ОГЛАВЛЕНИЕ

ГЛАВА 1
РЕТРОСПЕКТИВЫ 101

Основная цель первой главы — познакомить вас с ретроспективами. Я расскажу, как их использовать в семейной жизни, покажу модель поэтапной организации процессов и дам несколько советов по наполнению этих этапов содержанием. В главе описано все необходимое для начала работы с первой ретроспективой. Итак, начнем.

ЧТО ТАКОЕ РЕТРОСПЕКТИВА

Ретроспектива (от лат. retrospectare) — это обзор, взгляд назад. Когда по ночам, лежа в постели, вы вспоминаете события минувшего дня — это ретроспектива. Если за семейным ужином дети рассказывают, как провели время в школе, а родители делятся впечатлениями о работе — это ретроспектива. Творчество художника, литератора или режиссера тоже можно рассматривать с точки зрения ретроспективы. Например, в рамках ретроспективных выставок

демонстрируется целый ряд авторских работ, все важные произведения собраны в одном месте, чтобы дать полную картину творчества художника. Так можно получить общее впечатление и возможность сравнивать различные произведения искусства. Это было бы неосуществимо, имей мы доступ к одному-единственному примеру. Только получив целостное представление, можно оценить ситуацию и понять, почему художник поступил так, а не иначе.

Еще один вид ретроспективы есть на телевидении. В конце каждого года в рамках обзорных программ каналы конкурируют между собой, стремясь привлечь к участию в своих передачах самых веселых, красивых и известных людей. Но развлечение публики — приоритет для телевидения, и оно не слишком заботится о получении полной картины. Именно поэтому годовые «телеотчеты» довольно неоднородны и не позволяют делать выводы или рассматривать связи между различными событиями.

В этой книге словом «ретроспектива» я называю кое-что другое. Точно так же подразумевается взгляд в прошлое, но это только первый шаг. Задача — получить благодаря этому взгляду знания и понимание, которые помогут нам извлечь нужные уроки и правильно адаптироваться. Можно учиться как на успехах, так и на неудачах, а хорошее сделать еще лучше. Это сравнимо с эволюцией: то, что не сработало, вымирает, но все, способствующее сохранению видов, остается и развивается. В конце концов, каждая

из таких адаптаций — не что иное, как эксперимент, потому что результат никогда точно не известен. В оптимальном случае эксперименты приводят к улучшению ситуации. Но иногда они только усугубляют ее, и это необходимо анализировать уже в следующей ретроспективе.

Каждую ретроспективу проводит фасилитатор. Он гарантирует, что группа достигнет поставленных целей, и помогает развивать практические результаты, которые станут основой для будущего успеха. Фасилитатор не является участником (правда, в небольших группах это правило не всегда соблюдается). Он сопровождает процесс, но не вовлекается в реализацию решений. Хороший фасилитатор необходим для успешной ретроспективы.

Такая ретроспектива была впервые описана Норманом Кертом в книге «Ретроспектива проекта. Как проектным командам оглядываться назад, чтобы двигаться вперед» [1].

> *Ретроспектива — это ритуальное собрание сообщества в конце проекта для обзора событий и изучения опыта. Никто не знает всей истории проекта. У каждого человека есть часть истории. Ритуал ретроспективы — это коллективное рассказывание истории и добыча опыта для мудрости.*

В своей книге Керт объясняет, чем ретроспективы отличаются от так называемых посмертных и извлеченных

уроков. Главная разница в том, что ретроспективы фокусируются на будущих позитивных действиях и используют их в качестве катализатора изменений. Они представляют собой не конец проекта, а вехи в процессе постоянного совершенствования.

В 2001 году несколько человек собрались на горнолыжном курорте, где и написали манифест для гибкой разработки программного обеспечения [2]. Основа манифеста состоит из четырех пар ценностей и двенадцати принципов, последний из которых отлично описывает то, что происходит в ретроспективе.

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

Этот принцип — одна из главных причин того, что agile-сообщество с энтузиазмом включило ретроспективы в свой рабочий процесс. Люди поняли: не обязательно ждать окончания проекта, чтобы узнать, что произошло, и внести нужные изменения. Достаточно проводить ретроспективу после каждой итерации (то есть определенного периода). Этот интервал не должен превышать одного месяца, иначе есть риск слишком растянуть цикл обратной связи.

ЧТО ТАКОЕ ИТЕРАЦИЯ

Слово «итерация» происходит от латинского iterare, что означает «повторить». Итерации широко применяются там, где задачи решаются пошагово. В информатике итерация – это название процесса выполнения различных шагов до тех пор, пока не будет достигнуто требуемое условие (например, цикл FOR). В Scrum итерация называется «спринт».

Я использую термин «итерация» для описания процесса выполнения проекта в четко определенных, коротких, повторяющихся шагах. После каждой итерации вы останавливаетесь, чтобы определить, была ли и в какой степени реализована цель проекта, и при необходимости адаптируете исходный план. Цель — свести к минимуму риск неопределенности и неожиданностей. Та же процедура может использоваться в управлении изменениями.

Проведение ретроспектив позволяет наладить процесс непрерывного совершенствования, который постоянно проверяет, на правильном ли вы пути, а также дает возможность оперативно вмешаться и внести необходимые изменения. Выделив время для размышлений, вы сумеете решить проблему немедленно, не дожидаясь окончания процесса. Если не проводить ретроспективы до конца проекта, то вы рискуете к началу следующего забыть то, что узнали в текущем. Вы также получаете возможность реализовать улучшения в каждой итерации.

ЧТО ОЗНАЧАЕТ ТЕРМИН AGILE В ЭТОМ КОНТЕКСТЕ

Слово agile имеет латинское происхождение, и его примерные значения «делать» или «действовать». Как уже было сказано, эта методология основана на 12 принципах agile-манифеста [2].

Суть agile-манифеста в следующем: мы постоянно открываем для себя более совершенные методы разработки программного обеспечения, занимаясь им 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.

Иными словами, не отрицая важности того, что записано справа, мы все-таки больше ценим то, что слева.

Соответствующие 12 принципов выглядят так:

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 трудиться мотивированные профессионалы. Чтобы работа была сделана, создайте условия, обеспечьте поддержку и полностью доверьтесь им.
6. Непосредственное общение – наиболее практичный и эффективный способ обмена информацией как с самой командой, так и внутри нее.
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. Непрерывное 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.

Как видите, некоторые из принципов непосредственно нацелены на разработку программного обеспечения. Однако большинство можно легко применять и в других областях. Agile-манифест основан на фундаментальной идее о том, что мы живем в сложном и непредсказуемом мире. Так что создание детального плана проекта на несколько лет или даже месяцев вперед не имеет смысла. Как известно большинству людей, которые когда-либо составляли план проекта, после очень короткого времени он лишь отдаленно соответствует реальности.

Agile-разработчики понимают эту ситуацию и стараются минимизировать ее эффект, используя короткие циклы обратной связи и тесно сотрудничая с заказчиком.

На основе agile-манифеста были разработаны различные структуры и процессы. Среди них XP, DSDM, Open UP и, конечно же, очень популярный сегодня Scrum. В то же время идеи agile-разработки программного обеспечения распространились и на другие сферы.

Например, в своей книге The Leader's Guide to Radical Management Reinventing the Workplace for the 21st Century («Руководство лидера по радикальному управлению: переосмысление рабочего места в XXI веке») [3] Стивен Деннинг описывает применение идей agile-манифеста в области управления.