

УПРАВЛЕНИЕ ПРОЕКТАМИ: ТЕХНОЛОГИИ И ИНСТРУМЕНТАЛЬНЫЕ РЕШЕНИЯ

Шестопалов Д. А.

*Научный руководитель – к.ф.-м.н., доцент Бояришинова О. А.
Белорусский национальный технический университет,
oboyarshinova@bntu.by*

Аннотация. В условиях постоянно растущей сложности и масштаба проектов, эффективное управление проектами становится критически важным для достижения поставленных целей и задач внутри компании. Наличие единого средства для управления проектами позволяет упростить координацию между участниками команды и другими заинтересованными сторонами. Автоматизация процессов и интеграция с имеющимися продуктами снижает нагрузку на сотрудников и позволяет сосредоточиться более приоритетных задачах. Инструменты для отслеживания показателей производительности и других метрик позволяют менеджменту контролировать прогресс выполнения проектов в режиме реального времени. Отчетность и аналитика помогают выявить отклонения от составленного плана и принимать корректирующие меры.

Ключевые слова: бизнес-процессы, управление проектами, технологии управления.

С развитием технологий управления персоналом и проектами на предприятиях различного масштаба увеличивается потребность в выборе методологии и программных средств, позволяющих получить максимальную выгоду и сократить издержки на получение конечного результата. Но не все технологии применимы для разрабатываемого проекта, так как неправильный выбор методологии может привести к убыткам и увеличению срока выполнения поставленных задач. Далее будут рассмотрены различные существующие методологии управления проектами.

Каскадная модель, также известная как Waterfall, является моделью процесса разработки программного обеспечения, при которой линейно и непрерывно выполняются фазы идеи, инициации, анализа требований, дизайна интерфейсов, разработки частей продукта, тестирования полученного результата, запуска продукта для промышленного использования и эксплуатации. Каждая фаза зависит от предыдущей и не может быть прервана. Изменения в проект вносить после начала фазы являются сложным процессом, и они должны иметь оценку изменения бюджета и конечного результата. Клиент получает результат после завершения конкретных итераций [1]. Схема алгоритма действий каскадной модели показана на рисунке 1.

Преимуществами каскадной модели являются структурированный подход:

- каскадная модель имеет конкретную структуру разработки проекта, что упрощает контроль и управление;
- обязательные фазы и роли: каждая фаза имеет четкие требования и ответственную команду за их выполнение, что способствует быстрой организации проекта;
- планирование и бюджетирование: планы и бюджеты заранее определены, что делает управление более простым и предсказуемым.



Рисунок 1 – Схема алгоритма действий каскадной модели [2]

Недостатками каскадной модели являются неспособность справляться с изменениями: если требования или обстоятельства проекта меняются, модель оказывается неэффективной и потенциально требует пересмотра бюджета; длинные по времени циклы разработки: так как проекты выполняются последовательно, каскадная модель требует больших временных затрат и она не подходит для проектов с короткими сроками; ограниченная обратная связь с заказчиком: заказчик видит результаты только в конце фазы, что уменьшает возможность вносить изменения и корректировки в течение всего процесса.

Agile – это методология управления проектами и разработки программного обеспечения, основными принципами которой являются гибкость, итеративность и активная связь с заказчиком. Данная методология подходит для проектов, требующих высокой степени готовности к изменениям и способности адаптироваться к новым требованиям в ходе выполнения проекта. Проекты разрабатываются и отдаются короткими итерациями, что позволяет клиенту заранее увидеть результат и внести изменения в требования при необходимости. Клиент активно участвует в процессе разработки, отдает обратную связь и корректирует требования по мере необходимости. Agile-команды более организованы и сами планируют задачи, которые необходимо взять в разработку [3]. Схема этапов в пределах итерации показана на рисунке 2.

