

Предисловия

Создание программного обеспечения для предприятия — непростое дело. Несмотря на наличие множества инструментов и базовых средств разработки, упрощающих эту задачу, нам по-прежнему приходится решать, как использовать эти средства наилучшим образом. Для этого имеется немало подходов, но самое главное — знать, какой из них применять в конкретной ситуации, и вряд ли один подход пригодится на все случаи жизни. За последние несколько лет появилось целое поколение людей, занимающихся поиском подходов к проектированию приложений для предприятия и их документированию в виде шаблонов (соответствующий обзор можно посмотреть по адресу <http://martinfowler.com/articles/enterprisePatterns/html>). Те, кто занимается данной проблемой, в том числе и я, пытаются найти общие подходы и показать в своих трудах, как и когда их лучше всего применять. Результаты такой работы сильно разнятся, и поэтому читатель оказывается перед очень широким выбором.

Когда я приступил к работе над книгой *Patterns of Enterprise Application Architecture* (изд-во Addison-Wesley, 2002 год. *Архитектура корпоративных программных приложений*, пер. с англ. ИД “Вильямс”, 2004 г.), то искал подобного рода рекомендации для проектирования в литературе по программным продуктам Microsoft. Я пытался найти хоть что-нибудь достойное внимания, но в конечном итоге обнаружил лишь одну редкую работу на данную тему — предыдущую книгу Джимми. Мне понравилась информативная манера изложения и стремление автора вникнуть в понятия, которые многие другие рассматривали весьма поверхностно. Именно Джимми решил воспользоваться идеями, выдвинутыми мной и другими авторами, и показать, как воплощать их в приложениях, создаваемых в .NET.

Основное внимание тех, кто занимается шаблонами для предприятий, сосредоточено на документировании удачных проектов. Мы увлечены и другим интересным направлением, являясь большими поклонниками гибких методов, охватывающих в том числе разработку посредством тестирования (РПТ — Test-Driven Development (TDD)) и рефакторинг кода. Эти идеи Джимми изложил в своей книге. Многие находят определенное противоречие в том, что те, кто занимается шаблонами, уделяют основное внимание проектированию, а те, кто занимается РПТ, — эволюции. Тот факт, что деятельность и тех и других во многом пересекается, свидетельствует о том, что это не так, и Джимми попытался в своей книге связать воедино оба эти направления.

В итоге получилась книга о гибких методах проектирования в среде .NET с применением готовых шаблонов для предприятия. В этой книге показано, как следует применять такие методы, как РПТ, объектно-реляционное преобразование и проек-

тирование на основе предметной области, в проектах, выполняемых в .NET. Если вы еще не сталкивались с подобными понятиями, эта книга послужит вам в качестве введения в методы, которые многие разработчики считают основными для разработки программного обеспечения в будущем. Если же вы знакомы с этими идеями, книга поможет вам поделиться опытом со своими коллегами.

Многие считают, что в сообществе разработчиков приложений на платформе Microsoft не так хорошо, как у остальных, распространяются полезные советы по проектированию приложений для предприятия. По мере усложнения и расширения возможностей применяемой технологии все более актуальной становится проблема ее эффективного применения. И эта книга представляет собой важный шаг на пути к пониманию данной проблемы.

Мартин Фаулер (Martin Fowler), <http://martinfowler.com>

Самый лучший способ научиться проектированию на основе предметной области (ППО — Domain-Driven Design (DDD)) — сесть рядом с дружески настроенным, терпеливым, опытным практиком и поэтапно решать вместе с ним конкретные задачи. Именно так и происходит при чтении этой книги.

Эта книга не открывает ничего особенно нового. Она лишь беспристрастно сообщает о личном опыте специалиста-практика и ряде современных практических методов, которые привлекли его внимание.

Джимми Нильссон повторяет то, что многие из нас уже говорили не раз: ряд современных направлений, в частности, ППО, шаблоны архитектуры приложения для предприятия (РоЕАА) и разработка посредством тестирования (РПТ) не являются альтернативой друг другу, а взаимно дополняют друг друга как элементы успешной разработки.

Более того, все три направления оказываются сложнее, чем кажется на первый взгляд. Они требуют обширных и глубоких знаний. В этой книге уделяется некоторое внимание пропагандированию данных подходов, но большая ее часть все же посвящена подробностям их осуществления на практике.

Эффективное проектирование не сводится к набору заученных методов. Это способ мышления. Рассматривая конкретный пример, Джимми раскрывает перед нами ход своих мыслей. Он не только показывает и поясняет свое решение, но и дает нам возможность увидеть, как он к нему пришел.

