

Предисловие

Маркс видел в пианисте производителя продукта, который не является товаром. А потом появилась звукозапись, и искусство пианиста стало товаром.

Джон Марков (John Markoff)

Эта книга посвящена индустриализации разработки программного обеспечения (ПО). Краткий обзор достижений последнего десятилетия показывает лишь незначительный рост производительности разработки ПО или качества программных продуктов. Большая часть программ в наши дни по-прежнему создается вручную, начиная с чистого листа, с применением трудоемких методов. В результате разработка программ ведется медленно, стоит дорого и порождает продукты, содержащие серьезные дефекты, которые ведут к проблемам удобства использования, надежности, производительности и безопасности. Эти проблемы — симптомы успеха, порожденного повышенными требованиями заказчиков. Однако по мере взросления индустрии ПО, она неизбежно должна вырасти за рамки методов, приведших ее к такому состоянию, и перенять подходы, которые демонстрируют соседние области человеческой деятельности. Сюда входит сборка продуктов из компонентов, автоматизация рутинных или вспомогательных задач, формирование линеек продуктов и применение изменений и стандартизация процессов, архитектур и форматов пакетирования. В этой книге мы представим путеводитель для перехода программной индустрии от кустарного производства к промышленному, и опишем наиболее существенные новшества, которые обеспечивают возможность такого перехода уже сегодня.

Кратко о фабриках программ

Большую часть последних двух десятилетий программная индустрия была сосредоточена на системном ПО и ПО, призванном повысить персональную производительность. Теперь она обратила свое внимание на приложения для автоматизации бизнес-процессов. Это изменение приоритетов озаменовано растущим уровнем инвестиций в инструменты разработки бизнес-приложений и платформы для их исполнения.

Бизнес-приложения — это новый рубеж.

По мере того, как разные отрасли бизнеса требуют все более высоких уровней автоматизации, они также нуждаются в гораздо более широком наборе функциональности и предъявляют более строгие требования к качеству обслуживания, с тем, чтобы обеспечивать целостность и доступность важных данных и операций с ними. Это ведет к усложнению архитектуры. В связи с происходящей в настоящее время миграцией от многослойных архитектур к архитектурам, ориентированным на службы, программы должны теперь обеспечивать безопасный доступ

Повышение уровня автоматизации приводит к росту сложности архитектуры, к новым требованиям и ускоряет эволюцию платформ.

партнерам и заказчикам для поддержки интеграции бизнеса. В то же время требования изменяются все быстрее, поскольку глобальные коммуникации разрушают традиционные барьеры конкуренции, и позволяют заказчикам и партнерам более тесно участвовать во внутренних бизнес-процессах. Приложения должны выходить на рынок быстрее и меняться чаще, нежели раньше. Технология платформ также развивается быстрее для поддержки этих новых требований, изменяя традиционное представление о разработке непрерывным инкрементным процессом сопровождения и расширения.

Наверное, не удивительно, что эта индустрия имеет проблемы с удовлетворением ожиданий. Одной из наиболее явных проблем является острый недостаток квалифицированных разработчиков. Узкое место высококвалифицированного труда приводит к быстрому росту затрат, задержке выхода продуктов на рынок и постоянным проблемам с надежностью, безопасностью и производительностью. Конечно, разработка программ всегда была непростой задачей, и основные факторы, делающие ее такой, остаются прежними. В самом деле, историю разработки ПО можно рассматривать как непрерывную борьбу со сложностью и изменчивостью, где приобретения чередуются с потерями, когда заказчики реагируют на новшества, выдвигая все возрастающие требования. Как это случается время от времени в процессе этой борьбы, в данный момент мы находимся в положении, когда современные методы и приемы перестают отвечать растущим требованиям, и возникает необходимость в новых.