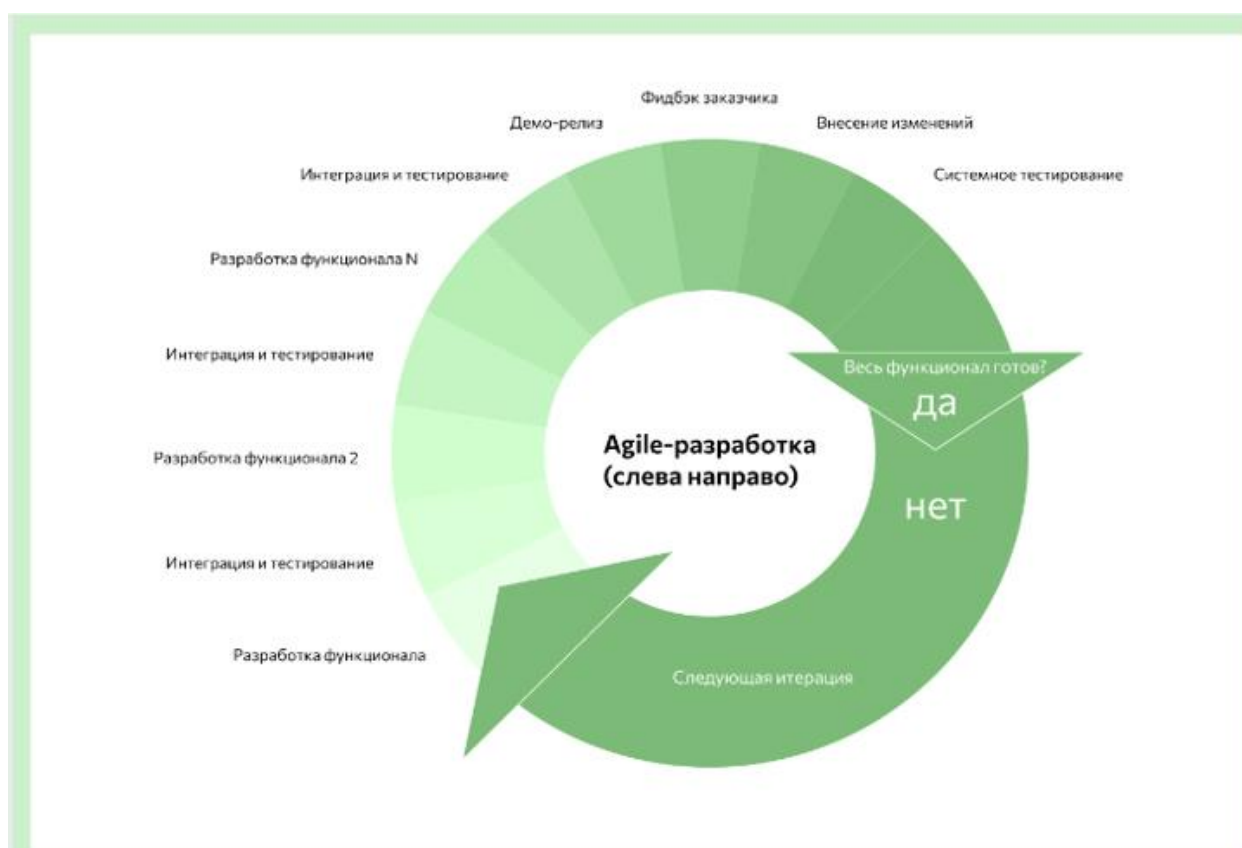


Рисунок 2 – Схема этапов итерации методологии Agile [4]

Преимуществами Agile являются гибкость и адаптивность: данные характеристики позволяют своевременно реагировать на меняющиеся требования и условия на рынке; повышение удовлетворенности клиентов: они вовлечены в процесс разработки и видят результаты выполнения сразу после завершения итерации; снижение рисков: уменьшение количества итераций снижает риски, недостатки можно найти заранее и своевременно исправить.

Недостатками Agile являются сложность управления: требуется координация и управление для обеспечения оперативности выполнения задач, эффективности разработки и качества получаемого продукта; совместимость не для всех проектов: не все проекты подходят для Agile, особенно если требования четко определены; необходимость обучения: команды и предприятия нуждаются в обучении и адаптации для внедрения Agile.

Гибкость и способность к быстрой адаптации являются отличительными свойствами методологий Agile и полезны для проектов, требующих быстрой разработки, частых изменений и взаимодействия с клиентами.

