

Глава 1

НЕОБХОДИМОСТЬ AGILE

Сегодня Agile становится доминирующим способом разработки программного обеспечения. Этот итерационный поступательный метод работы, основанный на продолжительном изучении и быстрой доставке бизнес-ценности, берет верх над хорошо спланированным, детерминированным мышлением прошлого.

Но почему все больше и больше корпораций — от традиционных энергетических и нефтегазовых компаний до гигантов инновационных технологий — хотят распространить гибкость за пределами сферы разработки продуктов и вживить ее в самую ДНК организации? Почему традиционные, еще вчера пригодные способы работы с вертикальной иерархией власти кажутся неподходящими в сегодняшней бизнес-среде?

В этой главе мы поговорим о том, почему руководители видят гибкость бизнеса одним из своих приоритетов¹. Я начну с истоков современной управленческой мысли, расскажу, где и почему она появилась, и объясню, как современная бизнес-среда определяется через нестабильность (volatility), неопределенность (uncertainty), сложность (complexity) и неоднозначность (ambiguity) — модель VUCA. Затем я поделюсь, как организации могут получать прибыль от беспорядка и как можно оптимизировать бизнес-системы, чтобы они приняли естественную изменчивость современной бизнес-среды, вместо того чтобы избегать ее.

Наконец, я расскажу о разработке Agile-манифеста (полное название «Манифест гибкой разработки программного обеспечения») — документа,

созданного в 2001 году семнадцатью лидерами технической мысли и обращенного к трудностям, характерным для сферы разработки ПО. Вы узнаете, как создавался манифест, почему он работает для ПО и как его ценности и принципы соответствуют потребностям крупных организаций в силу того, что они противодействуют сложности, скорости и неопределенности, присущим современной бизнес-среде.

Прочитав эту главу, вы поймете, как и почему гибкость бизнеса стала столь важной частью стратегических потребностей современных компаний. Вы также сможете распознавать среды, в которых Agile будет полезен, и те, в которых он не принесет пользы. Чтение этой главы подготовит вас к глубокому погружению в гибкость на уровне компаний, закладывая фундамент для остальной информации, изложенной в книге.

АФИНЫ ПОБЕЖДАЮТ MICROSOFT

Джеффри Уэст, выдающийся профессор теоретической физики и биологии престижного Института Санта-Фе, изучал нелинейные системы, такие как галактики, функции головного мозга и Солнечная система. Особенно интересными он нашел две системы: город и организацию. Обе они — конструкторы, созданные человеком и обладающие одними и теми же характеристиками. Однако у них есть важное отличие, которое заставило Уэста задаться вопросом: **почему города имеют тенденцию к вечной жизни, в то время как компании умирают сравнительно быстро?** В конце концов, базовые элементы обеих систем совпадают. И города, и компании — нелинейные системы, объединенные совместными ограничениями и управляемые известным лидером. Обе системы в конечном счете направлены на одну цель — максимально увеличить ценность для заинтересованных лиц, будь то горожане или акционеры².

Однако даже старейшая корпорация в мире — японская Kongo Gumi, чей возраст насчитывает более 1400 лет, — сущий ребенок в сравнении с городами вроде египетского Луксора, которому более 5200 лет*.

Города кажутся практически нерушимыми в сравнении с компаниями: они могут переживать тяжелые экономические потрясения, природные катаклизмы, даже ядерные войны, но восстанавливаться и процветать.

* Компания была основана в 578 году и просуществовала до 2007 года, на момент закрытия ей было более 1400 лет. *Прим. ред.*

Старейшие компании и города мира*Десять старейших компаний мира³*

Примерный год основания	Название	Сфера деятельности
578 г.	Kongo Gumi	Строительная компания
705 г.	Nishiyama Onsen Keiunkan	Отель
803 г.	St. Peter Stiftskulinarium*	Ресторан
900 г.	Sean's Bar	Паб
1000 г.	Château de Goulaine	Винодельня
1040 г.	Pontificia Fonderia Marinelli	Литейная компания
1488 г.	Rathbornes	Производство свечей
1526 г.	Beretta	Производство огнестрельного оружия
1613 г.	The Shirley Plantation	Сельскохозяйственная компания
1645 г.	Post-och Inrikes Tidningar	Газета

* До 2017-го он назывался Stiftskeller.

