

ГИБКАЯ МЕТОДОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (AGILE)

С.С. Сорокин

sorokin.sns@yandex.ru

А.М. Сорокина

maximova-alisa@mail.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

В настоящее время гибкая методология разработки программного обеспечения, также известная как Agile, стала одной из самых популярных и востребованных в индустрии информационных технологий. Вместо традиционного и жесткого подхода к разработке Agile предлагает итеративный и гибкий метод, который позволяет командам разработчиков действовать более эффективно и адаптироваться к изменениям в требованиях проекта. Эта статья рассмотрит основные принципы и методы Agile, а также преимущества, которые она может принести для команд разработки программного обеспечения.

Ключевые слова

Agile, гибкая методология, программное обеспечение, итеративный подход, инкрементальная разработка, скрам, экстремальное программирование, канбан

Поступила в редакцию 03.11.2023

© МГТУ им. Н.Э. Баумана, 2023

Введение. В современном быстро меняющемся мире программного обеспечения (ПО) требуется гибкий и адаптивный подход к разработке продукта. Традиционные методологии разработки больше не могут эффективно справляться с постоянно меняющимися требованиями рынка и ограничениями проектов. В таком контексте гибкая методология разработки ПО, известная как Agile, рекомендовала себя как эффективный и устойчивый подход.

Статья посвящена гибкой методологии разработки ПО и ее ключевым концепциям. Рассмотрим, как Agile изменяет подход к разработке, фокусируясь на обеспечении быстрой адаптации к изменениям требований и построении гибкой рабочей модели. Также представим основные методы Agile, такие как Lean, Kanban и Scrum, и изучим их применение в различных сценариях разработки.

Основой методологии Agile служат коллективная работа и постоянное сотрудничество между разработчиками, заказчиками и заинтересованными сторонами. Вместо детального планирования и долгих циклов разработки Agile предлагает частую поставку небольших инкрементальных изменений, позволяя вносить коррективы и отвечать на меняющиеся требования клиента.

Цель Agile заключается в достижении высокой гибкости, адаптивности и реактивности ПО, что позволяет компании более успешно оперировать в современной динамичной и конкурентной среде. В результате Agile не только улучшает эффективность и качество процесса разработки ПО, но и позволяет

достичь большей удовлетворенности клиента и более высоких бизнес-результатов. Будучи на переднем крае разработки ПО, Agile выходит за рамки простой методологии и становится культурой сотрудничества, инноваций и постоянного совершенствования.

Цель этой статьи — предоставление четкого понимания принципов и преимуществ Agile как методологии разработки ПО. Также будут рассмотрены ее ограничения и факторы успеха внедрения. В конце статьи планируется подвести итоги и представить дополнительные рекомендации для успешного применения Agile в организации бизнес процессов.

Природа Agile. Разработка ПО, как и любая другая техническая дисциплина, сталкивается с проблемами в области качества, стоимости и надежности результатов разработки. В связи с этим правильная организация процесса разработки ПО лежит в основе достижения запланированного результата в ожидаемые сроки, с ожидаемым уровнем качества и с адекватным бюджетом.

Среди общераспространенных проблем процесса разработки ПО встречаются следующие:

- 1) изменение требований непосредственно в процессе разработки;
- 2) нечеткое распределение ответственности за выполняемую работу и ее результат;
- 3) наличие непрерывного потока мелких, «быстрых», наваливающихся требований, отвлекающих разработчиков и менеджеров от основного направления работ;
- 4) как следствие, срыв сроков, раздувание бюджетов, потеря качества [1].

Для решения задачи успешной организации процесса разработки ПО была создана гибкая методология. Гибкая методология разработки (англ. *Agile software development*) — это набор принципов и правил, в рамках которого осуществляется разработка ПО. Методология Agile — это семейство процессов разработки, а не единственный подход к разработке ПО [1]. Agile не включает конкретных практик, а определяет ценности и принципы, которыми руководствуются успешные команды.