Scrum – это методология управления проектами и разработки программного обеспечения, благодаря которой команда налаживает гибкий рабочий процесс, разработка ведется итерациями, четко обозначаются цели каждой итерации и задачи каждого члена команды.

Состав команд состоит из владельца продукта – работника, который налаживает связь между командой и заинтересованными лицами. Он понимает, что нужно клиентам, контролирует структуру проекта и его цели. Scrum-мастер –

один из членов команды, в задачи которого входит внедрение и укрепление ценностей Scrum на командных совещаниях и поддерживает участников во время выполнения задач. Члены команды – остальные участники Scrum-команды. Все они равноправны и каждый выполняют свою задачу [5]. Структура методологии показана на рисунке 3.

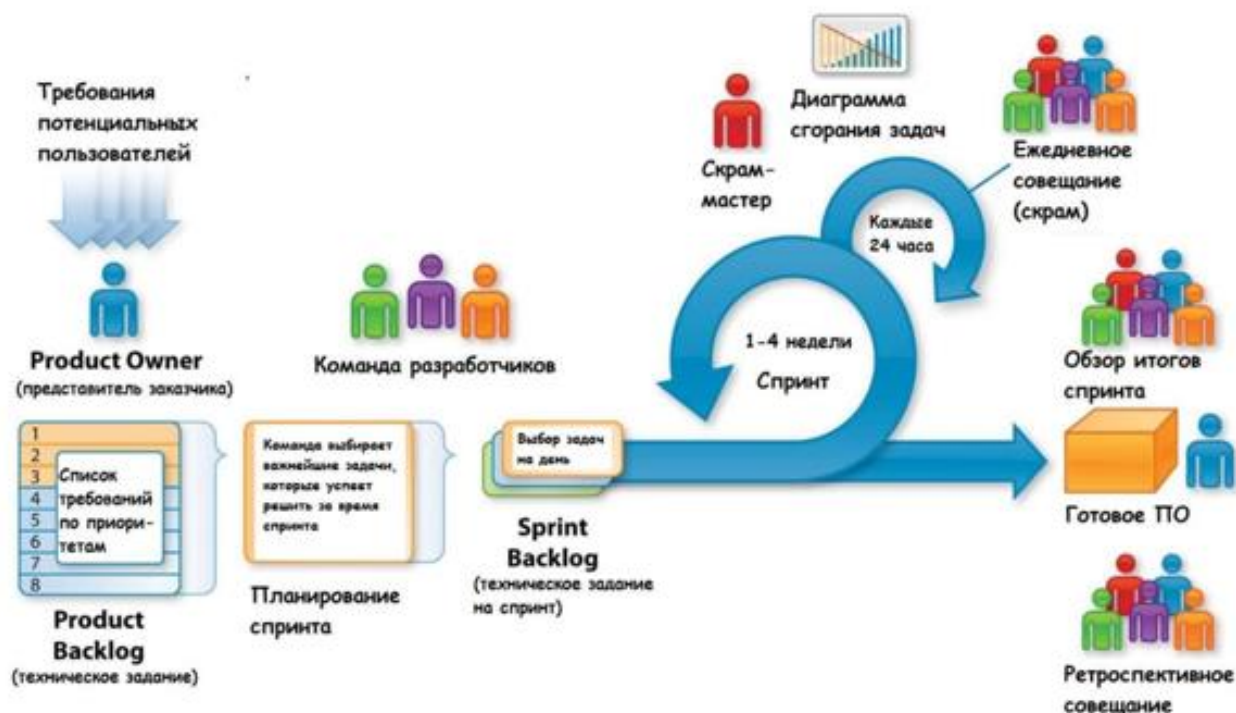


Рисунок 3 – Структура методологии Scrum [6]

Особенности Scrum:

Спринты: проекты делятся на короткие периоды времени, называемые спринтами, которые обычно длятся от двух недель до месяца. Задачи выполняются в течение каждого спринта.

Периодический обзор: после каждого спринта проводится обзор результатов, на котором команда и заказчик обсуждают работу и корректируют планы на будущее.

Бэклог задач: все задачи и требования записываются в бэклог, где заказчик и команда расставляют приоритеты на будущие спринты.