Десять старейших населенных городов мира⁴

Примерный год основания*	Название
~8000-10 000 г. до н. э.	Дамаск
7000-9000 г. до н. э.	Иерихон
6000-7000 г. до н. э.	Алеппо
4000-5000 г. до н. э.	Афины
4000-5000 г. до н. э.	Лоян
4000-5000 г. до н. э.	Пловдив
4000-5000 г. до н. э.	Библос
3000-4000 г. до н. э.	Сидон
3000-4000 г. до н. э.	Рей
2000-3000 г. до н. э.	Иерусалим

* Оценка точного возраста городов в разных источниках отличается.

Наблюдение профессора Уэста было подтверждено: компании значительно более хрупки. Средний срок жизни компании, согласно рейтингу Fortune Global 500, уменьшается все быстрее: с 60 лет в 1958 году до 18 лет сегодня. Компании не просто умирают молодыми, они быстрее исчезают. В исследовании Boston Consulting Group 2015 года отмечалось, что каждый год разоряется десятая часть всех ОАО: это в четыре раза больше, чем в 1965 году⁵.

Города продолжают процветать: по состоянию на 2006 год более половины населения земли проживало в городах; к 2050 году ожидается, что в городах будут жить более 75% жителей планеты. Фактически каждую

неделю с настоящего момента до 2050 года городское население станет увеличиваться на один миллион человек⁶.

Как две столь похожие сущности могут настолько отличаться в темпах роста и успешности?

Профессор Уэст утверждает: ключевое различие состоит в том, как управляют городом, — а точнее, как город себя организует. У обеих систем есть лидеры (СЕО и мэры соответственно) и последователи (горожане и наемные сотрудники), но их способы достижения целей значительно отличаются. Компании обычно управляются сверху вниз, имеют жесткую иерархическую структуру, в которой лидер указывает, что должно быть сделано, и описывает в общих чертах, как это должно быть сделано. Граждане, наоборот, управляются свободной сферой влияния, в которой лидеры редко контролируют, но значительно воздействуют на поведение граждан посредством установления линий поведения, стимулов и ограничений.

Разница в результатах ошеломляет: по мере роста городов их продуктивность возрастает суперлинейно. Таким образом, чем больше людей добавлено в равенство, тем более продуктивными, инновационными и ценными оказываются города. По мере роста город становится сильнее и стабильнее. Прирост населения Берлина, например, вдвое превосходит ожидаемый, а его экономическое развитие с 2006 года продолжает постоянно обгонять экономический рост Германии.

Компании же демонстрируют резкое падение продуктивности по мере роста. Кривая продуктивности компаний по факту сублинейная: продуктивность падает с каждым новым сотрудником в таких сферах, как инновации (где она измеряется в количестве патентов на сотрудника), прибыльность (прибыль на сотрудника) и эффективность (полученный результат на сотрудника). Данные шокируют: за редким исключением, с течением времени явление повторяется снова и снова — компании слабеют, теряют продуктивность и конкурентоспособность по мере роста.

Почему так происходит? С увеличением размера компании усложняется ее структура, становится труднее работать быстро. Компания, в которой более пяти тысяч сотрудников, с трудом может изменить курс или оперативно переоценить свои приоритеты. Однако города показывают нам, что повышать уровень продуктивности по мере роста вполне возможно. Что такого есть в городах, что делает их практически нерушимыми и все более продуктивными по мере развития? Почему компании не могут повторить этого и теряют преимущества и инициативу?

Позже в этой главе мы вернемся к открытиям профессора Уэста и их возможному влиянию на современные организации. А для начала обратим пристальное внимание на то, почему компании управляются с использованием иерархической (сверху вниз), тщательно спланированной методологии, и изучим истоки традиционной управленческой мысли. Управление как дисциплина — сравнительно новое направление научной мысли, насчитывающее не более 100 лет.

