

Содержание

Предисловие 7

Глава 1. Введение..... II

Инструменты мышления

Глава 2. Системное мышление..... 21

Глава 3. Бережливый подход..... 53

Глава 4. Теория массового обслуживания..... 115

Глава 5. Ложные дихотомии..... 151

Глава 6. Будьте гибкими..... 168

Инструменты организации

Глава 7. Фича-команды..... 179

Глава 8. Команды..... 230

Глава 9. Области требований..... 254

Глава 10. Организация..... 266

Глава 11. Масштабированный скрам..... 332

Разное

Глава 12. Азбука скрама..... 349

Библиография 379

Предисловие

Мы благодарим вас за то, что вы решили прочитать эту книгу! Мы постарались сделать ее полезной. Дополнительные материалы и рекомендации можно найти на сайтах www.craiglarman.com. и www.odd-e.com. Пожалуйста, свяжитесь с нами, если у вас возникнут вопросы.

Условные обозначения в тексте

Момент, который нужно подчеркнуть, или не очень существенный новый термин. **Важная идея** или **важный новый термин**. [Bob67] — ссылка на библиографию.

Об авторах

Крэг Ларман — научный директор в международной консалтинговой и аутсорсинговой компании Valtech, имеющей подразделения в Европе, Азии и Северной Америке. Как консультант и коуч в области менеджмента и разработки работает с крупными и офшорными продуктовыми группами, помогая им с внедрением гибких и бережливых методологий разработки преимущественно с акцентом на встраиваемых системах. Руководил переходом на гибкую разработку (на основе скрама) в Valtech India; был автором и ведущим коучем в проекте по внедрению бережливой разработки ПО в компании Xerox.

На протяжении многих лет в качестве консультанта и коуча помогал с крупномасштабным внедрением гибкой разработки и скрама в компаниях Nokia, Siemens, NSN и многих других. Родился в Канаде, с 1978 г. периодически живет в Индии. Автор книг «Гибкая и итеративная разработка: руководство для менеджера» (Agile and Iterative Development: A Manager's Guide and Applying UML) и «Применение UML 2.0 и шаблонов проектирования»¹ — самых продаваемых в мире книг по гибкой и итеративной разработке и объектно-ориентированному проектированию и дизайну (OOA/D). Также вместе с Басом Водде написал книгу «Практики масштабирования бережливой и гибкой разработки: Масштабированный скрам для больших, распределенных и офшорных продуктовых групп» (Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum).

В юности мечтал стать странствующим уличным музыкантом, но ему не хватило таланта, поэтому он получил степень бакалавра, а затем магистра компьютерных наук в ведущем канадском Университете имени Саймона Фрейзера в Ванкувере. В 1970-х гг. занимался разработкой систем на языках APL и 4GL, а в начале 1980-х заинтересовался искусственным интеллектом (вследствие недостатка собственного).

Бас Водде — независимый консультант и коуч по разработке продуктов и использованию масштабированного скрама. В течение нескольких лет руководил в Nokia Networks внедрением гибкой разработки и скрама в масштабах всей компании. Входил в руководящую команду очень крупной распределенной командой разработки (работающей в Европе и Китае), осуществлявшей переход на скрам. Работал старшим разработчиком/архитектором в области встраиваемых телекоммуникационных систем, а также менеджером по качеству. Руководил разработкой решений и коучингом в проектах разработки через тестирование (TDD) встраиваемых систем. Вместе с Крэггом Ларманом написал книгу «Практики масштабирования бережливой и гибкой разработки». Родился в Голландии, много лет жил в Китае, сейчас живет в Сингапуре.

¹ Ларман К. Применение UML 2.0 и шаблонов проектирования: Введение в объектно-ориентированный анализ, проектирование и итеративную разработку. — Киев: Диалектика, 2019.

Благодарности

Прежде всего мы хотим сказать спасибо всем нашим клиентам.

Мы также хотим поблагодарить наших рецензентов и помощников: Питера Альфвина, Алана Атласа, Габриэль Бенефилд, Бьярте Богснес, Майка Бриа, Ларри Кая, Майка Кона, Пита Демера, Эстер Дерби, Ютту Экштейн, Кенджи Хиранабе, Клинтона Кита, Куроива-сан, Диану Ларсен, Тимо Леппянена, Эрика Линдли, Мэри Поппендик, Тома Поппендика, Кена Швабера, Маартена Смитса, Джеффа Сазерленда, Дэйва Томаса и Вилле Валтонена.

