

---

---

# Рефакторинг баз данных

*эволюционное  
проектирование*

---

---

# Refactoring Databases

## *Evolutionary Database Design*

Scott W. Ambler  
Pramodkumar J. Sadalage



**ADDISON-WESLEY**

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco  
New York • Toronto • Montreal • London • Munich • Paris • Madrid  
Capetown • Sydney • Tokio • Singapore • Mexico City

---

# Рефакторинг баз данных

*эволюционное  
проектирование*

Скотт В. Эмблер  
Прамодекмар Дж. Садаладж



Москва • Санкт-Петербург • Киев  
2007

ББК 32.973.26-018.2.75

С14

УДК 681.3.07

Издательский дом “Вильямс”

Зав. редакцией *С.Н. Тригуб*

Перевод с английского и редакция *К.А. Птицына*

По общим вопросам обращайтесь в Издательский дом “Вильямс” по адресу:

info@williamspublishing.com, <http://www.williamspublishing.com>

115419, Москва, а/я 783; 03150, Киев, а/я 152

**Эмблер, Скотт В., Садаладж, Прамодкумар Дж.**

С14 Рефакторинг баз данных: эволюционное проектирование. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2007. — 672 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1157-5 (рус.)

В настоящей книге приведены рекомендации, касающиеся того, как использовать методы рефакторинга для усовершенствования баз данных. Она посвящена описанию процедур проектирования базы данных с точки зрения архитектора объектно-ориентированного программного обеспечения, поэтому представляет интерес и для разработчиков прикладного кода, и для специалистов в области реляционных баз данных. В книгу включены многочисленные советы и рекомендации по улучшению качества проектирования базы данных. Значительное место уделено описанию того, как действовать в тех практических ситуациях, когда база данных уже существует, но плохо спроектирована, или когда реализация первоначального проекта базы данных не позволила получить качественную модель. Прежде всего книгу можно использовать в качестве технического руководства для разработчиков, непосредственно занятых на производстве. С другой стороны, она представляет собой теоретическую работу, стимулирующую дальнейшие исследования в направлении объединения объектно-ориентированного и реляционного подходов.

**ББК 32.973.26-018.2.75**

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства Addison-Wesley.

Authorized translation from the English language edition published by Addison-Wesley, Copyright © 2006 by Scott W. Ambler and Pramodkumar J. Sadalage.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Russian language edition is published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2007.

ISBN 978-8459-1157-5 (рус.)

ISBN 0-321-29353-3 (англ.)

© Издательский дом “Вильямс”, 2007

© Scott W. Ambler and Pramodkumar J. Sadalage, 2006

# Оглавление

Об авторах	17
Введение	24
Глава 1. Эволюционная разработка баз данных	31
Глава 2. Операции рефакторинга базы данных	45
Глава 3. Процесс рефакторинга базы данных	65
Глава 4. Развертывание на производстве	87
Глава 5. Операции рефакторинга базы данных	97
Глава 6. Операции рефакторинга структуры	109
Глава 7. Операции рефакторинга качества данных	183
Глава 8. Операции рефакторинга ссылочной целостности	229
Глава 9. Операции рефакторинга архитектуры	255
Глава 10. Операции рефакторинга методов	299
Глава 11. Преобразования	317
Приложение А. Обозначения языка моделирования данных UML	333
Приложение Б. Глоссарий	339
Приложение В. Литература	345
Приложение Г. Список операций рефакторинга и операций преобразования	347
Приложение Д. Отзывы	353
Предметный указатель	356