ПРОИСХОЖДЕНИЕ СОВРЕМЕННОГО МЕНЕДЖМЕНТА

Когда корпорации только появились, к ним не прилагался единственно верный, оптимальный способ управления. Менеджмент и рыночные отношения эволюционировали со времен первой индустриальной революции, но теории и практики организации работы и увеличения прибыли все еще сравнительно новы. Многие из того, что преподают в курсе менеджмента, отражает мышление начала XX века. Чтобы лучше понимать, почему современные организации обращаются к более гибким способам работы, нужно обратиться к истории современной управленческой мысли — к тому, где она началась и почему ее теоретические основы, возникшие на базе мануфактур и фабричного труда, не подходят для многих современных организаций. Далее мы рассмотрим научное управление, становление работника информационной сферы и увидим, как разработчики ПО научились принимать неопределенность и стремиться к гибкости. Мы посмотрим на истоки и создание Agile-манифеста, обсудим VUCA и представим Sunefin — управленческий фреймворк для принятия решений в мире быстро меняющихся бизнес-контекстов.

НАУЧНОЕ УПРАВЛЕНИЕ: СОЗДАНИЕ БОЛЕЕ ЭФФЕКТИВНЫХ МАШИН

Менеджмент как академическая дисциплина — ребенок в сравнении с другими отраслями знания. Философия родилась в Древней Греции, математика и физика — тоже. Даже сравнительно новые академические дисциплины вроде психологии и социологии довольно старые по сравнению с менеджментом. Первая в мире программа MBA была создана в 1908 году в Гарвардской школе бизнеса, через 270 лет после его основания⁷.

Сначала менеджмент как дисциплина был довольно ограничен и покрывал только базовый функционал: бухгалтерское дело, управление компанией,

управление общим имуществом. Так было до тех пор, пока Фредерик Уинслоу Тейлор не инициировал движение эффективности и не расширил горизонты бизнеса, применив инженерное мышление к предприятиям. В своей основополагающей книге «Принципы научного менеджмента», выпущенной в 1911 году, Тейлор показал, как количественная оценка продуктивности и прогресса и контроль ресурсов помогают снизить себестоимость единицы продукции и повысить производительность труда⁸.

Один из основных аргументов, лежащих в основе работ Тейлора, — что организация, в сущности, близка к машине. Машина сделана из набора известных взаимосвязанных частей; и, чтобы убедиться, что она может работать максимально эффективно, управляющий должен выжать из своих ресурсов максимум. Его работа — провести оптимизацию так, чтобы максимизировать результат, снизить себестоимость единицы продукции, максимально повысить прибыль. Так, Тейлор утверждал, что, внимательно изучив рабочий процесс, можно определить лучший способ выполнить работу. Затем, разбив процесс выполнения на этапы, необходимые, чтобы выполнить работу лучшим способом, можно создать точное руководство, согласно которому сотрудник сможет выполнять задачу, и оптимизировать его работу в целях повышения эффективности. Если определить и перечислить сотни и тысячи процессов, естественных для организации, сотрудники (или ресурсы, как их называл Тейлор) смогут следовать точным инструкциям и не отклоняться от плана. Чем меньше отклонений, тем лучше — независимое мышление и творчество сотрудника мешают делу, поскольку могут изменить оптимальный способ выполнения работы и снизить эффективность ресурса.

Сегодня взгляды Тейлора могут показаться странными, но нужно брать в расчет время, когда были написаны его труды. В 1911 году на западный мир влияла вторая индустриальная революция. Организационная продуктивность поднялась до невиданных ранее высот, в руках владельцев фабрик были сосредоточены колоссальные богатства. Однако имелись и существенные негативные эффекты: автоматизация оставила торговцев без работы, создавая массовую безработицу, зарплаты значительно снизились. Ход бизнеса диктовали машины, и люди редко рассматривались как способ их поддержки, увеличивающий скорость работы. Взгляд на организацию как на машину, ресурсы которой должны быть оптимизированы, имел смысл, это воспринималось как разумный способ управления бизнесом.