Нынешних и бывших членов команды (и рецензентов) компании Flexible, в том числе Кати Вилки, Петри Хаапио, Лассе Коскела, Пола Наги, Йоонаса Рейндерса, Габора Гунихо, Сами Лилья и Ари Тикка. И нынешних и бывших членов команды (и рецензентов) компании IPA LT, особенно Тero Пелтола и Люй И.

Бас признателен своей жене Сунь Юань за ее долготерпение в те периоды жизни, когда он «сосредоточивается на книгах», а не на ней, и за ее поддержку в ходе многочисленных переездов и разъездов по разным странам, где находятся его корпоративные клиенты. А также он благодарен Крэгу за стимулирующие дискуссии и годы совместной работы над крупными проектами, включая «отладку» организаций, и над книгами.

Крэг благодарит Альбертину за поддержку его писательской деятельности.

Мы также благодарим Луизу Адэр, Райну Хробак, Криса Гузиковски, Джули Нахил и Мэри Лу Нор за помощь в издании этой книги.

Глава 1

Введение

Будущее теперь уже не то, что раньше.

Йоги Берра

Мы сидели в переговорной комнате и пили горячий кофе. За окном — типичное для Северной Европы пронизывающее холодом зимнее утро. Вскоре вошел наш новый клиент и пожал нам руки. «Спасибо, что приехали, — сказал он. — Начнем с того, что у нас довольно небольшая группа разработки, человек 80, плюс-минус».

Однажды мы помогали с внедрением гибкой разработки в команде, которая мечтала вырасти до *очень* большого размера: до 12 человек.

У людей разные шкалы того, что значит «большая» продуктовая группа¹. Для кого-то это 50 человек или даже меньше. Для других — в разы больше. Мы обычно работаем с однопродуктовыми группами, насчитывающими 100–500 человек (которые внедряют принципы скрама, бережливого мышления и гибкой разработки, как правило, в области программно-интенсивных встраиваемых систем). Вот наш подход: *«Если собрать всю группу в одной комнате, вы сможете вспомнить имена всех ее членов? Если нет, значит, у вас большая группа»*. Таково наше определение

¹ Методология скрама (и эта книга) применима к разработке продуктов как для внешнего рынка, так и для внутреннего пользования.

«большой» продуктовой группы, основанное на фундаментальном критерии: нашей ограниченной памяти.

Наша ключевая рекомендация: проработав много лет с большими, распределенными и офшорными (удаленными) продуктовыми группами, мы проанализировали наш опыт и сделали следующий вывод: *не работайте с такими группами!*

Есть гораздо более эффективные способы разработки больших продуктов, чем задействовать рой разработчиков, рассеянных по разным местам. Лучше создать небольшую группу из крутых разработчиков и других специалистов, способных работать командами, хорошо им платить и разместить их в одном месте с продуктовым менеджером, который выступает в качестве «голоса клиента».

Но мы знаем, что вы не прислушаетесь к нашему совету и *все равно* будете использовать большие, распределенные и офшорные группы разработки. Причина в том, что ваша существующая система уже структурирована таким образом, или в том, что вы придерживаетесь (пред)убеждения, будто «большие системы требуют большого количества людей». Наши клиенты регулярно спрашивают нас: «Как рассчитать, сколько людей нам может потребоваться?» Мы всегда советуем: «Начните с небольшой группы сильных специалистов и увеличивайте ее только тогда, когда без этого действительно невозможно обойтись». Такая необходимость возникает редко.

Раз уж вы *все равно* будете использовать большие, распределенные и офшорные группы, мы хотим поделиться с вами нашим собственным и чужим опытом по применению принципов и практик бережливой и гибкой разработки в такой рабочей среде¹.

Инструменты мышления и организации

Когда Бас входил в руководящую команду одной большой продуктовой группы, он постоянно во время совещаний спрашивал: «Почему мы

¹ В книге «Практики масштабирования бережливой и гибкой разработки: Масштабированный скрам для больших, рассредоточенных и офшорных продуктовых групп» содержатся детальные практические советы касательно масштабирования и планирования, продукт-менеджмента, распределенной и офшорной разработки, контрактов, требований, проектирования и архитектуры, координации, унаследованных кодов, тестирования и многого другого.