Ретроспектива: анализ итогов спринта вместе с командой, во время которой разбираются ошибки и предлагаются советы по улучшению работы. Общий бэклог продукта актуализируется в зависимости от результатов работы над обновлениями и смены приоритетов у заинтересованных лиц.

Преимуществами Scrum являются быстрое реагирование: более короткие по времени спринты позволяют быстрее реагировать на требования заказчика; прозрачность: регулярные обзоры прогресса делают процессы разработки более понятными для заказчика; вовлечение заказчика: активное участие заказчика в процессы разработки способствует полной реализации требований.

Недостатками Scrum являются неприменимость ко всем проектам: Scrum может оказаться неэффективным для небольших проектов или проектов с частыми изменениями требований; требование опыта: успешное внедрение Scrum требует знаний и опыта в применении Scrum; дополнительное время на обзоры: регулярные обзоры могут добавить дополнительное время к процессу разработки и более поздним срокам выполнения проекта.

Scrum – это мощная методология, которая позволяет командам быстро адаптироваться к изменениям и работать с заказчиками для достижения лучших результатов проекта.

Сегодня на рынке представлено множество решений для управления проектами. Управление проектами имеет множество аспектов, и инструментальные решения играют важную роль в эффективности и успехе проекта. На рынке представлено множество инструментальных решений. Основными продуктами являются Jira, Trello и Asana.

Trello: инструмент основан на концепции доски Kanban. Пользователи могут создавать доски, списки задач и карточки, представляющие отдельные задачи или этапы проекта для разных проектов. Задачи можно перемещать между списками. Trello также поддерживает комментарии, вложения файлов и сроки.

Преимущества Trello:

1. Простота использования: Trello предлагает интуитивный интерфейс, состоящий из досок, списков и карточек, что делает его простым в использовании даже для новичков.

2. Гибкость и настраиваемость: Пользователи могут настраивать информационные панели, создавать списки задач, определять столбцы для каждого этапа выполнения задачи и устанавливать метки и сроки выполнения на карточках.

3. Сотрудничество и командная работа: Trello обеспечивает совместную работу над проектами и задачами. Пользователи могут приглашать коллег на свои доски, комментировать карточки, назначать исполнителей и обмениваться данными.

4. Интеграция с другими сервисами: Trello интегрируется с различными сервисами, такими как Google Drive, Slack и Dropbox, позволяя пользователям связывать задачи и файлы между собой и общаться с коллегами через другие платформы.

5. Бесплатная версия: Trello предлагает бесплатную версию со многими функциями, полезными для команд и проектов.

Недостатки Trello:

1. Ограниченная функциональность для сложных проектов: В отличие от других инструментов, Trello не обладает достаточной функциональностью для некоторых проектов.

2. Не подходит для строгих организаций: несмотря на свою гибкость, Trello не всегда подходит для организации задач в рамках строгого процесса управления проектами.

3. Отсутствие функции учета рабочего времени: В базовой версии Trello отсутствует функционал отслеживания времени, что может быть недостатком для команд, которым необходимо точно знать, когда задачи должны быть выполнены.

Asana: это мощный инструмент управления проектами и задачами, который позволяет пользователям создавать проекты, задачи и подзадачи и устанавливать сроки их выполнения. Он также позволяет назначать владельцев задач, отслеживать ход выполнения, делиться данными в виде комментариев и файлов. Кроме того, он поддерживает создание шаблонов проектов для повторного использования.

Asana обладает следующими преимуществами:

1. Удобный интерфейс и простота использования: Asana обладает интуитивно понятным интерфейсом, что делает его простым в использовании для начинающих пользователей.

2. Гибкость в настройке задач и проектов: Asana позволяет пользователям создавать задачи с различными полями, устанавливать сроки, назначать ответственных, создавать проекты и подпроекты.

3. Отличная интеграция с другими сервисами: Asana интегрируется со многими другими популярными сервисами, такими как Slack, Google Drive, Dropbox и Microsoft Teams. Это позволяет связывать задачи и файлы, а также общаться с командой через другие платформы.

4. Функциональное отслеживание прогресса: Asana предоставляет широкие возможности для отслеживания хода выполнения задач и проектов. Инструмент поддерживает создание диаграмм Ганта, отчетов о проделанной работе, диаграмм мониторинга и других инструментов анализа.