Вы, должно быть, спросите, почему я рассказываю в книге о гибкости на уровне компаний о мысли, которой более ста лет. Причина проста: призрак

Фредерика Тейлора и его традиционные взгляды на бизнес влияют на бизнес-образование и сегодня. Системы, используемые в большинстве крупных организаций, базируются на теории менеджмента Тейлора. Вам ничего не напоминают следующие фразы?

- «Основанные на автоматизации схемы оптимизации индивидуальной работы».
- «Проектные планы и обязательства, основанные на оценке, установленной менеджером, который не задействован непосредственно в выполнении работы».
- «Фиксированные годовые бюджеты со строгими механизмами контроля».

Это малая часть идей Тейлора — наблюдений и теорий, основанных на бизнес-среде 1911 года.

Нельзя не признать, что долгое время идеи, сформулированные в «Принципах научного менеджмента», выполняли свою задачу. На протяжении второй индустриальной революции эффективность — способность максимально выжать любой возможный ресурс — была первостепенным делом. Фактически она стала единственной конкурентной задачей того времени. Организации были относительно изолированы от конкуренции в силу ограниченной доступности капиталовложений, доступа к информации и инфраструктуры, которая делала практически невозможной конкуренцию по времени. Компании, способные быстро организовать и занять доминирующую позицию, с самого начала имели значительное конкурентное преимущество, которое было трудно превзойти.

В такой бизнес-среде у растущих компаний было больше времени, чтобы перестроить свои бизнес-стратегии к ситуации, если они понимали, что двигаются в неправильном направлении. При необходимости они могли эффективно подавить конкурентов до того, как те начинали представлять реальную угрозу. С начала XX века и до 50-х годов единственным ключом к успеху была оптимизация ресурсов: определение плана действий, основанного на тщательном анализе, а затем работа согласно плану.

СТАНОВЛЕНИЕ ИНФОРМАЦИОННОГО РАБОТНИКА: РАСКРЫТИЕ ТВОРЧЕСКОГО ПОТЕНЦИАЛА

В 50–60-х годах прошлого века (иногда это время называют третьей индустриальной революцией или цифровой революцией) западный мир перешел

от индустриальной экономики к экономике знаний, которой свойственна высокая дифференциация продуктов. Стало понятно, что текущие системы не рассчитаны на поддержание нового типа работы. Если в 1920-х Генри Форд мог заявить, что «покупатель может получить автомобиль любого цвета при условии, что этот цвет — чёрный», то теперь экономический успех все больше основывался на нематериальных вещах — знаниях, умениях, инновационном потенциале как ключевых ресурсах конкурентного преимущества⁹.

Все больше высокообразованных сотрудников понимали, что трудятся в компаниях, пытающихся контролировать работу, которую просто невозможно контролировать. Конечно, можно точно измерить, сколько времени займет закручивание гайки, можно определить степень свободы с учетом заданных технических характеристик, чтобы обеспечить качество, — но как подобный тип количественного мышления можно применить к оценке сложного умственного труда врача или инвестиционного аналитика? Как измерить продуктивность журналиста, пишущего статью? Подсчет количества слов на странице — подходящая замена потребительской ценности?

Некоторые мыслители заметили такое развитие ситуации и описали фундаментальные изменения, необходимые, чтобы приспособиться к новой экономике и совершенно иному способу работы. Рутинные задания, оформление сделок или простое принятие заказов — все это потеряло смысл. Усиливалась потребность в обработке данных для установления отношений, определения трендов и понимания причинно-следственных связей.

Австрийский теоретик менеджмента Питер Друкер, будучи консультантом по вопросам управления, проанализировал опыт множества компаний и пришел к следующему выводу: для успеха на рынке, где доминируют информация и знания, компании должны поднять глаза от метрик эффективности и выгоды и сфокусироваться на раскрытии потенциала людей внутри организации. Он заметил, что сотрудники обычно знают больше, чем их менеджеры, и основная цель менеджмента — подготовить людей к работе и позволить им выполнять ее.