делаем это именно так? Что будет, если мы сделаем это иначе?» Через несколько месяцев один из членов команды признался Крэгу: «Первое время эти вопросы выводили меня из себя. Но потом я оценил их значение». Бас не стремился вывести людей из себя; он хотел побудить их к *системному мышлению*, позволяющему: 1) увидеть более глубокую динамику организационной системы в целом; 2) понять, как система стала такой, какая она есть; и 3) пересмотреть предположения, лежащие в основе существующей организации.

Когда люди впервые слышат о скраме с его короткими, ограниченными по времени итерационными циклами разработки, они поначалу воспринимают это как локализованную практику инкрементального создания продукта за счет небольших управляемых шагов с использованием обучения и корректировки, которые ведут к поставленной цели. Поэтому они говорят: «О, аджайл меня не касается. Это для *разработчиков*». Но запуск таких циклов обучения на нижнем организационном уровне потенциально способен оказать гораздо более масштабное воздействие: как правило, это требует запуска аналогичных «петель» обучения и на более высоком уровне — создания обучающейся организации, где люди будут постоянно пересматривать структуры и правила, которые определяют и окружают разработку продуктов. Внедрение принципов скрама или бережливого подхода в очень больших продуктовых группах неизбежно ведет к такой высокоуровневой организационной трансформации.

Пример. Представим компанию, в которой подразделение разработки внедряет скрам, чтобы повысить адаптивность. Но отдел продаж продолжает работать по старинке: его сотрудники стремятся увеличить личные комиссионные и ежеквартальные продажи, с почти безграничным оптимизмом расписывая клиентам то, что «могут сделать наши самые лучшие разработчики», и обещая им достать звезды с неба. В результате подразделение разработки сталкивается с «обязательствами», которые оно не давало и не в состоянии выполнить; в компании его обвиняют в неисполнении *«наших обещаний»* и делают вывод, что «скрам не работает».

Если бы эта книга была посвящена внедрению скрама в небольшой однопродуктовой команде из двух десятков человек, системное мышление и организационные инструменты были бы интересными, но не критически важными темами. Однако они играют ключевую роль для успешного внедрения скрама, если данная методология масштабируется

на однопродуктовую группу, скажем, из 400 человек, которая, возможно, работает в составе крупной компании-разработчика, насчитывающей тысячи сотрудников, и совершает переход к скраму в тесном сотрудничестве с отделами продаж и доставки в условиях ограничений, налагаемых традиционными корпоративными правилами в отношении управления человеческими ресурсами, структуры команд, отчетности, оценки эффективности, управления проектами, контрактов и вознаграждения.

Следовательно, эта книга утверждает, что один из краеугольных камней успешного масштабирования скрама и гибкой разработки — люди, которые готовы учиться и применять необходимые *инструменты мышления*¹, включая системное мышление, анализ ментальных моделей, бережливое мышление, теорию массового обслуживания и распознавание ложных дихотомий (но не ограничиваясь этим).

Применение этих инструментов мышления позволяет увидеть, когда и где существующая организационная структура препятствует потоку создания ценности и требует реорганизации. Следовательно, второй краеугольный камень успешного масштабирования скрама, который рассматривается в этой книге, — *инструменты организации*, включая фича-команды, области требований, а также многие другие изменения в структуре, процессах, задачах, управлении людьми и системе вознаграждения.

Инструменты действия

Наряду с инструментами мышления и организации для успешной работы большой, распределенной и офшорной продуктовой группы требуются *инструменты действия* — конкретные практики разработки. Эффективное использование этих инструментов (которые подробно описаны в нашей книге «Практики масштабирования бережливой и гибкой разработки») отчасти зависит от организационной трансформации. Многие практики могут применяться и без более глубоких структурных изменений, но их эффективность в этом случае будет ограничена.

Таким образом, внедрение инструментария, описанного в этой книге, можно рассматривать как предварительное условие применения инструментов действия (практик), представленных в книге «Практики масштабирования бережливой и гибкой разработки». Но на деле *практики* всегда внедряются первыми — потому что людям проще начинать

¹ Термин «инструменты мышления» был популяризирован в [Reinertsen97].