Как и раньше, истощение возможностей существующих методов и приемов доказывает возрастающий уровень неудовлетворенности, вызванный множеством хронических проблем, упорно сопротивляющихся решению. На этот раз проблемы вызваны сложностью обнаружения и ликвидации брешей в безопасности, сложностью создания компонентов, которые одинаково хорошо работают и в лабораторных условиях, и в среде промышленной эксплуатации, невозможностью достижения коммерчески оправданного уровня повторной применимости, несмотря на повсеместное распространение объектной технологии, а также трудностью обеспечения правильного баланса между ловкостью индивидуальных команд разработчиков и обеспечением поддержки межкомандной кооперации. Конечно, эти проблемы являются только симптомами других, более глубоких проблем, отражающих фундаментальные изъяны в принятом способе построения ПО. Как будет показано в настоящей книге, коренные причины неудач можно разбить на четыре основных категории: монолитная конструкция, чрезмерная обобщенность, одноразовая разработка и незрелость организации процесса.

Конечно, мы написали эту книгу не только для того, чтобы документировать проблемы существующих методов и приемов, но также и для того, чтобы предложить пути их решения. Ясно, что значительный прогресс достигнут в снижении затрат времени и усилий, необходимых для построения ПО, а также в повышении удобства, надежности, производительности и возможности сопровождения полученных в результате продуктов. Некоторые задачи в наше время решаются гораздо легче, чем когда-либо ранее. Например, извлечение из хранилища данных, удовлетворяющих определенным критериям, теперь на-

Фундаментальными движущими силами в разработке программ является сложность и изменчивость.

Устаревание текущих методов и приемов подтверждается хроническими проблемами.

Рекуррентными моделями, обеспечивающими существенный прирост производительности, являются задачи автоматизации разработок ПО путем интегрирования языков, шаблонов, структур и инструментальных средств.

много проще с появлением систем управления базами данных и декларативных языков запросов. Аналогично, конструирование графического интерфейса пользователя теперь намного легче, нежели это было раньше, до появления каркасов интерфейса на основе оконных элементов управления (“виджетов”) и редакторов WYSIWYG (What You See Is What You Get — “что видишь, то и получаешь”). Присмотревшись внимательно к тому, как достигнут этот прогресс, мы обнаружим повсеместное использование рекуррентной модели, интегрирующей четыре приема автоматизации задач разработки ПО. Была разработана структура для обеспечения разработки продуктов на основе общего архитектурного стиля путем адаптации, конфигурирования и сборки компонентов каркаса. Затем был определен язык для поддержки процесса сборки, автоматизированный инструментальными средствами. Инструменты помогли разработчикам вовлечь в процесс заказчиков и быстро реагировать на изменения их требований, организуя построение программ инкрементным способом, поддерживая их построение и функционирование по мере внесения изменений, фиксируя проектные решения в форме, непосредственно производящей исполняемые программы.

В настоящее время применение такого рода автоматизации чрезвычайно дорого и экономически оправдано только в широких “горизонтальных” областях применения — таких, как конструирование пользовательского интерфейса и доступ к данным. Мы намерены показать, как можно использовать новые технологии для радикального снижения затрат на автоматизацию подобного рода, с тем, чтобы экономически оправданно достичь повышения производительности в узких “вертикальных” областях, подобных здравоохранению и финансовым службам.

Новые технологии снижают стоимость автоматизации.

Наш подход основан на концепции, называемой фабриками программного обеспечения. Фабрика ПО, как мы определяем этот термин — это конфигурация языков, шаблонов, каркасов и инструментов, которые могут быть использованы для быстрого и недорогого производства открытого набора уникальных вариантов первоначального продукта. Однако фабрики ПО — не только средство автоматизации разработки программ внутри индивидуальной организации. Широкое применение фабрик ПО обеспечит возможность проведения изменений, распределяя затраты и риск по сети специализированных независимых поставщиков. Оно также повышает степень настраиваемости программного обеспечения за счет увеличения уровня последовательной интеграции внутри каждого поставщика.

Фабрики ПО производят уникальные варианты первоначального продукта.

Всеобъемлющие изменения, как индустрия

Одним из наиболее серьезных вызовов промышленного масштаба является значительный разрыв между старыми и новыми способами и решениями задач. Языки, шаблоны, каркасы и инструменты, обеспечивающие автоматизацию сегодня, производятся поставщиками платформ и потребляются остальными субъектами промышленности. Мы оказались в ситуации, когда экономическая модель перестает работать. Для того чтобы обеспечить дальнейший рост производительности, все эти средства должны быть в большей степени сосредоточены вертикально, обеспечивая решение узких классов задач.