В 1959 году Друкер выпустил книгу «Ориентиры будущего» (The Landmarks of Tomorrow), в которой впервые предложил термин «информационный работник». Работа современных сотрудников выполняется с помощью их умов, а не рук, и менеджмент должен развиваться так, чтобы поддерживать эту реальность, создавая среду для роста, развития и обучения, писал Друкер¹⁰.

Развивая это понятие «информационного работника», в 1990 году Питер Сенге, старший преподаватель Массачусетского технологического института

и сооснователь компании Society for Organizational Learning, предложил определение «обучающаяся организация». В книге «Пятая дисциплина: искусство и практика самообучающейся организации»* он пояснял: обучающаяся организация — это организация, которая «обеспечивает обучение всех ее членов и постоянно трансформируется сама». Сенге объясняет, что компании должны меняться, чтобы поддерживать более связный способ мышления. Чтобы пробудить потенциал большинства сотрудников, компания должна вести себя как сообщество, поддерживающее совместные обязательства. (Помните, что мы говорили о городах?)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПОГЛОЩАЕТ МИР: ПРИНЯТЬ НЕОПРЕДЕЛЕННОСТЬ И СТАТЬ ГИБКИМ

Сразу после появления теории Друкера и Сенге о сообществе и обучающихся организациях стали невероятно популярны, но нигде они не нашли такого отклика, как в новой группе информационных работников, возникшей в конце 1980-х, — разработчиков программного обеспечения. Эта профессия была сравнительно молода, но изобретение персональных компьютеров и интернета создало огромный спрос на разработчиков ПО и на вещи, естественной частью которых оно являлось, — от огромных ЭВМ до карманных калькуляторов и кофемашин; а их количество росло. Число вакансий в сфере разработки ПО быстро множилось, знаменуя появление нового типа работника (и работы), что требовало составления новых правил.

Многие нужные разработчикам качества совпадали с качествами ранних информационных работников — архитекторов, юристов, академиков; но создание ПО делало акцент на коллективную работу, продолжительное обучение, эффективную коммуникацию. Разработка ПО — и искусство, и наука. Суть этой дисциплины — решение сложных проблем посредством алгоритмов, которые могут использовать вычислительную мощностъ машин.

Представление о том, что лучшее понимание проблемы появляется посредством экспериментов и совместной работы членов команды, применяется и за пределами разработки ПО — в мире разработки продуктов в целом. В 1986 году в журнале *Harvard Business Review* вышла статья «Разработка нового продукта. Новые правила игры», авторы которой, Хиротака Такеучи и Икуджиро Нонака, провели аналогию между новым способом работы

* Сенге П. Пятая дисциплина: искусство и практика обучающейся организации. М. : МИФ, 2018. *Прим. ред.*

и регби, метафорически описав, как игроки постоянно делятся информацией и передают мяч знаний друг другу.

Сравнение весьма удачное: подобно спортивной команде, команда разработки ПО — это группа людей, чьи умения дополняют друг друга, сбавшаяся, чтобы достичь общей цели.

Для создания решений, которые понравятся потребителям, нужно проверять предположения об их нуждах, способствуя тем самым обучению — неотъемлемой части процесса разработки продукта. Такеучи и Нонака предупреждали, что способы работы, при которых все предположения о потребителях установлены заранее, несостоятельны:

«Транснациональные компании должны стать быстрыми и гибкими в разработке продуктов. Чтобы сделать это, нужно использовать динамический процесс, который во многом полагается на метод проб и ошибок и на обучение в процессе работы. Сегодня, в мире постоянных изменений, нужны непрерывные инновации»^{11}.*

Это понимание заинтересовало Джеффа Сазерленда, бывшего военного летчика, занимавшегося развитием IT-систем в Easel Corporation. Вместе с коллегами он начал искать более гибкий путь разработки программного обеспечения, который учитывал бы обучение, основанное на опыте, напряженную совместную работу и частые петли обратной связи. В 1995 году Сазерленд вместе с Кеном Швабером, разработчиком ПО и промышленным консультантом, формализовали этот способ работы и назвали его фреймворком Scrum. Он был представлен на индустриальной конференции OOPSLA¹². Scrum помог открытиям Такеучи и Нонаки оформиться в восхитительно простую по структуре, но сложную в овладении методологию, применяемую для разработки продуктов (больше о Scrum можно узнать в главе 3 «Технологии»).