именно с них. Тем не менее в какой-то момент вам все равно придется вернуться назад и воспользоваться организационными и мыслительными инструментами.

Мы рекомендуем коучам и другим агентам изменений, участвующим во внедрении скрама или бережливой разработки на больших масштабах, начать с внедрения в команде инструментов действия и параллельно самим как следует изучить инструменты мышления и организации. В какой-то момент ситуация созреет и люди будут готовы перейти от обсуждения «Как мы будем осуществлять полномасштабную непрерывную интеграцию?» к вопросам: «*Не мешают ли существующие правила управления персоналом созданию эффективных команд?*» и «*Каков в данном случае поток создания ценности и как ему может препятствовать существующая организационная структура?*».

Эксперименты: «Попробуйте... Избегайте...»

Скрам делает упор на эмпирический контроль процесса; в процессах разработки слишком высок уровень сложности и вариабельности, чтобы использовать шаблонные подходы. Поэтому инструменты в обеих наших книгах представлены в виде советов, начинающихся со слов: «Попробуйте...» или «Избегайте...». Это всего лишь эксперименты, и некоторые из них, вероятно, не сработают в ваших конкретных обстоятельствах. Подход в скраме, как и в бережливом мышлении *кайдзен*, состоит в том, чтобы сначала изучить и понять существующую ситуацию (*инспектировать*), а затем с помощью экспериментов найти способы, как ее улучшить (*адаптировать*). Нацеленность на непрерывное экспериментирование — один из столпов бережливого подхода. На самом деле самый неудачный эксперимент — тот, который не был проведен. В Toyota Тайити Оно, ключевой создатель концепции бережливого подхода, шел непосредственно на место и инспектировал все документы с описанием стандартов. Если те были в буквальном или переносном смысле покрыты пылью, давно не менялись, он заставлял людей пересмотреть их. Непрерывная эволюция «стандартов» была его требованием.

В скраме этот цикл инспекции и адаптации (на основе экспериментов) повторяется в каждой двух- или четырехнедельной итерации на протяжении всего времени, пока существует продукт. А в бережливом подходе

этот цикл постоянного экспериментирования и улучшения применяется как к отдельным продуктам, так и ко *всей организации в целом*.

Ограничения

Для нас есть еще большое поле для исследования и изучения в этой теме. В основе этой книги лежат наши сегодняшние (довольно ограниченные) опыт и понимание, которые мы рассчитываем углубить в ближайшие годы. Например, прожив несколько лет в Китае и Индии, мы только-только затронули верхушку в такой важной области, как мультикультурность в офшорных и распределенных группах разработки. Тем не менее мы надеемся на то, что наши советы окажутся для вас полезными, и будем благодарны, если вы поделитесь с нами своими историями и идеями.

Масштабное внедрение скрама способно повлиять практически на все аспекты организации и деятельности продуктоориентированной компании. Но, чтобы эта книга не получилась слишком пространной, а также из-за нашего ограниченного опыта в некоторых областях, мы лишь поверхностно затронули или же вовсе не стали рассматривать некоторые темы, которые, безусловно, заслуживают более пристального внимания, а именно:

- бюджетирование и финансы;
- продажи;
- маркетинг;
- разработка аппаратного обеспечения;
- разработка не-ИТ-продуктов;
- развертывание/поставка;
- сервисная поддержка.

В сущности, все написанное в этой книге применимо к разработке любых продуктов в целом. Скрам и бережливая разработка не ограничиваются только программным обеспечением [NT86]. Акцент на преимущественно программных системах (как правило, встраиваемых) объясняется только нашей сферой специализации, а также неуклонно растущей распространенностью ПО в повседневной жизни, включая стиральные машины и даже обувь.

В частности, в этой книге мы рассматриваем некоторые принятые в традиционных организациях предположения и правила, которые препятствуют *потоку создания ценности и эффективности команд*. В некоторых компаниях такой анализ может стать очень серьезным вызовом. Мы не хотим вас пугать, но организационная перестройка, призванная обеспечить поддержку бережливой и гибкой разработки, невозможна без отказа от многих традиционных моделей и радикального увеличения прозрачности. Организационные изменения также могут привести к ненужности старых ролей и перемещению с них талантливых людей. Как и в Toyota, мы поощряем поиск новых сфер приложения сил для компетентных людей, во-первых, потому что они этого заслуживают, а во-вторых, потому что в противном случае это будет тормозить перемены.