Слово *agile* переводится как «подвижный», «проворный», «легкий», т. е. метод предполагает быстрое движение и реагирование. Многие специалисты в области гибких методологий считают Agile даже не фреймворком, а философией. Однако если говорить о методологии в рамках подхода к управлению, то она сочетает следующие принципы работы:

- 1) фокусирование на нуждах клиентов;
- 2) упрощение организационной структуры и процессов;
- 3) работу короткими циклами;
- 4) активную обратную связь;
- 5) повышение полномочий сотрудников;
- 6) гуманистический подход в управлении.

В Agile на нуждах и интересах клиентов фокусируется не только заказчик, но и вся команда, работающая над созданием продукта или услуги. Каждый участник четко понимает, кто такой клиент, с какими проблемами он столкнулся и как он хотел бы их решить. Подобный подход и фокусировка позволяют создать более качественные решения. Упрощение организационной структуры позволяет сотрудникам думать не о регламенте и правилах, а о своей работе и ценности, которую они создают. Во избежание создания неактуального продукта важно работать короткими циклами. В Agile такой подход носит название итеративно-инкрементального: работа происходит небольшими фиксированными периодами, например, от двух до четырех недель [2]. В конце каждого периода команда представляет заказчику работоспособный вариант продукта, которым уже можно пользоваться. Например, редактор газеты решил перейти в онлайн-пространство и создать сайт газеты. Его задача состоит в том, чтобы в первые две недели организовать для пользователей сайт с минимальным набором функций и рубрик. Еще через один цикл работы на сайте появится больше рубрик и поисковое окно, позже — прогноз погоды и т. д. В конце каждой итерации пользователь получает рабочий сайт, однако вместе с тем с каждой итерацией продукт улучшается. Преимуществом такого подхода является не только ранний выход на рынок, опережение конкурентов, но и возможность внесения изменений внутри каждого рабочего цикла.

Активная обратная связь важна для любого процесса, ведь она позволяет вовремя скорректировать вектор работы, удалить из создаваемого продукта ошибки, тем самым свести потери к минимуму. Во многих областях человеческой деятельности, которые связаны с созданием чего-то нового, работа ведется с активным экспериментированием. Для Agile эксперимент — обычная практика, используемая системно и в создании продукта, и в построении процессов, и даже во взаимоотношениях в команде. Вместе с тем важно анализировать и обрабатывать полученную в ходе испытаний информацию, чтобы вовремя остановить процесс, обратить внимание на ошибки, исправить их и пойти дальше.

Причин, по которым следует наделять членов команды большими полномочиями, достаточно много. Выделим основные:

1) занятые умственным трудом люди не любят, когда у них отнимают возможность самостоятельно принимать решения. Таких сотрудников подобные ситуации демотивируют;

2) получая больше полномочий, работник вынужден учиться принимать решения и нести за них ответственность. В таком случае работа над продуктом не остановится, если самоорганизованная команда столкнется с неизвестной ранее проблемой;

3) подобное отношение к работе сокращает время принятия решений [1].

Манифест Agile. В начале 2001 г. в городе Сноуберд, штат Юта, собрались 17 человек, чтобы обсудить будущее разработки программного обеспечения.

Участников этой группы объединяло беспокойство по поводу текущего положения дел в отрасли. При этом их не пугало, что все они по-разному представляли оптимальное решение.

Они сошлись во мнениях об основной проблеме: компании настолько сосредоточены на избыточном планировании и документировании своих циклов разработки ПО, что забыли о главном — о том, что нужно приносить радость клиентам.

Навязывая корпоративные ценности, такие как «мастерство» и «добросовестность», компании почти не помогали людям (особенно разработчикам ПО) повысить эффективность работы. Это нужно было менять. У многих участников группы Snowbird 17 уже были идеи по поводу того, как открыть новую эру разработки ПО. Поездка в горы позволила им это обсудить [2].