5. Поддержка agile-методологий: в Asana есть функции для управления agile-проектами, такие как создание досок scrum и kanban, настройка итераций и спринтов, а также отслеживание бэклогов и дефектов.

Недостатки Asana:

1. Сложности с большими командами и проектами: Крупные проекты и команды, особенно с большим количеством задач, могут столкнуться с трудностями в использовании Asana из-за ограниченных возможностей по организации информации и перемещению между различными проектами.

2. Высокая стоимость полных версий: хотя Asana предлагает бесплатную версию с базовым функционалом, полная версия со всеми возможностями может быть довольно дорогой, особенно для небольших организаций и стартапов.

Jira: Jira часто ассоциируется с разработкой программного обеспечения и управлением задачами в agile-контексте, но может использоваться и в других областях управления проектами. Jira позволяет создавать задачи, отчеты об ошибках и пользовательские рабочие процессы. Она предоставляет широкие функциональные возможности для настройки рабочих процессов, отслеживания ошибок и управления релизами.

Преимуществами Jira являются:

1. Гибкость и настраиваемость: Jira предоставляет широкие возможности для настройки рабочих процессов, пользовательских полей, типов задач и рабочих процессов. Это позволяет адаптировать Jira к специфическим потребностям различных типов проектов и команд.

2. Интеграция с другими инструментами: Jira легко интегрируется с другими инструментами разработки программного обеспечения, такими как Bitbucket, GitHub и Confluence, обеспечивая более плавные рабочие процессы между различными этапами жизненного цикла проекта.

3. Мощные функции отслеживания: Jira предоставляет широкие возможности для отслеживания задач, сообщений об ошибках, релизов и хода выполнения проекта. Это включает в себя создание диаграмм Ганта, диаграмм отслеживания и настраиваемых отчетов.

4. Поддержка agile-методологий: Jira предоставляет набор инструментов для управления agile-проектами, такими как Scrum и Kanban. Это позволяет командам эффективно планировать и отслеживать итерации, задачи и обновления.

5. Обширное сообщество и поддержка: Jira имеет большое сообщество пользователей и разработчиков, предоставляющее доступ к учебным пособиям, документации и форумам поддержки.

Недостатки Jira:

1. Сложность использования для новичков: из-за своей гибкости и множества функций Jira может показаться сложной для начинающих пользователей. Может потребоваться время, чтобы изучить все функции и настроить систему под свои нужды.

2. Высокая стоимость: Jira – платный инструмент, и его стоимость может быть довольно высокой для небольших команд и стартапов. Кроме того, могут потребоваться дополнительные расходы на некоторые дополнительные функции и интеграцию с инструментом.

3. Не всегда подходит для небольших проектов: благодаря своей мощности и расширенным функциям Jira может оказаться излишней для небольших проектов или команд, которым нужен более простой инструмент управления задачами.

Данное исследование показало, что выбор конкретной методологии и инструментов управления стоит выбирать исходя из требований проекта, при сравнении методологий нет варианта, который будет единственно верным для выбора. Ни один из инструментов не предоставит весь необходимый функционал для выполнения задач и отслеживания прогресса.

Список использованных источников:

1. Waterfall Методология разработки [Электронный ресурс]. – Режим доступа: <https://qaevolution.ru>. – Дата доступа: 12.04.2024.

2. Классика Waterfall [Электронный ресурс]. – Режим доступа: <https://worksection.com/images/upload/Waterfall.jpg>. – Дата доступа: 12.04.2024.

3. Agile [Электронный ресурс]. – Режим доступа: <https://scrumtrek.ru/>. – Дата доступа: 13.04.2024.

4. Agile [Электронный ресурс]. – Режим доступа: <https://blog.skillfactory.ru/wp-content/uploads/2023/02/agile-1.png>. – Дата доступа: 13.04.2024.
5. Гибкая методология разработки «Scrum» [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/247319>. – Дата доступа: 13.04.2024.
6. Коротко о Scrum [Электронный ресурс]. – Режим доступа: <https://worksection.com/images/upload/scrum.jpg>. – Дата доступа: 13.04.2024.