Это большая книга. Мы сожалеем, что, несмотря на все наши старания, не смогли сделать обсуждение масштабированного скрама... чуть менее масштабным.

Итак, поговорим об инструментах мышления.

Инструменты мышления

Глава 2

Системное мышление

*Я прошел курс скорочтения и прочитал роман
«Война и мир» за 20 минут.*

Он про Россию.

Вуди Аллен

«Что бы мы ни делали, количество дефектов в нашем бэклоге остается примерно тем же», — пожаловался нам один менеджер. Речь шла о продукте, насчитывающем 15 млн строк исходного кода C и C++, над которым работала группа из нескольких сотен разработчиков, внедрявшая принципы бережливой разработки. В чем же дело? Ответить на этот вопрос может только системное мышление. В небольших группах все действующие силы легко увидеть и понять без каких-либо формальностей, но при разработке крупного продукта — и в любых больших системах вообще — может быть действительно трудно. Джерри Вайнберг в этой ситуации выделяет два решающих фактора:

*Закон Вайнберга–Брукса: от действий менеджеров, основанных на **неправильных моделях системы**, пострадало больше проектов разработки программных продуктов, чем от всех остальных причин, вместе взятых.*

Ловушка причинно-следственной связи: у каждого следствия есть причина... и мы всегда можем точно определить причину и ее следствие [Weinberg92].

Эти факторы отражают влияние на систему наших **ментальных моделей**, о чем мы поговорим в этой главе чуть позже.

Проблемы, проистекающие из ментальных моделей и предположений, — это только один аспект. Другой состоит в том, что крупномасштабное внедрение скрама, принципов бережливого подхода и гибкой разработки не происходит изолированно только в группе разработки. Это затрагивает и такие области, как управление продуктами, бюджетирование, бета-тестирование, запуск, корпоративное управление и управление персоналом. Следовательно, при крупномасштабном внедрении

Стоит ли полагаться на здравый смысл?

«Здравый смысл говорит нам...» — эту фразу часто можно услышать не только в повседневной жизни, но и на встречах скрам-команд. Но что такое здравый смысл? Эйнштейн дал ему такое определение: «Здравый смысл — это сумма предубеждений, приобретенных человеком к 18 годам».

Тайити Оно, отец производственной системы Toyota, писал: *«Ошибочные представления легко превращаются в здравый смысл. Тогда споры становятся бесконечными. Каждая сторона пытается переспорить другую, и дело не движется с места. Вот почему [у нас в компании] был период, когда я постоянно говорил людям «выйти за рамки здравого смысла» и начать думать вне их. В рамках здравого смысла некоторые вещи могут казаться нам правильными, но только лишь из-за встроенных в него заблуждений. Кроме того, зачастую мы считаем необходимым руководствоваться здравым смыслом только по одной причине: потому что многолетний опыт показал, что данный образ, хотя и не несет больших преимуществ, не имеет и серьезных недостатков... Все мы — люди, а людей можно сравнить с ходячими наборами заблуждений, которые верят, что привычный способ делать вещи — лучший. И даже если вы не считаете его действительно лучшим, устоявшийся здравый смысл говорит вам: «Мы ничего не можем с этим поделать, так обстоят дела» [Ohno07].*

«Здравый смысл» не самый надежный способ понять нелинейные системы, такие как крупномасштабная разработка продуктов.

гибкого фреймворка не просто полезно, но и необходимо привлечь людей из этих областей к тому, чтобы *эффективно поразмышлять* о ментальных моделях, причинно-следственных связях, петлях обратной связи и механизмах контроля (или иллюзии контроля) в большой системе, которая вскоре будет серьезно *нарушена*. Системное мышление — инструментарий, который поможет это сделать.

В 1958 г. в журнале *Harvard Business Review* была опубликована новаторская статья «Индустриальная динамика: важный прорыв для принимающих решения лиц» Джея Форрестера, профессора Школы менеджмента Слоуна Массачусетского технологического института (МТИ) [Forrester58]. Эта статья положила начало преподаванию **системной динамики** в Школе менеджмента Слоуна, а вскоре обучение системному мышлению стало неотъемлемой частью бизнес-образования. Системная динамика иногда трактуется как синоним **системного мышления**, но последнее более общий термин.

