

Основные особенности методологии Agile

Семенова Анастасия Сергеевна, студент
Самарский государственный технический университет
Пальмов Сергей Вадимович, кандидат технических наук, доцент
Поволжский государственный университет телекоммуникаций и информатики (г. Самара),
Самарский государственный технический университет

В статье кратко рассмотрена гибкая методология разработки (Agile). Приведена история ее возникновения, область применения, достоинства и недостатки. Также произведено ее сравнение с каскадной моделью (Waterfall).

Ключевые слова: Agile, гибкая разработка, Waterfall, методология, XP, Scrum, Lean.

Небольшие коллективы разработчиков программного обеспечения (ПО) стали применять метод Agile примерно полтора-два десятилетия назад. Однако услуги создания сложных коммерческих программных продуктов доступны на рынке уже около полувека. Этим же временным интервалом датируется практическое внедрение первых методологий управления проектами. В частности, летом 1970 года вышла статья сотрудника Lockheed Y. Ройса (W.W. Royce) «Managing the development of large software systems» [1]. Указанная работа содержит критику господствовавшего на тот момент подхода последовательной разработки программного обеспечения. Автор утверждал, что необходимо учитывать специфику этой новой сферы деятельности и не приравнивать ее к конвейерной сборке, применяемой на промышленных предприятиях. Требуется совершенно иной подход.

Вместо ожидания окончательного завершения каждой отдельной фазы (этапа), Ройс предложил применять на практике phase approach (фазовый подход). Суть его такова: разработчики предварительно собирают всю необходимую для работы с проектом информацию, заранее описывая все условия, после чего происходит реализация архитектуры, разрабатывается дизайн, пишется код и т.д.

Ройс задал темп для развития абсолютно новых методов создания проектов. Все эти методы можно объединить в единый класс — гибкие методы разработки (agile software development).

Не все проекты можно реализовать с помощью классических методик. Именно поэтому в последнем десятилетии прошлого века были созданы различные варианты гибких методологий для разработки ПО, способных упростить сложные и трудоемкие процессы проектирования.

Позже, в 2001 году, в штате Юта был собран совет из семнадцати разработчиков ПО [2]. В ходе анализа различных способов реализации проектов на съезде, был сформулирован и опубликован «Agile Manifesto» (Agile-манифест). В данном манифесте описаны основные принципы гибкой методологии, а также ее ключевые ценности. Именно он задал направление для дальнейшей работы над созданием ПО.

Agile — это принцип организации создания нового продукта или услуги. Основная идея может быть сформулирована так: каждый участник процесса создания проекта вовлекается в общее дело [3]. Все участники команды разработчиков, а также заказчик проекта, имеют возможность внести свои коррективы и предложения, и даже полностью остановить работу по созданию проекта. Благодаря этому, все участники разработки выполняют свою работу наиболее качественно, осознавая, что они работают над общей

целью — созданием высококачественного и простого в использовании продукта.

Основным принципом Agile-методологии является визуальное управление. Для поддержания связи между участниками команды, работающих над проектом, используются таблички различных оттенков. Каждая табличка имеет свое уникальное значение и сигнализирует другим членам команды о текущем этапе проектирования. Так, один цвет дает сигнал о завершении планирования определенного элемента конечного продукта, другой — о завершении его разработки, третий — о готовности и т.п. Визуальное управление упрощает взаимодействие между членами команды, а также предотвращает возникновение разного рода недопониманий.

Суть Agile-подхода можно коротко сформулировать так [4]:

- реализация происходит циклами (итерациями), длительностью не более четырех недель;
- в финале каждого этапа клиент получает готовое приложение (или его часть), которое можно использовать в бизнесе;
- по ходу всей разработки проекта команда взаимодействует с клиентом;
- любая новая информация моментально принимается на рассмотрение.

Agile предполагает, что при создании проекта необходимо опираться не только на заранее разработанные планы. Необходимо, чтобы команда разработчиков умела быстро определять дальнейший ход действий несмотря на изменчивость внешней и внутренней среды, а также быстро откликаться на ответную реакцию от заказчиков и потребителей. Это помогает разработчикам находить новые решения для одних и тех же задач, а также экспериментировать, тем самым делая проект индивидуальным и современным, не ограничивая себя жесткими рамками и стандартами, а также дает возможность устранять недопонимания на начальном этапе разработки.

Agile изначально возник в IT-среде, но позже методология распространилась и в другие сферы — от промышленной инженерии до искусственного интеллекта.

К наиболее известным гибким методологиям можно отнести следующие [5]:

1. Scrum — система гибкой разработки проектов, основанная на принципе спринта. Рабочая версия продукта изготавливается в краткие сроки (не больше месяца).
2. Экстремальное программирование (XP) — метод, при котором взаимодействие между клиентом и разработчиками происходит постоянно. На каждом этапе команда и заказчик обсуждают проделанную работу, вносят коррективы. Этот подход позволяет выявить недостатки продукта

до окончания разработки, а также определить необходимый функционал и другие параметры.

3. Lean — основывается на системе управления производством. Отличительной чертой является принцип постоянного повышения качества продукта на всех уровнях организации процесса. Lean сочетает в себе гибкость и структурированность, что помогает сделать выполнение проекта более четким и качественным.

Можно с уверенностью сказать, что гибкая модель подойдет для осуществления проектов в любой сфере деятельности, она является универсальной. Сложность принятия решения по выбору модели заключается только в ограничениях заказчика по длительности разработки и дороговизной. Качество обратной связи при данном подходе поможет прийти к согласованному решению. Выбирая эту модель, заказчик может быть уверен, что его проект будет единственным в своем роде, интересным и проверенным до мельчайших деталей.