Когда я что-то проектирую, в моей голове мелькают десятки соображений по этому поводу. Если эти мысли относятся к факторам, с которыми мне часто приходится иметь дело, они проносятся в моей голове так быстро, что я едва ли успеваю их осознать. Если же они касаются областей, где я чувствую себя не столь уверенно, я останавливаюсь на них дольше. Я вполне допускаю, что такой ход мыслей типичен для разработчиков, но передать его другому человеку не так-то просто. Поясняя примеры, Джимми как бы замедляет ход своих мыслей до такой степени, чтобы за ними можно было уследить. В каждое мгновение этого процесса две или три альтер-

нативы представляются, взвешиваются и отбрасываются в пользу одной, выбираемой окончательно.

Допустим, что требуется смоделировать объекты, реализуемые без осложнений, связанных с технологией сохраняемости. Каковы же те восемь способов, которыми базовая структура сохраняемости способна усложнить реализацию объекта предметной области? Какие соображения позволяют прийти к компромиссу в ряде этих вопросов? И какие ограничения накладывают современные базовые средства разработки, включая .NET?

Джимми мыслит прагматично. Принимая проектное решение, он обращается к своему опыту, который, в конечном итоге, приводит его к желанной цели. При этом он придерживается более основательного принципа, чем выбор, который слишком похож на хрестоматийный пример, а все его решения имеют предварительный характер.

Главный принцип проектирования, которого Джимми придерживается в первую очередь, составляет основную цель ППО — проектирование, отражающее глубокое понимание решаемой экономической задачи в такой форме, которая допускает приспособляемость к новым идеям. Так зачем столько рассуждать о технической структуре и архитектуре?

Типичное и, вероятно, естественное заблуждение состоит в том, что такое предпочтение предметной области не требует особых инженерных способностей и технических навыков. Если бы это было так на самом деле, стать компетентным разработчиком приложений для предметной области было бы нетрудно. Как ни странно, для воплощения четких и пригодных принципов предметной области в программном обеспечении, причем так, чтобы они не оказались по спудом массы технических вопросов, требуется особое умение применять конкретную технологию. Мои наблюдения показывают, что те, кто лучше всего владеет техническими и архитектурными принципами, зачастую знают, как применять технологию к месту, и относятся к числу наиболее эффективных создателей моделей предметной области.

Я имею в виду не знание всех особенностей сложных инструментов, а овладение теми знаниями, которые изложены книге *Архитектура корпоративных программных приложений* Мартина Фаулера, поскольку наивное применение технологии, как ни странно, приводит к ее насильственному проникновению в приложение.

Эта книга позволит многим читателям восполнить пробел в реализации выразительных объектных моделей на практике. Я лично извлек для себя ряд полезных способов продумывания вариантов применения технических структур и укрепил свои представления об особенностях выполнения ППО в среде .NET.

Помимо технической архитектуры, Джимми уделяет в своей книге много внимания написанию тестов. РПТ по-разному дополняет ППО. В отсутствие должного внимания к уточнению модели с целью ее улучшения, РПТ находит фрагментарное применение, где целенаправленное решение каждой проблемы в отдельности приводит к появлению негибкой системы. Полный тестовый набор фактически замедляет продвижение группы разработчиков к конечной цели, но именно в этом и заключается наибольшая ценность РПТ.

В наилучшем варианте тестовый набор представляет собой лабораторию для модели предметной области и ее технического выражения на универсальном языке. Особого рода тесты продвигают процесс моделирования вперед и сохраняют его сосредоточенный характер. В книге рассматриваются примеры поэтапной разработки таких текстов.

Джимми Нильссону присуще редкое сочетание уверенности в себе и скромности — свойств, которые, по моим наблюдениям, характерны для самых лучших разработчиков. Когда он рассказывает нам, читателям, как и почему изменились его представления о том, во что он раньше верил, мы можем проследить, как он пришел к своим нынешним представлениям, и это помогает нам пропустить делала применяемых методов и перейти непосредственно к основополагающим принципам. Скромность Джимми делает его подверженным самым разным влияниям, а мы получаем в итоге сплав идей из различных источников. Он испробовал много разного, результаты этих проб и его опыт стали для него руководством к действию. Его выводы отображают нынешнее его представление о предмете, причем с ясным пониманием того, что наши знания никогда не будут полными. Все это делает его рекомендации еще более полезными для читателя. Ответственное отношение к делу как нельзя лучше характеризует его как ведущего и успешного специалиста в области разработки программного обеспечения.

Эрик Эванс (Eric Evans)