В МТИ работали и другие выдающиеся исследователи в области системной динамики, например Питер Сенге¹.

В соответствии с законом Вайнберга–Брукса, исследование Форрестера показало, что, когда принимающим решения лицам представляли динамические модели бизнес-систем и предлагали улучшить их производительность, они *обычно делали их только хуже* [SKRRS94]. Исследование также показало, что большинство людей имеют слабое представление о том, как можно фундаментально улучшить системную динамику, и обычно применяют ошибочный «здравый смысл» и «быстрые решения», которые не приводят к устойчивым системным улучшениям.

Почему поведение большой группы разработки (как некой системы) так трудно понять и еще труднее грамотно им руководить? Ответ отчасти кроется в поведении стохастических систем с очередями и изменчивостью, которое будет рассмотрено в главе о теории массового обслуживания. Отчасти — в *теории управления*: большинство интересующих нас систем (например, большая группа разработки продукта) имеют сложные петли положительной и отрицательной обратной связи и нелинейное поведение. Поведение этих систем зачастую не поддается

¹ Сенге написал книгу «Пятая дисциплина», посвященную системному мышлению и обучающимся организациям, которую журнал *Harvard Business Review* назвал одной из основополагающих книг по менеджменту за последние 75 лет. См. [Senge94].

интуитивному пониманию. И наконец, отчасти — в такой «второстепенной» проблеме, как *люди*.

Таким образом, основные причины неспособности понять большую систему и грамотно ею руководить включают следующие факторы (но не ограничиваются ими):

- ❑ недостаток знаний о системной динамике, петлях обратной связи, поведении нелинейных систем и непредвиденных воздействиях в системах рабочей среды;
- ❑ непонимание корневых причин проблем (и того, как их найти) — именно *причин*, а не одной причины: системное мышление позволяет увидеть, что у проблем множественные, косвенные и динамические причины;
- ❑ непонимание того, как и почему «быстрые решения» или локальные решения на уровне отделов могут вести к ухудшению общей производительности системы.

Короче говоря, все дело в неспособности мыслить системно¹.

Эти причины могут иметь серьезные последствия, когда широкомасштабное внедрение принципов бережливого и гибкого подходов пересекаются с традиционным менеджментом. Управленческая команда является частью системы, которая подвергается трансформативному нарушению, и, если ее члены не применяют системное мышление, они сами могут стать серьезным нарушающим фактором — в плохом смысле.

В качестве краткого описания системного мышления нам нравятся следующие 11 «законов», описанных в книге Питера Сенге «Пятая дисциплина»:

1. Сегодняшние проблемы вызваны вчерашними «решениями».
2. Сила воздействия на систему равняется силе ее противодействия.
3. Прежде чем улучшиться, поведение системы ухудшается.
4. Легкий путь обычно ведет назад.
5. Лечение может быть хуже болезни.

¹ Еще одна причина — вера в возможность большего контроля, чем это возможно на самом деле. Наука о комплексных системах показывает существование фундаментальных ограничений в прогнозировании и управлении полухаотическими социальными системами [Stacey07]. Это довольно большой ящик Пандоры, который мы не будем открывать в этой книге.

6. Тише едешь, дальше будешь.
7. Причины и следствия могут быть разделены во времени и пространстве.
8. Малые изменения могут привести к значительным результатам, но такие области с наивысшей отдачей не всегда очевидны.
9. Можно погнаться за двумя зайцами и поймать обоих — пусть не одновременно.
10. Разделив слона пополам, вы не получите двух маленьких слоников.
11. Не нужно искать виноватых.

Внутренний девиз Toyota: «Правильное мышление, хорошие продукты». Системное мышление — это *набор* инструментов мышления, которые помогают:

□ **Видеть системную динамику:**

- организация, занимающаяся разработкой, — это система, образованная людьми и правилами, с неявными петлями обратной связи и непреднамеренными воздействиями;
- мы можем научиться видеть эту динамику и, следовательно, улучшать систему с помощью **диаграмм причинно-следственных циклов** (*causal loop diagrams*), создаваемых на групповых встречах.