# Содержание

<b>Об авторах</b>	<b>17</b>
<b>Предисловия</b>	<b>18</b>
<b>Введение</b>	<b>24</b>
Необходимость в осуществлении эволюционной разработки баз данных	25
Суть адаптивных методов	27
Как читать эту книгу	28
<b>Благодарности</b>	<b>30</b>
<b>Глава 1. Эволюционная разработка баз данных</b>	<b>31</b>
1.1. Рефакторинг баз данных	33
1.2. Эволюционное моделирование баз данных	34
1.3. Регрессионное тестирование базы данных	37
1.4. Управление конфигурациями артефактов базы данных	41
1.5. Варианты среды, предназначенные для разработчиков	41
1.6. Недостатки эволюционных методик разработки баз данных, препятствующие их внедрению	43
1.7. Резюме	44
<b>Глава 2. Операции рефакторинга базы данных</b>	<b>45</b>
2.1. Рефакторинг кода	45
2.2. Рефакторинг баз данных	47
2.2.1. Среда базы данных с одним приложением	49
2.2.2. Среда базы данных с несколькими приложениями	51
2.2.3. Сохранение семантики	54
2.3. Категории операций рефакторинга базы данных	56
2.4. Признаки нарушений в работе базы данных	58
2.5. Перспективы дальнейшего распространения операций рефакторинга	60
2.6. Возможности упрощения операций рефакторинга схемы базы данных	61
2.7. Резюме	63
<b>Глава 3. Процесс рефакторинга базы данных</b>	<b>65</b>
3.1. Проверка приемлемости рассматриваемой операции рефакторинга базы данных	68
3.2. Выбор наиболее подходящей операции рефакторинга базы данных	69
3.3. Обозначение как устаревшей исходной схемы базы данных	70
3.4. Тестирование до выполнения, во время выполнения и после выполнения операции рефакторинга	74
3.4.1. Проверка схемы базы данных	74
3.4.2. Проверка результатов переноса данных	76
3.4.3. Тестирование внешних программ доступа	76

3.5. Модификация схемы базы данных	78
3.6. Перенос исходных данных	80
3.7. Рефакторинг внешних программ доступа	81
3.8. Выполнение регрессионных тестов	82
3.9. Применение в работе средств контроля версий	83
3.10. Объявление о проведении операции рефакторинга	83
3.11. Резюме	84
<b>Глава 4. Развертывание на производстве</b>	<b>87</b>
4.1. Эффективное развертывание с передачей из одной специализированной среды в другую	88
4.2. Применение наборов операций рефакторинга базы данных	90
4.3. Планирование подходящих интервалов развертывания	91
4.4. Развертывание всей системы	93
4.5. Удаление устаревшей схемы	95
4.6. Резюме	96
<b>Глава 5. Операции рефакторинга базы данных</b>	<b>97</b>
5.1. Преимущественное использование небольших изменений	98
5.2. Однозначная идентификация отдельных операций рефакторинга	98
5.3. Реализация крупных изменений в виде нескольких небольших изменений	100
5.4. Применение таблицы конфигурации базы данных	101
5.5. Преимущественное применение для синхронизации триггеров, а не представлений или пакетов	102
5.6. Применение достаточно продолжительного переходного периода	103
5.7. Упрощение стратегии группы контроля над внесением изменений в базу данных	103
5.8. Упрощение процедуры согласования с другими группами	104
5.9. Инкапсуляция средств доступа к базе данных	104
5.10. Возможность легко настраивать среду базы данных	105
5.11. Предотвращение дублирования кода SQL	105
5.12. Перевод информационных активов базы данных под контроль процедур управления изменениями	106
5.13. Учет необходимости перераспределения обязанностей в самой организации	106
5.14. Резюме	107
Сетевые ресурсы	107
<b>Глава 6. Операции рефакторинга структуры</b>	<b>109</b>
Проблемы, часто возникающие при реализации операций рефакторинга структуры	110
Операция рефакторинга “Удаление столбца”	112
Обоснование	112
Потенциальные преимущества и недостатки	112
Процедура обновления схемы	113
Процедура переноса данных	114
Процедура обновления программ доступа	114
Операция рефакторинга “Удаление таблицы”	117
Обоснование	117
Потенциальные преимущества и недостатки	117
Процедура обновления схемы	117