Результатом длинных выходных стал Манифест Agile. Этот краткий и выразительный документ состоял всего из 68 слов и навсегда изменил разработку программного обеспечения. За почти два десятилетия, прошедшие с момента его создания, эти слова (и 12 последовавших принципов) были приняты (в той или иной степени) огромным количеством людей, команд и компаний. Этот манифест состоит из пяти пунктов:

- 1) люди и взаимодействия важнее процессов и инструментов;
- 2) работающее ПО важнее исчерпывающей документации;
- 3) сотрудничество с заказчиков важнее согласования условий контракта;
- 4) готовность к изменениям важнее следования первоначальному плану;
- 5) т. е., не отрицая важности того, что справа, мы все-таки больше ценим то, что слева.

Кроме того, представители Agile-альянса составили еще один документ, а именно «Основополагающие принципы Манифеста Agile» [2].

1. Приоритетом данного манифеста является удовлетворение потребностей заказчика благодаря регулярной и ранней поставке ценного программного обеспечения.

2. Изменение требований приветствуется даже на поздних стадиях разработки. Agile-процессы позволяют использовать изменения для обеспечения заказчика конкурентного преимущества.

3. Работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев.

4. На протяжении всего проекта разработчики и бизнес-представители должны ежедневно работать вместе.

5. Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, нужно создать условия, обеспечить поддержку и полностью довериться им.

6. Непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой командой, так и внутри нее.

7. Работающее программное обеспечение — основной показатель прогресса.

8. Инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно. Agile помогает наладить такой устойчивый процесс разработки.

9. Постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта.

10. Простота — искусство минимизации лишней работы — крайне необходима.

11. Самые лучшие требования, архитектурные и технические решения рождаются у самоорганизующихся команд.

12. Команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы.

Читая эти строки, можно сделать вывод, что в то время как жесткие методологии управления направлены на сохранение «треугольника управления проектами», в который входят затраты, содержание и сроки, гибкие методологии изначально ориентируются на создание качественного продукта, исключая беспокойство о бюджете и сроках (рис. 1) [3].



Рис. 1. Треугольник управления проектами

Методологии Agile. Agile по своей сути — это «зонт» (рис. 2), под которым находятся иные методологии. Наиболее известные из них Lean, Kanban, Scrum и Waterfall. Рассмотрим их.

Lean. Методология Lean считается предшественником Agile и построен на бережливом управлении. Работа такого подхода, с одной стороны, крайне проста. Сначала находят барьеры, которые мешают росту и конкурентоспособности компании. После их оценивают: как преодолеть, обойти, изменить. При этом планируют не однократное действие, а длительный процесс, в который вовлечены как владельцы бизнеса, так и рядовые работники. Lean строится на созда-

нии ценностей в компании и достижении их с одновременным уменьшением производственных издержек. Под ценностью имеют в виду не только товар или услугу, которую желает приобрести потребитель, а процесс производства, т. е. от поставщика сырья до получения продукта покупателем. Принципы такого подхода:

- 1) процессы, создающие ценность, необходимо постоянно оптимизировать;
- 2) процессы, не создающие никакой ценности, но при этом нужные, требуются сократить до полного минимума;
- 3) потери нужно исключить полностью.



Рис. 2. «Зонт» Agile

Таким образом, при использовании lean-менеджмента, т. е. «бережливого производства», работа компании делится на операции и процессы, главная задача которых — либо создание ценности, либо ее исключение. Задачей всего управления становится планомерное исключение тех процессов, которые не добавляют ценности для потребителя [4].

Kanban. Данная методология относится к «бережливому управлению». Впервые метод был применен в автомобильной компании Toyota в 1960-е годы. В переводе с японского *кан-бан* — это «сигнальная доска» или «вывеска» (рис. 3): «В производстве такая доска используется для визуализации нарастающего темпа, что позволяет давать больше продукции. Сотрудники на каждом этапе процесса не могут перейти к следующей фазе работы, пока посредством канбандоски не будет дан соответствующий сигнал». Систему Kanban удобно использовать, чтобы достичь баланса между нагрузкой на команду и ее способностью выполнять задачи.