□ **Видеть ментальные модели:**

- одна из причин неоптимальных решений — ложные предположения и ошибочные выводы;
- ментальные модели обнаруживаются с помощью диаграмм причинно-следственных циклов и метода «Пяти почему».

□ **Выявлять корневые причины:**

- чтобы реально улучшить систему, необходимо научиться выявлять корневые причины проблем и видеть более глубокие связи;
- выявить корневые причины помогают диаграммы причинно-следственных циклов, метод «Пять почему» и диаграммы Исикавы.

□ **Видеть локальную оптимизацию:**

- еще один источник неоптимальных решений — **локальная оптимизация**, когда принимается решение, «лучшее» с точки зрения отдельного человека или отдела, в противовес **глобальной оптимизации**, исходящей из цели бережливого подхода системного уровня — *создавать ценность быстро, с высоким качеством и высоким моральным духом.*

Итак, эта глава построена вокруг следующих областей системного мышления: учимся видеть: (1) *системную динамику*, (2) *ментальные модели*, (3) *корневые причины* и (4) *локальную оптимизацию*.

Учимся видеть системную динамику

Статическая сложность против динамической сложности

Многие из нас, особенно инженеры и финансисты, обучены хорошо справляться со **статической сложностью** — решением таких задач, как анализ и управление информацией (требованиями, финансовыми данными и пр.), разбивка сложных структур на более простые и т.д., то есть со сложностью статического, информационного или структурного характера.

Почему большие программные системы имеют тенденцию деградировать, несмотря на то что на работу с дефектами тратится все больше и больше времени? Что может случиться, если США вторгнутся в Ирак? Чтобы увидеть динамику, стоящую за этими вопросами, требуется анализ **комплексной динамики**.

В отличие от обучения статическому анализу, многие из нас не получают *формального* образования в области анализа *динамической сложности*¹, особенно динамики рабочей среды. Возможно, причина этого — в распространенном мнении, что здесь достаточно полагаться на здравый смысл. Но Форрестер продемонстрировал, что «здравый смысл» не справляется с комплексными системами, и показал, что обучение людей формальным методам, в частности таким, как использование *моделей системной динамики*, визуализированных в виде *диаграмм*

¹ Среди исключений — макроэкономика, психология, социология и биология.

потоков [Forrester61], позволяет улучшить их способность к системному мышлению в рабочей среде.

Диаграмма потоков охватывает материальные, финансовые и информационные потоки, метрики (переменные, имеющие количественные значения, такие как финансовые средства или количество дефектов), последствия решений и правил, а также причинно-следственные связи. Популярное упрощение — **диаграмма причинно-следственных циклов**, которая фокусируется на действующих в системе причинно-следственных отношениях и петлях обратной связи [Sterman00]. Существует несколько вариантов нотаций таких диаграмм; во всех них присутствуют метрики (количественные переменные), причинно-следственные связи и задержки (отложенные эффекты). Вайнберг [Weinberg92] называет это *диаграммой влияний*.

Первый закон построения диаграмм: моделируйте, чтобы обсуждать

Чтобы научиться видеть системную динамику, используйте диаграммы причинно-следственных циклов, которые в идеале следует нарисовать на доске совместно с коллегами в ходе ретроспективы спринта. Но прежде чем двигаться дальше, давайте рассмотрим *первый закон построения диаграмм*:

Основная ценность диаграмм заключается в дискуссии, возникающей в процессе их построения: мы моделируем, чтобы инициировать обсуждение.

Когда группа собирается, чтобы нарисовать на доске диаграмму причинно-следственных связей (рис. 2.1), основная ценность состоит в обсуждениях и общем понимании, которое зарождается в ходе построения модели. Визуализация в виде наглядной диаграммы также *важна*, чтобы сделать идеи конкретными и однозначными — явно отобразить ментальные модели, которые есть у участников, — потому что слова сами по себе могут быть расплывчатыми и неправильно интерпретируемыми. Тем не менее сама диаграмма вторична по отношению к тому, что люди уносят с собой: новые идеи и более глубокое понимание, пересмотренное в результате обсуждения.



