

Пять точек контроля

1

Спринт

Спринт — это временная «коробка», в которой команда разрабатывает продукт. Он обычно длится от двух до четырех недель и заканчивается созданием инкремента продукта.

2

Планирование спринта

Планирование спринта — это церемония, на которой команда определяет цель спринта и выбирает задачи из Product Backlog для Sprint Backlog. Это помогает команде сосредоточиться на конкретных задачах на ближайшее время.

3

Ежедневный скрам

Ежедневный скрам — это короткая встреча, на которой команда обсуждает свои текущие задачи, проблемы и планы на день. Это помогает команде быть в курсе происходящего и оперативно решать возникающие вопросы.

Пять церемоний, пять точек контроля

4

Обзор спринта

Обзор спринта — это церемония, на которой команда демонстрирует инкремент продукта заказчику и получает обратную связь. Это позволяет быстро корректировать курс разработки в соответствии с требованиями клиента.

5

Ретроспектива спринта

Ретроспектива спринта — это церемония, на которой команда анализирует свой процесс работы, выявляет проблемы и определяет способы их улучшения. Это помогает команде постоянно совершенствовать свою работу.

Роли в Scrum

Менеджер проекта

В Scrum традиционная роль менеджера проекта упраздняется, но его обязанности не исчезают — они распределяются между тремя нижеуказанными ролями. Данная модель направлена на уменьшение командования и контроля, повышение гибкости команды, чувства ответственности и эффективности.



1

Владелец продукта

Product Owner — визионер и ответственный за ценность продукта.

Стратегическое видение продукта: Способность сформулировать и донести долгосрочную цель продукта до команды и стейкхолдеров.

Искусство управления бэклогом: Приоритизация элементов Product Backlog для максимизации ценности и достижения целей.

Экспертиза в предметной области: Глубокое понимание рынка, бизнеса и потребностей клиентов.

Эффективная коммуникация: Навыки работы со стейкхолдерами: управление ожиданиями, сбор обратной связи и прозрачность.

Принятие решений: Способность быстро принимать обоснованные решения о функциях продукта, балансируя интересы бизнеса и команды.

Роли в Scrum

2 Скрам-мастер

Scrum Master — слуга-лидер, фасилитатор и коуч команды.

Глубокое знание Scrum: Безупречное понимание теории, практик, правил и ценностей Scrum.

Мастерство фасилитации: Способность эффективно проводить все события Scrum, направляя команду к достижению целей встречи.

Коучинг и менторинг: Умение задавать правильные вопросы и помогать команде и Продакт-Овнеру самостоятельно находить решения и совершенствовать процессы.

Устранение препятствий: Способность выявлять и помогать устранять барьеры, мешающие прогрессу команды.

Служащее лидерство: Фокус на потребностях команды и организации. Цель — empowered и самоорганизующаяся команда.

Развитие организаций: Помощь организации в понимании и принятии Agile-подхода.



3 Команда разработки

Это не просто группа исполнителей, получающих задания. Это самоорганизующаяся единица, которая сама решает, как лучше достичь цели спринта. Она кроссфункциональна, то есть внутри нее есть все необходимые навыки для создания работающего продукта. Самое главное — она несет коллективную ответственность за результат. Это означает, что команда в целом отвечает за успех или неудачу, что мотивирует к совместной работе и взаимопомощи.

Лидерство и Фасилитация в Scrum

Лидерство в Scrum - Совместное сервисное лидерство, распределенное между ролями

Продуктовое лидерство - определяет стратегию и ценность продукта через видение и приоритизацию задач.

Процессное лидерство - обеспечивает эффективность работы через устранение препятствий и совершенствование методологии.

Техническое лидерство - проявляется в самоорганизации команды, коллективной ответственности за качество и постоянном совершенствовании профессионального мастерства.

Фасилитация в Scrum: управление процессом для достижения целей

Скрам-мастер выступает главным фасилитатором, который проводит все события фреймворка, сохраняя нейтральную позицию.