Дальнейший рост производительности требует другой экономической модели.

И это создает две проблемы: 1) рынки для “вертикальных” решений намного уже, чем рынки для решений “горизонтальных”, и 2) знания о предметной области сосредоточены в основном в организациях конечных пользователей, а не у традиционных поставщиков программного обеспечения.

Как правило, отрасли промышленности сопротивляются переменам. И производство ПО, не будучи исключением, долго сопротивлялось индустриализации. Как последний дом с террасой в квартале, застроенном небоскребами, она остается “индустрией коттеджей”, полагающейся на мастерство квалифицированных умельцев, занимающихся трудоемкой ручной работой, в то время как их соседи уже перешли на более современные методы. Мы полагаем, однако, что наступило, наконец, время индустриализации, и область разработки программного обеспечения стоит на пороге перемен, которые полностью преобразят ее за следующие 10 лет.

Некоторые из наших коллег смотрят на вещи иначе. Некоторые говорят, что разработка программ не может быть переведена на промышленную основу из-за ее творческой природы. Другие считают, что мы недооцениваем потенциал существующих методов и приемов. Все это понятно. Квалифицированные разработчики действительно могут производить высококачественное программное обеспечение вручную, с нуля, не выходя за рамки сроков и бюджета. Проблема, однако, в том, что эти мастера являются исключением, а не правилом.

Поэтому мы призываем наших коллег внимательно прислушаться к нашим выводам, и не отвергать их с порога как ошибочную попытку свести изначально творческую дисциплину к механическим и детерминистическим процессам. Совсем наоборот, мы предлагаем сделать разработку программ еще более творческим за счет применения методов, которые призваны приблизить их к проблемам реального мира, которые они призваны решать, оставив большую часть механистических и детерминистических аспектов процесса инструментальным средствам. С этой целью мы попытались идентифицировать проблемы, сдерживающие продвижение вперед, а также найти решения, которые позволят справиться с требованием все более компьютеризованного общества.

Изменения, которые мы предлагаем, подобны изменениям, которые пришли на смену старым методам и практическим приемам прошлых поколений, давшие нам то, что мы имеем сегодня. Многие еще помнят те времена, когда объектно-ориентированный подход рассматривался как угроза всей отрасли, и каким болезненным и медленным оказалось его внедрение. Теперь, конечно, объекты — основное направление, и это несомненный признак того, что перемены назрели. Сейчас, как и тогда, большая часть отрасли оказывает сопротивление странно звучащим идеям, которые представляют собой угрозу для установившейся практики. Но сейчас, как и тогда, идеи, которые ускоряют, облегчают и удешевляют разработку программного обеспечения, в конечном итоге изменят лицо отрасли.

Индустрия ПО находится на пороге серьезных перемен.

Идеальный проект — исключение, а не правило.

Мы призываем коллег внимательно прислушаться к нашим выводам.

Идеи, которые делают разработку программного обеспечения быстрее, легче и дешевле, целиком изменят индустрию.

Путеводитель по остальной части книги

Путеводитель, показанный на рис П.1, показывает, какое место в общей структуре книги занимает каждая глава. Как следует из этой иллюстрации, данная книга делится на три части.