Таким образом, ключевыми преимуществами Agile являются [6]:

1. Качество продукта. Вовлечение заказчика в процесс каждой итерации дает возможность вносить корректировки в процесс разработки проекта на отдельном этапе, что безусловно улучшает результат.

2. Высокая скорость разработки. Итерация занимает менее одного месяца, к концу этого периода обязательно есть готовый результат.

3. Минимизация рисков. Крупный проект дает возможность заказчику оплатить несколько итераций, и в ходе работы понять, что он вовремя получит именно то, что хочет и за приемлемую цену, а также позволяет корректировать в ходе работы некоторые детали проекта. Ни одна другая методология таких возможностей не предоставляет.

Таблица 1. Сравнение ключевых идей Agile и Waterfall методологий [7].

Agile	Waterfall
Люди и их взаимодействие	Процессы и инструменты
Готовый проект	Документация
Сотрудничество с заказчиком	Жесткие контрактные ограничения
Реакция на изменения	Следование сценарию

Одним из подвидов Waterfall считается поэтапная модель с промежуточным контролем. Она подразумевает проектирование ПО с прямой последовательностью стадий, но в нескольких версиях, т.е. с плановой модернизацией продукта во время всего жизненного цикла разработки. Проектирование ПО ведется итерациями с обратной связью между фазами. На первом этапе разработки проекта определяются ключевые требования по приоритетам, далее выполняется реализация системы по принципу приращений, начиная с более приоритетного компонента. Когда все части системы определены, переходят к детализации первой части, параллельно согласовывая требования для детализации других частей с заказчиком. ПО реализуют по частям, по готовности первой части ее предоставляют клиенту, что позволяет сформировать у него требования к следующим частям. Совершенствование модели происходит до тех пор, пока все части ПО не будут реализованы окончательно. Данная модель проектирования отлично подойдет для сложных и комплексных систем, с четким пониманием конечного результата.

4. Клиент всегда имеет возможность наблюдать за процессом реализации, проверять и запускать проект, а также может остановить разработку на любом этапе.

Однако Agile-методология имеет и ряд минусов:

1. Повышенные требования к разработчику и заказчику — реализовать проект высокого качества без постоянной связи между двумя сторонами практически невозможно. Из-за большого выбора методов и инструментов в Agile для качественной реализации требуется опытная команда разработчиков.

2. Участники должны иметь возможность взаимодействовать лично как между собой, так и с заказчиком проекта. Не подходит для реализации проектов, где стороны взаимодействуют только онлайн.

3. Вероятность никогда не выпустить итоговую версию проекта — этот недостаток, как не парадоксально, вытекает из возможности редактирования продукта непрерывно.

4. Не работает без четкого видения бизнес-целей проекта. Поскольку Agile-команда ориентируется на партнеров по бизнесу, клиентов, собственников и инвесторов, то без формулировки четких целей и концепции продукта разработка невозможна.

На данный момент единого (универсального) метода создания проектов не существует. Все подходы в корне различаются по времени и стоимости реализации, а также по возможности исправления недочетов в ходе итераций.

Метод реализации подбирается исходя из идей бизнеса, объема работ, сроков выполнения и бюджета.

Помимо Agile, не менее популярной считается Waterfall. Agile и Waterfall показывают методику разработки управления проектами абсолютно с разных сторон. Обе методологии являются основой для множества модификаций, направленных на конкретный формат проекта. Сравним идеи данных методологий в табл. 1.

Agile-модель отлично подойдет для небольших компаний в сфере IT, стартапов и новых проектов. Несмотря на все преимущества гибкой модели, каскадная модель не уступает своих позиций в сфере строительства, а также в ситуациях, где важным фактором является срок реализации, а не финансы.

Эффективную методологию заказчик сможет определить и самостоятельно, исходя из области бизнес-проекта, необходимых сроков выполнения работы, а также исходя из необходимых критериев риска и вовлеченности заинтересованных лиц.

Итак, проведя анализ Agile, можно сделать вывод, что применение гибких методов позволяет реализовать наиболее эффективный процесс разработки программного обеспечения. Agile можно легко определить по трем характеристикам: минимизация рисков, работа в коротких циклах, коммуникация между разработчиками и заказчиками. Несмотря на все свои плюсы, методология достаточно сложна в реализации и ее практически невозможно реализовать группе начинающих разработчиков в сжатые сроки.

www.esa-conference.ru

Литература:

1. Бентли Д. Жемчужины программирования. М.: Питер, 2016. – 272 с.
2. Вольфсон Б. Гибкое управление проектами и продуктами. М.: 2017. – 300 с.
3. Книберг Х. Scrum и XP: заметки с передовой [Электронный ресурс] URL: http://scrum.org.ua/wp-content/uploads/2008/12/scrum_xp-from-the-trenches-rus-final.pdf (дата обращения: 02.12.2018).
4. Кон М. Scrum: гибкая разработка ПО. М.: Вильямс, 2014. – 576 с.
5. Лайкер Д. Дао Toyota. 14 принципов менеджмента ведущей компании мира. М.: 2013. – 400 с.
6. Листер Т. Вальсируя с Медведями: управление рисками в проектах по разработке программного обеспечения. М.: 2015. – 196 с.
7. Мартин Р., Ньюкирк Д, Косс Р. Быстрая разработка программ. Принципы, примеры, практика. М.: Вильямс, 2014. – 752 с.