Scrum обращен к управленческой стороне разработки продуктов, но не освещает специфические технические практики в ПО. Кент Бек, выдающийся инженер ПО, обратился напрямую к этой стороне вопроса, когда в конце 1990-х представил концепцию экстремального программирования (Extreme Programming, XP) в одноименной книге. Главная цель XP — уменьшение стоимости изменений; чем быстрее схлопывается петля обратной связи для

* Источник: <http://agilerussia.ru/methodologies/%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0-%D0%BD%D0%BE%D0%B2%D0%BE%D0%B3%D0%BE-%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D0%B0-%D0%BD%D0%BE%D0%B2%D1%8B%D0%B5-%D0%BF%D1%80%D0%B0/>. Прим. пер.

поступательных циклических изменений в процессе работы, тем с большей вероятностью мы создаем ПО, которые хотели создать. Бек убеждал менеджеров признать: изменения — естественный и желательный аспект разработки ПО. Вместо того чтобы пытаться определить стабильный набор требований, нужно быть готовым к изменениям и предвидеть их как ожидаемую часть процесса разработки продукта (см. рис. 1.1)¹³.

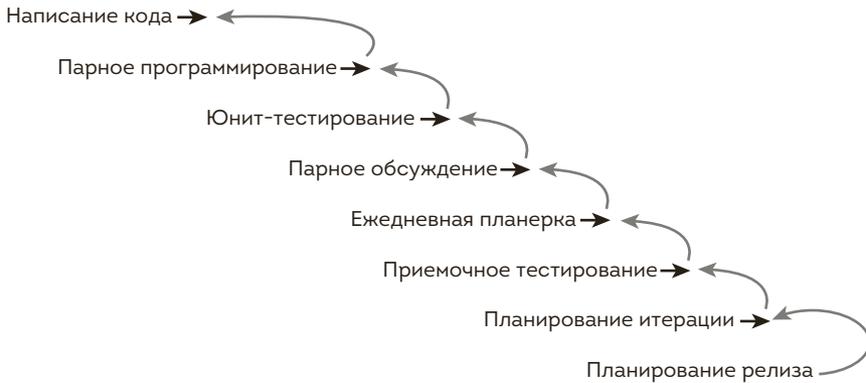


Рис. 1.1. Практики экстремального программирования создают возможности быстрого обучения посредством множественных слоев петли обратной связи

Работа Бека привлекла внимание Роберта Мартина, успешного консультанта по C++ и объектно-ориентированному проектированию, работавшего в Чикаго. Клиенты часто просили Мартина придумать процесс, который сможет систематизировать практики, поставляемые им своим клиентам. Поэтому его заинтриговала работа Бека и после мюнхенской конференции, посвященной вопросам ПО, они стали общаться. Мартин чувствовал, что Бек знает что-то действительно важное, и поехал в Портленд, чтобы научиться разработке через тестирование и парному программированию. Чем больше Мартин узнавал, тем больше убеждался, что они с Беком приближаются к тому, о чем просили клиенты.

Одновременно с этим, в 1991 году, Алистеру Коуберну, методологу, работающему в новой консалтинговой группе компании IBM, дали задание найти причины успеха проектных групп. Углубившись в изучение проектов на Smalltalk и C++, Коуберн обнаружил, что в литературе того времени не были описаны факторы, делающие проектные группы успешными. Опросив десятки таких групп, он выделил ключевые элементы успешных команд в семейство методологий Crystal: серию легких процессов, подходящих к определенным типам проектов. Основная идея заключается в том, что

каждому проекту нужна своя методология и методы должны меняться так же, как меняется ситуация¹⁴.