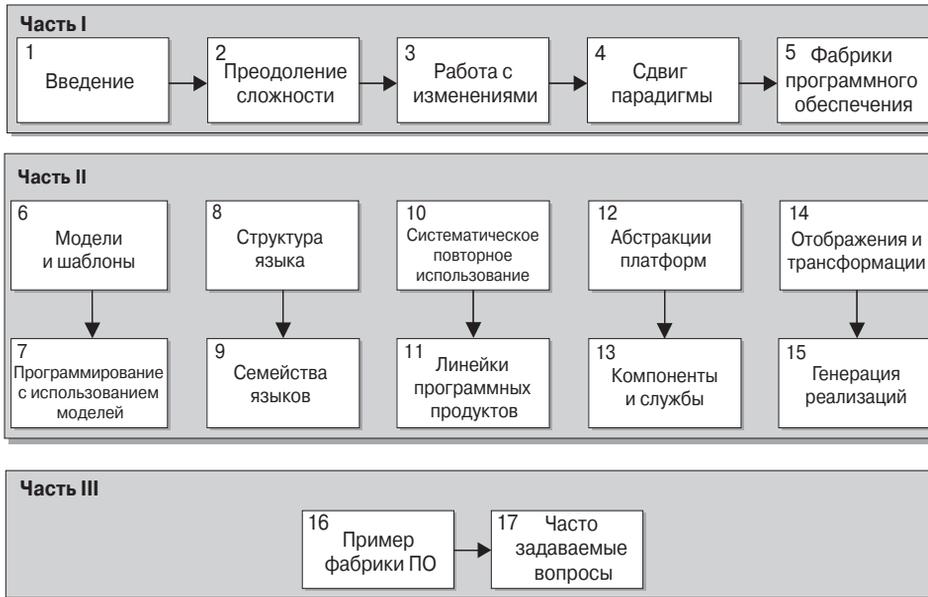


Рис. П.1. Путеводитель по остальной части книги

Часть I представляет фабрики ПО. В главе 1 показан пример, используемый на протяжении всей книги, и в ней перечислены проблемы, с которыми приходится сталкиваться разработчикам программного обеспечения. Глава 2 посвящена проблеме сложности и в ней рассматриваются существующие методы и приемы преодоления ее. Глава 3 описывает проблему изменений и там анализируются существующие методы и приемы ее решения. В главе 4 описаны хронические проблемы, которые объектно-ориентированный подход преодолеть не в состоянии, и в ней представлены критичные новшества, составляющие основу фабрик ПО.

В главе 5 подробно описаны фабрики ПО и представлено наше видение индустриализации.

Часть II состоит из нескольких потоков, каждый из которых описывает одно из основополагающих новшеств, представленных в части I. Хотя потоки можно читать параллельно, главы в каждом потоке следует читать последовательно.

- В главах с 6 по 9 обсуждается разработка на основе моделей. Главы 6 и 7 описывают модели, шаблоны и программирование с моделями. Главы 8 и 9 раскрывают анатомию языка и инструменты для построения семейств языков.

Часть I представляет фабрики ПО.

Часть II посвящена наиболее важным новшествам, представленным в части I и содержит потоки, которые можно читать параллельно.

- Главы 10 и 11 посвящены разработке на базе семейств. Глава 10 закладывает фундамент для описания систематического повторного использования. В главе 11 описаны линейки программных продуктов — одно из наиболее важных нововведений.
- В главах 12 и 13 обсуждается разработка с применением сборок. Глава 12 посвящена абстракциям, базирующимся на платформах, включая классы, библиотеки и каркасы, а глава 13 — компонентам и службам, а ней также представлена концепция сборки “оркестровкой”.
- В главах 14 и 15 рассматриваются вопросы обобщенного программирования. В главе 14 рассказывается об отображениях и трансформациях. В главе 15 затем рассматривается то, как на основе моделей может быть сгенерирован код и другие артефакты реализации.

В части III пересматриваются концепции, представленные в части I, на основе материалов части II, и подводятся определенные итоги. В главе 16 предложен пример фабрики ПО, а глава 17 содержит ответы на часто задаваемые вопросы. Есть еще два приложения, следующие за главой 17. В приложении А описаны формы абстракции и конкретизации, а в приложении Б — сильные и слабые стороны использования UML (унифицированного языка моделирования).

Для краткого обзора фабрик ПО читайте начало главы 1, главу 5, главу 16 и главу 17. Для обзора важнейших новшеств, которые делают их возможными, читайте главу 4. Для проникновения в суть проблем современных методов и приемов, преодолеть которые призваны фабрики ПО, читайте остальной материал глав 1, 2 и 3. Детальные описания важнейших инноваций приведены в главах с 6 по 15.

Часть III подводит книгу к заключению.

Как читать эту книгу.

Дополнительная информация

За дополнительной информацией о фабриках ПО обращайтесь на Web-сайт, посвященный этой теме, доступный по адресу <http://www.softwarefactories.com>.