## 8 Содержание

Процедура переноса данных	118
Процедура обновления программы доступа	118
Операция рефакторинга “Удаление представления”	118
Обоснование	118
Потенциальные преимущества и недостатки	119
Процедура обновления схемы	119
Процедура переноса данных	120
Процедура обновления программ доступа	120
Операция рефакторинга “Введение вычисляемого столбца”	120
Обоснование	120
Потенциальные преимущества и недостатки	121
Процедура обновления схемы	121
Процедура переноса данных	123
Процедура обновления программы доступа	123
Операция рефакторинга “Введение суррогатного ключа”	123
Обоснование	125
Потенциальные преимущества и недостатки	125
Процедура обновления схемы	126
Процедура переноса данных	128
Процедура обновления программ доступа	128
Операция рефакторинга “Слияние столбцов”	130
Обоснование	130
Потенциальные преимущества и недостатки	131
Процедура обновления схемы	131
Процедура переноса данных	132
Процедура обновления программ доступа	132
Операция рефакторинга “Слияние таблиц”	133
Обоснование	133
Потенциальные преимущества и недостатки	135
Процедура обновления схемы	135
Процедура переноса данных	136
Процедура обновления программ доступа	137
Операция рефакторинга “Перемещение столбца”	139
Обоснование	139
Потенциальные преимущества и недостатки	140
Процедура обновления схемы	140
Процедура переноса данных	143
Процедура обновления программ доступа	144
Операция рефакторинга “Переименование столбца”	145
Обоснование	145
Потенциальные преимущества и недостатки	146
Процедура обновления схемы	146
Процедура переноса данных	147
Процедура обновления программ доступа	147
Операция рефакторинга “Переименование таблицы”	148
Обоснование	148
Потенциальные преимущества и недостатки	148
Процедура обновления схемы путем введения новой таблицы	149
Процедура обновления схемы с помощью обновляемого представления	150

Процедура переноса данных	151
Процедура обновления программ доступа	151
Операция рефакторинга “Переименование представления”	152
Обоснование	152
Потенциальные преимущества и недостатки	152
Процедура обновления схемы	152
Процедура переноса данных	153
Процедура обновления программ доступа	153
Операция рефакторинга “Замена данных типа LOB таблицей”	154
Обоснование	154
Потенциальные преимущества и недостатки	154
Процедура обновления схемы	156
Процедура переноса данных	158
Процедура обновления программ доступа	158
Операция рефакторинга “Замена столбца”	160
Обоснование	161
Потенциальные преимущества и недостатки	161
Процедура обновления схемы	161
Процедура переноса данных	162
Процедура обновления программ доступа	162
Операция рефакторинга “Замена связи “один ко многим” ассоциативной таблицей”	163
Обоснование	165
Потенциальные преимущества и недостатки	165
Процедура обновления схемы	165
Процедура переноса данных	167
Процедура обновления программ доступа	167
Операция рефакторинга “Замена суррогатного ключа естественным ключом”	168
Обоснование	168
Потенциальные преимущества и недостатки	169
Процедура обновления схемы	170
Процедура переноса данных	171
Процедура обновления программ доступа	171
Операция рефакторинга “Разбиение столбца”	172
Обоснование	173
Потенциальные преимущества и недостатки	174
Процедура обновления схемы	174
Процедура переноса данных	175
Процедура обновления программ доступа	175
Операция рефакторинга “Разбиение таблицы”	177
Обоснование	177
Потенциальные преимущества и недостатки	179
Процедура обновления схемы	179
Процедура переноса данных	180
Процедура обновления программ доступа	180
<b>Глава 7. Операции рефакторинга качества данных</b>	<b>183</b>
Проблемы, часто возникающие при осуществлении операций рефакторинга качества данных	184
Операция рефакторинга “Добавление поисковой таблицы”	185