Пока эти лидеры мнений искали способы написать лучшее ПО, индустрия сама по себе находилась в плачевном состоянии. На протяжении 1970–1990-х было несколько происшествий, породивших заголовки о крупных превышениях сметы, хронических проблемах с качеством и возмутительных задержках продукта. В 1994 году группа из Стэндиша опубликовала отчет СНАОС (акроним, англ. chaos — хаос), согласно которому среднее превышение запланированных расходов на проекты программного обеспечения составляло 189%¹⁵. Отчет часто цитировали. Согласно другим данным, примерно половина проектов работали, но немногие из них считались успешными. Кроме того, три четверти всех крупных программных продуктов, представленных потребителям, не отвечали их требованиям.

Ситуация не осталась без внимания программистов. Они поняли, что текущая ситуация неустойчива. На протяжении 1990-х профессионалы сферы ПО регулярно встречались и делились идеями. Что общего есть у появляющихся подходов? Какими способами можно улучшить написание ПО? Люди понимали, что ситуацию надо менять, но их идеи еще не могли срастись во что-то значимое.

Нужно было что-то делать. Поскольку XP начало продвигаться в сообществе, Мартин встретился в Чикаго с Мартином Фаулером, экспертом XP, с которым он познакомился на конференции Кента Бека XP Leadership осенью 2000 года. Вместе они пришли к выводу, что необходимо организовать что-то вроде саммита, чтобы люди с похожими взглядами смогли собраться и создать манифест легких процессов. Мартин и Фаулер начали подбирать участников этой встречи так, чтобы на ней были представлены разные взгляды. Одним из тех, кто получил приглашение на «саммит легких процессов» по электронной почте, был Алистер Коуберн.

Коуберн, который в тот момент был занят организацией похожей встречи по тем же причинам, что и Мартин с Фаулером, с радостью принял приглашение, предложил взять на себя логистику и организовать встречу неподалеку от его дома в Юте¹⁶.

Несколько месяцев спустя, в феврале 2001 года, Бек, Коуберн, Мартин, Фаулер, Швабер, Сазерленд и еще одиннадцать опытных лидеров мнений собрались в городе Сноуберде, чтобы понять, что не так в способах создания ПО. Они задались вопросом: как они могли бы улучшить способы разработки ПО, используя свой опыт? Как разработчики ПО, они истово

гордились собственным ремеслом, однако их не удовлетворяло ни состояние отрасли в целом, ни негативное восприятие разработки ПО как профессии. Они хотели создавать ПО, которое нравится потребителям, и повлиять на организации, формировавшие среду с лучшим ПО^{17, 18}.

Через несколько дней споров и обсуждений разработчики создали «Манифест гибкой разработки программного обеспечения» (Agile-манифест), состоящий из четырех ценностей и двенадцати принципов¹⁹.

Манифест гибкой разработки программного обеспечения*

Мы постоянно открываем для себя более совершенные методы разработки программного обеспечения, занимаясь разработкой непосредственно и помогая в этом другим.

Благодаря проделанной работе мы смогли осознать, что:

- **люди и взаимодействие** важнее процессов и инструментов;
- **работающий продукт** важнее исчерпывающей документации;
- **сотрудничество с заказчиком** важнее согласования условий контракта;
- **готовность к изменениям** важнее следования первоначальному плану.

То есть, не отрицая важности того, что справа, мы все-таки больше ценим то, что слева.

Следом за ценностями через несколько недель после встречи в Сноуберде были разработаны двенадцать принципов.

1. Наивысший приоритет для нас – 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, чтобы обеспечить заказчику конкурентное преимущество.
3. Работающий продукт следует выпускать как можно чаще, с периодичностью от двух-трех недель до двух-трех месяцев.
4. На протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе.
5. Над проектом должны работать мотивированные профессионалы. Чтобы проект был реализован, создайте условия, обеспечьте поддержку и полностью доверьтесь им.
6. Непосредственное общение – наиболее практичный и эффективный способ обмена информацией как с командой, так и внутри команды.
7. Работающий продукт – 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.

* Цитируется по: <http://agilemanifesto.org/iso/ru/manifesto.html>. *Прим. пер.*



[Почитать описание, рецензии
и купить на сайте](#)

Лучшие цитаты из книг, бесплатные главы и новинки:

