Если писать сложную программу без плана, то ничего не выйдет. Это как выпустить автомобиль: нужно подобрать двигатель, изучить рынок, проработать конструкцию, создать дизайн и собрать все на конвейере. Когда не знаешь, что делать и в каком порядке, то ни машина, ни программа не заведутся.

Чтобы процесс создания программы шел правильно, программисты используют методологии. Методология — это набор стратегий и способов создания продукта. Их много и новичок не знает, что выбрать: RUP, XP, Scrum, Waterfall или другой набор букв.

**Agile** — это молодая семья гибких и демократичных методологий. Её лозунги привлекательны как «Свобода, равенство и братство», а принципы даже кажутся утопичными.

Давайте посмотрим, что такое Agile и что делать, чтобы он заработал.

До появления Agile IT-компании использовали каскадную модель разработки (она еще известна как водопадная, потому что по-английски называется Waterfall). Чтобы стало чуть яснее, что́ с ней не так, скажу, что методологию сформулировали в 1970-м. Её суть в том, что работа над проектом состоит из жесткой последовательности этапов:

**Анализ требований**

**↓  
Проектирование  
↓  
Реализация  
↓  
Тестирование  
↓  
Интеграция  
↓  
Поддержка**

Пока не закончится предыдущий этап, не может начаться следующий. Тут и начинаются проблемы.

Предположим, тестировщики нашли в продукте серьезные ошибки, и его нужно перепроектировать. Работа откатывается на два шага назад, и проект снова проходит стадии проектирования, разработки и тестирования.

При каскадной разработке между анализом требований к продукту и интеграцией (выводом на рынок) проходят месяцы и годы. К этому времени рынок меняется, продукт порой устаревает еще до релиза.

Риск потерять уйму денег — хороший повод поразмыслить. В 2001 году практикующие айтишники собрались на горнолыжном курорте в Юте. Официальный сайт преподносит событие так: *«…seventeen people met to talk, ski, relax, and try to find common ground — and of course, to eat»*. Итогом стала *Библия Agile — Agile Manifesto*.

**Agile — простая и симпатичная философия**

Если вы думаете, что авторы Agile Manifesto написали сборник советов о работе над проектом, вы немного ошибаетесь. Манифест включает в себя 4 ценности, 12 принципов и 0 практик.

**Начнем с ценностей, потому что именно они соль Agile:**

*1. Люди и взаимодействие важнее процессов и инструментов.*

*2. Работающий продукт важнее исчерпывающей документации.*

*3. Сотрудничество с заказчиком важнее согласования условий контракта.*

*4. Готовность к изменениям важнее следования первоначальному плану.*

Разработка ПО — это бессонные ночи авралов, скандалы между проектировщиками и кодерами, переносы релизов, нервы и прочие прелести реальной жизни.

К счастью, Agile — больше, чем набор из четырех заповедей. Авторы прописали в манифесте также 12 принципов методологии. Принципы тоже непохожи на руководство стартапера, но в них уже есть настоящее мясо.

1. Наивысшим приоритетом для нас 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 заказчику 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 представители бизнеса должны ежедневно работать вместе.

5. Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте условия, обеспечьте поддержку и полностью доверьтесь им.

6. Непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой 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.

**Разработка делится на спринты**

Авторы Agile сделали основой методологии старую шутку «— Как съесть слона? — По кусочкам». Уже в этой части гибкий подход отличается от каскадного.

**«Водопад»** Месяцами «пилить» проект, запустить его по готовности.

**Agile** Разбить продукт на сотни небольших задач, выполнять их порциями и сразу выпускать обновления на рынок.

Команда проекта разбивает разработку на небольшие периоды — спринты. В последние годы популярны двухнедельные спринты.

С каждым спринтом продукт становится полезнее

Перед первым спринтом команда и заказчик проводят совещание. Заказчик — условная роль, им может быть и член команды. Он определяет задачи, которые команда решит за спринт.

Здесь появляется первая сложность. Согласно философии Agile, после каждого спринта продукт должен быть:

* работоспособным;
* полезным для пользователя;
* более совершенным, чем до спринта.

**Agile-подход к разработке продукта.**

Результатом первых недель работы должен стать продукт, готовый к выходу на рынок. Такой продукт называют минимально работоспособным (Minimum Viable Product, MVP).