В свойства данного метода входят:

- визуализация рабочего потока;
- определение рабочей нагрузки;

- контроль рабочего процесса;
- конкретизация рабочего процесса;
- совместная работа.

Важно начать с представления процесса работы от начальной точки — «выполнить» до конечной — «выполнено». На kanban-доске также может присутствовать область со значением «в процессе». Такая методика позволяет не только контролировать все этапы, но и разбить большую задачу на подзадачи, выполнять работу по мере поступления, а не делать все и сразу, что с большой вероятностью заканчивается неудачей. Кроме того, в некоторых компаниях практикуют добавление колонки с названием «Идеи» для записи возможных задач.

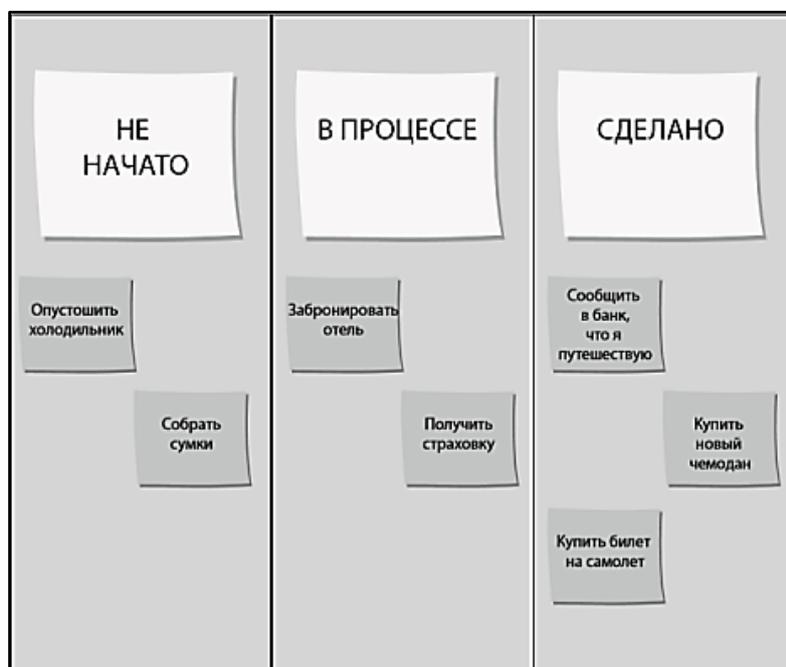


Рис. 3. Пример kanban-доски

Отличительное свойство Kanban заключается в том, что он не влечет огромных изменений внутри организации, а отталкивается от уже существующих принципов работы. Эта система помогает понять, как решить проблему управления изменениями. «Kanban дает инструменты, которые позволяют объяснить (и оправдать), почему разнообразие — это хорошо и выбрать его — значит поступить правильно» [4]. Именно поэтому Kanban может быть использован как целой командой в рамках масштабного проекта, так и отдельным человеком для контролирования выполненных работ. Три основополагающие принципа Kanban:

- 1) определитесь с постановкой задачи;

- 2) выработайте последовательные этапы задачи;
- 3) следуйте согласованным процессам, ролям, обязанностям и условностям [4].

Scrum. Scrum фреймворк начал свое развитие как технология разработки в 1980-х годах в Японии, но известность получил в 1990-е годы в США. «Нет плохого способа начать работать со Скрамом — главной ошибкой будет вообще не начинать», — пишет один из идеологов метода Стэнли Э. Портни.

Scrum — это термин из игры регби, переводится как «схватка». Подобный метод игры требует собранности команды, полного взаимодействия ее членов. Основа методологии заключается в объединении принципов «бережливого управления»: «Когда бы ни был запущен проект, вам ничто не мешает регулярно проверять ход работ и последовательно выяснять: справляетесь ли вы с заданием; в нужном ли направлении движетесь; создаете ли именно то, что на самом деле хочет получить заказчик; есть ли способы усовершенствовать методы разработки и выполнять работу наиболее качественно и быстро». Этот процесс один из авторов методологии Джеф Сазерленд назвал «проверять и адаптироваться», т. е. в любой момент нужно быть готовым сделать паузу в работе и провести анализ, все ли проходит так, как надо. При этом если процесс работы или продукт не соответствуют ожиданиям, следует проявить гибкость и изменить ход работ. «Конечным результатом применения методологии Scrum являются команды, наглядно увеличивающие свою производительность» [4]. Можно сделать вывод, что Scrum — скорее не методология управления проектами, а собрание принципов организации создания и выпуска продукта. Важно понимать, что, в отличие от методологий, описанных выше, в Scrum собирается новая команда, главные роли в которой отводятся владельцу продукта и scrum-мастеру.

Владелец продукта, или Product Owner, самостоятельно решает, как будет выглядеть конечный продукт и в чем заключается суть проекта. Он единственный в команде, кто четко понимает правила бизнеса, в котором работает. Чаще он выступает в роли основного спонсора, но также является человеком, представляющим интересы потребителей. «Хороший владелец продукта всегда заботится об извлечении максимальной выгоды, представляя интересы как бизнесмена, так и конечного потребителя и гарантируя пользу для обоих. В идеале владелец продукта должен быть маяком для скрам-команды, но, когда что-то идет не так, в этом нередко вина именно Владельца продукта».

Scrum-мастер отвечает за всю работу коллектива. Он является связующим звеном между самой командой и владельцем продукта. Но основной задачей scrum-мастера является инструктирование выше названных в работе гибких подходов в управлении: «Если владелец продукта — это мозг проекта, то scrum-мастер отвечает за ежедневное функционирование проекта, создавая наилучшее окружение для работы команды» [5].

Команда разработки — основной двигатель в работе. Участники обладают смежными навыками, что позволяет подхватывать работу друг друга в случае

стагнации проекта. При этом в состав команды в Scrum входит от 6 до 8 человек, поскольку считается, что чем больше людей, тем сложнее ими управлять, и это плохо для проекта. «Собирать огромную команду неэффективно и нерационально. Общение, отношения и, следовательно, работоспособность команды страдают, если она слишком велика» [3].

В ключевые scrum-события входят:

1) спринт — общий цикл для остальных событий, длится от одной до четырех недель.

2) планирование спринта — происходит в самом начале работы;

3) «дейли скрам», или ежедневна летучка — происходит каждый день без исключений и длится не более 15 минут. Здесь звучат главные вопросы: «Что вы делали вчера?», «Что вы будете делать сегодня?», «Какие затруднения у вас возникают?» [5];

4) обзор итогов — проводится в конце спринта и позволяет команде показать проделанную над продуктом работу Владельцу продукта;

5) ретроспектива — обсуждение прошедшей работы, трудностей, с которыми столкнулась команда.

Методологии, перечисленные выше, не являются «панацеей» и подходят далеко не для всех проектов. К примеру, строительство дома не подвергается гибкому управлению, поскольку это четко разработанный план. Мы не можем построить три этажа, получить претензию заказчика, которому не понравилось, как вырыт котлован, все снести и начать заново. Это будет просто неразумно. Кроме того, переход на такие методологии только на первый взгляд кажутся простой задачей. Для того чтобы понять, как работают Lean, Kanban и в особенности Scrum, многие проходят специальное обучение и практикуются годами.