Рис 2.1. В Valtech India: дискуссии и осмысление — самое важное в процессе построения диаграмм

Базовые проблемы и простые удобные инструменты

С годами мы овладели очень продвинутыми, сложнейшими приемами анализа и разработки решений в области инженерии, менеджмента и т.д. Поначалу мы были вдохновлены этими инструментами и стремились применять их и обучать им других людей при любом случае, пока в ходе реальной работы не поняли, что:

*подавляющее большинство проблем в бизнесе (в том числе в разработке) носят настолько базовый характер¹, что ключевым решением является обучение и планомерное использование **простых и удобных инструментов мышления и действия.***

Простота. Книги и курсы по моделированию системной динамики и построению диаграмм причинно-следственных циклов имеют тенденцию чрезмерно все усложнять, предлагая забираться в такие дебри, как концепция *архетипов*, описанная в «Пятой дисциплине», компьютерное моделирование, нелинейные уравнения и т.д. На практике это только отпугивает людей от использования этого, в сущности, очень простого инструмента: вам нужно всего лишь встать у доски, обсудить

¹ «Базовая» не означает простая или неважная. Например, мотивация и качество — это базовые, но далеко не простые и не пустяковые проблемы.

со своими коллегами, какие причинно-следственные связи и петли обратной связи действуют в вашей системе, и отобразить их в виде диаграммы на доске.

Когда вы выбираете инструменты мышления или действия для использования в рабочей среде, помните слова Вольтера:

«Le mieux est l'ennemi du bien» («Лучшее — враг хорошего»).

Удобство. Существует множество изощренных инструментов мышления и действия, которые так любят теоретики, но которые почему-то никак не приживаются на практике. Вместе с тем практики скрама или экстремального программирования (XP), будучи раз внедренными, становятся очень «липкими». Почему? Во-первых, они быстро приносят реальную отдачу: у них привлекательное соотношение затрат и выгод и их внедрение быстро окупается. Во-вторых, эти практики не требуют мучительных усилий; большинство людей, которые их используют, говорит, что применяет их с интересом и даже с удовольствием (и, разумеется, с пользой). Люди есть люди: они предпочитают делать то, что им нравится.



Удобство и простота инструментов особенно важны в крупномасштабных разработках, так как *устойчивое внедрение* практик и процессов становится тем труднее, чем больше размер группы. Ваши инструменты должны *привлекать* людей, как яркие ароматные цветы притягивают к себе пчел.

Диаграммы причинно-следственных циклов

В этой книге мы будем не раз обращаться к диаграммам причинно-следственных циклов, потому что они помогают увидеть динамику того, что происходит при крупномасштабной разработке в больших группах. Уже только по одной этой причине полезно разобратсья в этих диаграммах. Чтобы они были еще полезнее, мы рекомендуем:

Попробуйте... чертить диаграммы причинно-следственных циклов на групповых встречах, чтобы увидеть системную динамику.

Попробуйте... чертить диаграммы причинно-следственных циклов на доске вместе с коллегами

Практическая ценность этого совета намного важнее, чем может показаться на первый взгляд. Фраза «нам нужно системное мышление» слишком расплывчата и малоэффективна. Но, если вы со своими коллегами сделаете своей привычкой вместе вставать у большой доски и рисовать диаграммы причинно-следственных связей, это будет уже конкретная практика, которая превратит абстрактный призыв к системному мышлению в реальный системный подход — с потенциально далекоидущими последствиями.

Приведенные далее примеры могут показаться «теоретическими», когда они изложены на бумаге. Но представьте, что эти диаграммы были составлены вами в процессе живого общения с коллегами, когда вы вместе стояли у доски и горячо обсуждали каждый момент. Именно так мы рекомендуем применять системное мышление на практике.

Конкретный совет по построению диаграмм. Мы начинаем с того, что записываем на стикерах задействованные *переменные*, например «скорость разработки фич» или «количество дефектов», и размещаем стикеры на доске. Затем проводим между стикерами линии причинно-следственных связей. В процессе построения модели всегда приходится много переписывать, стирать, перерисовывать. Самый важный результат таких сессий — *совместное понимание*. Некоторые участники захотят сфотографировать то, что получилось на доске.

Нотации и примеры

Диаграммы причинно-следственных циклов содержат много элементов; вот список наиболее типичных и полезных, которые будут рассмотрены дальше в сценарии:

- переменные;
- причинно-следственные связи;
- обратные эффекты;
- ограничения;
- цели;
- реакции;
- «быстрые решения»;