Запуск MVP — важная точка проекта. Удачный MVP либо сразу привлекает пользователей, либо сразу проваливается и показывает нежизнеспособность идеи.

Допустим, мы делаем интернет-магазин. Магазину нужны карточки товаров, поиск, корзина, профили пользователей, бонусная система, рассылки и все такое. Если делать это по каскадной методологии, до запуска пройдут месяцы. Мы идем по Agile, поэтому ищем MVP. Для онлайн-магазина это, внезапно, лендинг, который собирает электронные адреса потенциальных клиентов. Посетитель, который оставит емейл, получает скидку на первую покупку. Сделать такой лендинг за спринт проще простого.

Этот MVP приносит пользу: лэндинг покажет, насколько наш магазин интересен аудитории, а потенциальные покупатели получат скидку. Это удачный MVP. Подробнее про суть минимально работоспособного продукта хорошо написал инвестор Аркадий Морейнис.

Возвращаемся к спринту. Во время него команда работает по модели, близкой к каскадной. Каждую новую функцию проектируют и программируют, а затем тестируют и документируют. Когда спринт завершен, команда имеет работоспособную, полезную и более совершенную версию продукта.

Перед следующим спринтом команда планирует следующий рывок. На этом этапе заказчик может добавить задачи, которых раньше не было. Agile поощряет то, что немыслимо для «водопада»: «Изменение требований приветствуется, даже на поздних стадиях разработки». В конце проекта продукт может сильно отличаться от того, что планировали на старте.

Пары «план — спринт» идут одна за другой, пока живет проект.

**Команда работает как захочет**

Внутренние и внешние связи agile-команды максимально демократичны. Одна из причин, по которым методология практически не работает в России — руководители не понимают, как можно «до такой степени распускать коллектив». Согласно Agile, команда — это самоорганизующаяся единица. Никто не имеет права указывать, как она решает задачи внутри спринта. Если хотят, пусть хоть на головах ходят.

Команда — это единое целое, которое не делится на конкретных людей. Ответственность лежит на всей команде: если наказывают или поощряют, то всех сразу. Главное в рабочем процессе — коммуникация. Команда сидит в одном помещении без перегородок и «кубиков», постоянно общаясь между собой. Общение с заказчиком также ежедневное.

В конце каждого спринта команда обсуждает, как работать эффективнее. Это называется ретроспективой. Суть Agile не только в постоянном улучшении продукта, но и в постоянном совершенствовании командной работы.

**На самом деле не все любят Agile**

У Agile есть противники, которые не разделяют общего восторга. Основная проблема методологии — хаос на дистанции. После каждого спринта меняются приоритеты и появляются новые задачи, поэтому у команды нет видения конечного продукта.

Две недели назад заказчик хотел интернет-магазин, а теперь понял, что это будет социальная сеть для ИП. Команда принимает в задачи на спринт чат и лайки. В следующий раз заказчик видит, что новый фильм про Джеймса Бонда взорвал Интернет. Он добавляет в спринт функции онлайн-кинотеатра. В результате архитектура проекта расползается, а полезное действие продукта становится размытым.

Еще одна проблема — плохой код. Agile проповедует максимальное число полезных изменений продукта в единицу времени. Поскольку цель — это полезные изменения, у программистов нет задачи сделать код как можно надежнее и понятнее.

Agile подталкивает к мысли: то, что функция работает, намного важнее того, как она реализована. На дистанции такой подход приводит к проблемам. Единственная страховка — сверхдисциплинированная и организованная команда.

**Если забыть о философии, ничего не выйдет**

Есть еще одна проблема, в которой авторы Agile не виноваты: методология стала слишком популярной и даже попсовой. Люди говорят, что работают по Agile, даже не понимая её сути. Они разбиваются на небольшие команды, нарезают проект на небольшие задачи, планируют спринты, регулярно релизят, но убогих проектов не становится меньше.

Помните, в начале статьи я говорил про принципы? Несмотря на их наивность, принципы — самое ценное в Agile. Сначала нужно осознать их и только потом браться за инструменты и практики.

**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**.

Люди бездумно следуют инструкциям, забывая, что Agile — это идеология и философия. А забывать нельзя: если не проникнуться духом методологии, сквозь нее пробьется «водопад» пополам с анархией и бюрократией.