Waterfall. Чаще всего Agile сравнивают с Waterfall (англ. *водопад*), или каскадной моделью (рис. 4). Данная методология используется и сейчас, однако является одной из самых «старых»: впервые о ней заявил в 1970 г. американский ученый в области информатики Уинстон Уокер Ройс. Главные характеристики каскадной модели — последовательность и ориентированность на требования. Для ее применения владельцу продукта и команде требуется иметь четкое представление о том, каким должен быть проект, ведь когда он уже будет в процессе, скорректировать курс не представится возможным. Каждый этап работы автономен, это означает, что для начала выполнения следующего шага необходимо завершить предыдущий.

Методология Waterfall имеет несколько преимуществ:

1) простота применения: благодаря интуитивно понятной системе работы данного метода его можно легко освоить и внедрить, даже не имея опыта;

2) структурированность: жесткость Waterfall может быть и недостатком, однако она позволяет четко разделить и организовать работу в виде этапов. По-

сколькx нельзя вернуться к завершенной работе, необходимо сразу идеально ее выполнить.

Вместе с тем в число ее недостатков входят:

1) высокий уровень риска: он обусловлен именно жесткостью рамок и выполнения. Если понадобится внести изменения в проект, придется начинать планирование нового;

2) сложный первый этап: вся методология построена на требованиях, поэтому важно еще в самом начале проекта правильно их понять и изложить. По этой причине Waterfall не подходит для долгосрочных проектов.

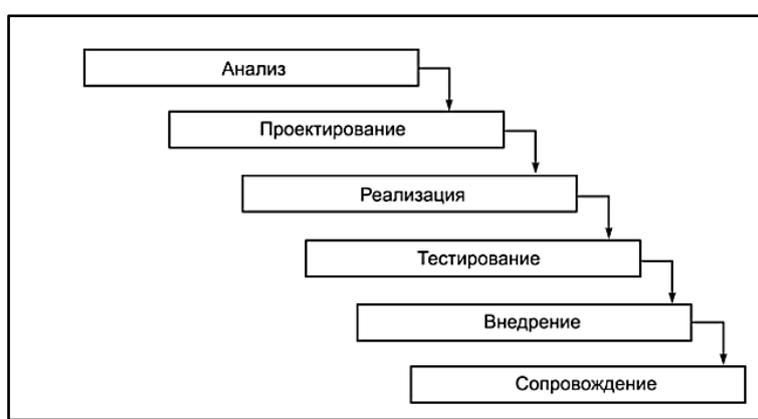


Рис. 4. Схема работы методологии Waterfall

Традиционные и гибкие методологии вовсе не исключают друг друга. Благодаря этому появились так называемые гибридные методологии, например, гибрид Agile и Waterfall, Scrum и Kanban (Scrum-Ban) или Agile и PMBOK. Данный феномен возник в процессе конкурентной борьбы между представителями гибких и классических методологий. Он включает в себя принципы как первых, так и вторых. Agile позволяет решать приоритетные задачи, находить и исправлять ошибки быстрее благодаря спринтам, своевременно реагировать на изменения рынка. Waterfall, в свою очередь, подходит для интеграционных проектов, состоящих из нескольких направлений работ. Фактически в выборе лучшей методологии следует отталкиваться от размера, цели и сложности проекта. В планировании гибридный метод основывается на классической декомпозиции работ, т. е. составлении иерархической структуры работ (ИСР). Если проект включает собой несколько направлений, каждое из них представляется в виде своей ветви ИСР. Это связано с тем, что разработка интеллектуального продукта может быть выполнена по принципам Agile, а работы по поставке оборудования или обучения конечных пользователей продукта — по принципам Waterfall. Следуя классическому построению ИСР, проект разбивают на «пакеты», срок выполнения каждого — от двух до трех недель. Реализуя гибкое

управление, можно разбить проект на двухнедельные пакеты, то есть спринты, по выпуску инкрементов.

Базовым для адаптации гибких методологий категории Agile для работы современного средств массовой информации является понятие проекта. Проект направлен на создание уникального продукта, услуги или результата. Окончание проекта может быть обусловлено несколькими причинами: цели проекта были успешно достигнуты, достичь цели проекта невозможно, заказчик прекращает работу из каких-либо побуждений [6].