Ценность такой фасилитации заключается в том, чтобы каждый голос был услышан, что ведет к более обоснованным решениям и повышению общей эффективности работы команды.



Kanban

Истоки и применение

Kanban родился в японском производстве, но идеально подходит для разработки ПО. Он позволяет командам визуализировать задачи на доске с колонками, что упрощает контроль над процессом и повышает прозрачность.

Основные принципы

Основные принципы Kanban: визуализация потока работ, ограничение одновременной работы (WIP), управление циклом разработки и непрерывное улучшение. Эти принципы помогают оптимизировать процесс и ускорить доставку продукта.



Преимущества Kanban



Гибкость

Kanban повышает гибкость: новые задачи добавляются в «Сделать» без перепланировки спринта, что позволяет оперативно реагировать на изменения.

Прозрачность

Клиенты видят прогресс в реальном времени, что укрепляет доверие и позволяет им быть в курсе текущего состояния проекта.

Эффективность

Уменьшаются простои и перегрузки за счёт ограничения одновременной работы (WIP). Это позволяет команде работать более продуктивно.

Простота внедрения

Метод легко внедряется поверх существующих процессов и не требует радикальных изменений. Это позволяет быстро начать использовать Kanban без значительных затрат.



DevOps



Определение DevOps

DevOps — это культура и набор практик, которые объединяют разработку и эксплуатацию в единый процесс. Это позволяет ускорить выпуск программного обеспечения и повысить его качество.



Цель DevOps

Основная цель DevOps — ускорить доставку новых функций к пользователям, одновременно обеспечивая стабильность и надежность системы. Это достигается за счет автоматизации и непрерывного улучшения процессов.



Преимущества DevOps

Внедрение DevOps позволяет сократить время выпуска программного обеспечения, уменьшить количество ошибок и улучшить взаимодействие между командами разработки и эксплуатации.



CI/CD как двигатель стабильности

Непрерывная интеграция

Непрерывная интеграция (CI) предполагает автоматическую сборку и тестирование кода после каждого изменения. Это позволяет быстро обнаруживать ошибки и улучшать качество продукта.

Непрерывное развертывание

Непрерывное развертывание (CD) автоматизирует процесс развертывания продукта в производственную среду. Это ускоряет время вывода продукта на рынок и улучшает его стабильность.

Культура DevOps

DevOps — это не только инструменты, но и культура совместной работы между разработчиками и операционной командой. Это позволяет устранить разрыв между ними и обеспечить быстрое и стабильное развитие продукта.



Литература

1. Эволюция методологий разработки <https://habr.com/ru/articles/778502/>
2. СМЕРНОВ К. С. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОЛОГИЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ SCRUM И KANBAN[J]. ИНТЕРНАУКА, 2022(18-1): 53-55.
3. СМЕРНОВ К. С. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОЛОГИЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ AGILE И WATERFALL[J]. Журнал, 2024, 9(2): 68-72.
4. БГАНЕ А. А., КОВАЛЕВА К. А. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОЛОГИЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ[С] // Название сборника трудов конференции. Краснодар: КубГТУ, 2023: 16-20.
5. Soumya Prakash Rath, Nikunj Kumar Jain, Gunjan Tomer, Alok Kumar Singh, "A systematic literature review of agile software development projects", Information and Software Technology, Volume 182, 2025, 107727, ISSN 0950-5849,
6. L. Rising and N. S. Janoff, "The Scrum software development process for small teams," in IEEE Software, vol. 17, no. 4, pp. 26-32, July-Aug. 2000, doi: 10.1109/52.854065.
7. D. Ismail and Arviansyah, "A Systematic Literature Review and Delphi Study on Agile Software Development Challenges," 2021 6th International Conference on Management in Emerging Markets (ICMEM), Bandung, Indonesia, 2021, pp. 1-6, doi: 10.1109/ICMEM53145.2021.9869414.



THANK YOU FOR READING!

Спасибо за внимание!