Отличительная черта проекта — уникальность создаваемого продукта, услуги или результата. Управление проектом осуществляется путем обязательного выполнения 47 процессов, сгруппированных в пять этапов: инициация, планирование, исполнение, мониторинг, контроль и закрытие [7]. Обязательным на стадии планирования проекта является составление ИСР, или иерархической структуры работ. Она представляет собой декомпозицию больших этапов на мелкие пакеты работ, которые можно выполнить в течение одной итерации. В новостной редакции могут использоваться жесткие, гибкие и гибридные методологии управления проектами. В ряд наиболее известных методологий, в контексте которых существует Agile, входят:

- Lean,
- Scrum,
- Waterfall,
- Scrum-Ban и другие.

Гибридные методологии возникли как реакция на конкуренцию между Agile и жесткими методологиями управления. При этом стоит принять во внимание, что правильное использование подобной интеграции позволит продвигаться по проекту вдвое быстрее, чем при использовании гибких и жестких методологий по отдельности. Степень преобладания одной из них можно регулировать в соответствии с задачами проекта.

Заключение. В данной статье рассмотрены гибкая методология разработки программного обеспечения, известная как Agile, и ее значимость в сфере разработки бизнес-процессов. Agile представляет собой эффективный и устойчивый подход к разработке программного обеспечения, который позволяет компаниям быстро адаптироваться к изменяющимся требованиям рынка и доставлять ценность своим клиентам.

Agile привносит в разработку новый уровень гибкости, прозрачности и непрерывного улучшения. Эта методология позволяет создавать инкрементальные изменения и быстро реагировать на обратную связь пользователя. Agile способствует более эффективной коммуникации, лучшему планированию и более высокому качеству программного обеспечения.

Применение Agile обеспечивает значительные преимущества, включая улучшение операционной эффективности, повышение безопасности и снижение рис-

ков. Он также способствует быстрому проектированию новых продуктов, внедрению инноваций и более гибкому реагированию на изменяющиеся рыночные требования. Вместе с тем реализация Agile не всегда проста и может потребовать изменений в культуре и организационной структуре компании. Необходимо обеспечить гибкое управление проектами, активное вовлечение всех заинтересованных сторон и непрерывное улучшение процесса разработки [8].

Литература

- [1] *Agile-манифест разработки программного обеспечения*. URL: <https://agilemanifesto.org/iso/ru/manifesto.html> (дата обращения 17.10.2023).
- [2] Штайн Э. *Философия Lean. Бережливое производство на работе и дома*. URL: <http://flibusta.is/b/424958/read> (дата обращения 17.10.2023).
- [3] Коул Р., Скотчер Э. *Блистательный Agile: гибкое управление проектами с помощью Agile, Scrum и Kanban*. Санкт-Петербург, Питер, 2019, 304 с.
- [4] Андерсон Д. *Kanban. Альтернативный путь в Agile*. URL: <http://flibusta.is/b/479765/read> (дата обращения 17.10.2023).
- [5] Внуков Н.С., Селиванов К.В. Методика эффективного проектирования высокоскоростных интерфейсов. *Технологии инженерных и информационных систем*, 2020, № 2, с. 14–27.
- [6] Власов А.И., Карпунин А.А., Курышев Р.Э. Визуальное моделирование SMART-технологий проектного управления. *Надежность и качество. Тр. междунар. симпозиума*, Пенза, ПГУ, 2020, т. 1, с. 64–70.
- [7] Ванройе Н.К., Карпунин А.А. Анализ принципов бизнес-планирования ИТ-проекта. *Информационные технологии в проектировании и производстве*, 2018, № 4 (172), с. 31–35.
- [8] Власов А.И., Карпунин А.А., Ганев Ю.М. Системный подход к проектированию при каскадной и итеративной модели жизненного цикла. *Надежность и качество. Тр. междунар. симпозиума*, Пенза, ПГУ, 2015, т. 1, с. 96–100.

Сорокин Сергей Сергеевич — студент кафедры «Проектирование и технология производства ЭА», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Сорокина Алиса Максимовна — студентка кафедры «Проектирование и технология производства ЭА», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:

Сорокин С.С., Сорокина А.М. Гибкая методология разработки программного обеспечения (Agile). *Политехнический молодежный журнал*, 2023, № 11 (88). <http://dx.doi.org/10.18698/2541-8009-2023-11-951>

FLEXIBLE SOFTWARE DEVELOPMENT METHODOLOGY (AGILE)

S.S. Sorokin

sorokin.sns@yandex.ru

A.M. Sorokina

maximova-alisa@mail.ru

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

As of today, flexible software development methodology, also known as the Agile, became one of the most popular and in-demand in the information technology industry. Instead of traditional and rigid approaches to development, Agile offers an iterative and flexible method making it possible for the development teams to be more efficient and adapt to changes in the project requirements. This article considers the Agile basic principles and practices, and the benefits it could provide to the software development teams.

Keywords

Agile, flexible methodology, software, iterative approach, incremental development, scrum, extreme programming, kanban

Received 03.11.2023

© Bauman Moscow State Technical University, 2023

References

- [1] Agile-manifest razrabotki programmnogo obespecheniya [Agile Software Development Manifesto]. URL: <https://agilemanifesto.org/iso/ru/manifesto.html> (accessed October 17, 2023).
- [2] Shtayn E. *Filosofiya Lean. Berezhlivoe proizvodstvo na rabote i doma* [Lean philosophy. Lean manufacturing at work and at home]. URL: <http://flibusta.is/b/424958/read> (accessed October 17, 2023).
- [3] Koul R., Skotcher E. *Blistatel'nyy Agile: gibkoe upravlenie proektami s pomoshch'yu Agile, Scrum i Kanban* [Brilliant Agile: Agile Project Management with Agile, Scrum and Kanban]. Sankt-Petersburg, Piter Publ., 2019, 304 p. (In Russ.).
- [4] Anderson D. *Kanban. Al'ternativnyy put' v Agile* [Kanban. Alternative path to Agile]. URL: <http://flibusta.is/b/479765/read> (accessed October 17, 2023).
- [5] Vnukov N.S., Selivanov K.V. Techniques for efficient design of high-speed interfaces. *Technologies of engineering and information systems*, 2020, no. 2, pp. 14–27. (In Russ.).
- [6] Vlasov A.I., Karpunin A.A., Kuryshev R.E. Visual modeling of SMART project management technologies. *Nadezhnost' i kachestvo. Tr. mezhdunar. simpoziuma* [Reliability and quality. Proceedings of the international symposium], Penza, PGU Publ., 2020, vol. 1, pp. 64–70. (In Russ.).
- [7] Vanroye N.K., Karpunin A.A. Analysis of principles of business planning of IT project. *Informatsionnye tekhnologii v proektirovanii i proizvodstve*, 2018, no. 4 (172), pp. 31–35. (In Russ.).
- [8] Vlasov A.I., Karpunin A.A., Ganey Yu.M. A systematic approach to design with a cascade and iterative life cycle model. *Nadezhnost' i kachestvo. Tr. mezhdunar. simpoziuma* [Reliability and quality. Proceedings of the international symposium], Penza, PGU Publ., 2015, vol. 1, pp. 96–100. (In Russ.).

Sorokin S.S. — Student, Department of Design and Technology of the ES Production, Bauman Moscow State Technical University, Moscow, Russian Federation.

Sorokina A.M. — Student, Department of Design and Technology of the ES Production, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:

Sorokin S.S., Sorokina A.M. Flexible software development methodology (Agile). *Politekhnichestkiy molodezhnyy zhurnal*, 2023, no. 11 (88). (In Russ.).

<http://dx.doi.org/10.18698/2541-8009-2023-11-951>