Обоснование	185
Потенциальные преимущества и недостатки	186
Процедура обновления схемы	186
Процедура переноса данных	187
Процедура обновления программ доступа	187
Операция рефакторинга “Применение стандартных кодовых обозначений”	188
Обоснование	188
Потенциальные преимущества и недостатки	190
Процедура обновления схемы	190
Процедура переноса данных	191
Процедура обновления программ доступа	191
Операция рефакторинга “Применение стандартного типа”	192
Обоснование	192
Потенциальные преимущества и недостатки	194
Процедура обновления схемы	194
Процедура переноса данных	196
Процедура обновления программ доступа	196
Операция рефакторинга “Осуществление стратегии консолидированных ключей”	197
Обоснование	197
Потенциальные преимущества и недостатки	198
Процедура обновления схемы	198
Процедура переноса данных	200
Процедура обновления программ доступа	201
Операция рефакторинга “Уничтожение ограничения столбца”	201
Обоснование	202
Потенциальные преимущества и недостатки	202
Процедура обновления схемы	202
Процедура переноса данных	203
Процедура обновления программ доступа	203
Операция рефакторинга “Уничтожение значения, заданного по умолчанию”	203
Обоснование	203
Потенциальные преимущества и недостатки	204
Процедура обновления схемы	204
Процедура переноса данных	204
Процедура обновления программ доступа	204
Операция рефакторинга “Уничтожение столбца, не допускающего NULL-значений”	205
Обоснование	206
Потенциальные преимущества и недостатки	206
Процедура обновления схемы	206
Процедура переноса данных	206
Процедура обновления программ доступа	207
Операция рефакторинга “Введение ограничения столбца”	207
Обоснование	208
Потенциальные преимущества и недостатки	208
Процедура обновления схемы	209
Процедура переноса данных	209
Процедура обновления программ доступа	209
Операция рефакторинга “Введение общего формата”	210
Обоснование	211

Потенциальные преимущества и недостатки	211
Процедура обновления схемы	211
Процедура переноса данных	212
Процедура обновления программ доступа	212
Операция рефакторинга “Введение заданного по умолчанию значения”	213
Обоснование	214
Потенциальные преимущества и недостатки	214
Процедура обновления схемы	214
Процедура переноса данных	215
Процедура обновления программ доступа	215
Операция рефакторинга “Преобразование столбца в недопускающий NULL-значения”	216
Обоснование	216
Потенциальные преимущества и недостатки	216
Процедура обновления схемы	217
Процедура переноса данных	217
Процедура обновления программ доступа	218
Операция рефакторинга “Перемещение данных”	219
Обоснование	219
Потенциальные преимущества и недостатки	220
Процедура обновления схемы	221
Процедура переноса данных	221
Процедура обновления программ доступа	221
Операция рефакторинга “Замена кодового обозначения типа флажками свойств”	222
Обоснование	222
Потенциальные преимущества и недостатки	224
Процедура обновления схемы	224
Процедура переноса данных	226
Процедура обновления программ доступа	226
<b>Глава 8. Операции рефакторинга ссылочной целостности</b>	<b>229</b>
Операция рефакторинга “Добавление ограничения внешнего ключа”	229
Обоснование	230
Потенциальные преимущества и недостатки	230
Процедура обновления схемы	230
Процедура переноса данных	232
Процедура обновления программ доступа	233
Операция рефакторинга “Добавление триггера для вычисляемого столбца”	234
Обоснование	234
Потенциальные преимущества и недостатки	234
Процедура обновления схемы	236
Процедура переноса данных	237
Процедура обновления программ доступа	237
Операция рефакторинга “Уничтожение ограничения внешнего ключа”	238
Обоснование	238
Потенциальные преимущества и недостатки	238
Процедура обновления схемы	238
Процедура переноса данных	239
Процедура обновления программ доступа	239

Операция рефакторинга “Введение каскадного удаления”	239
Обоснование	240
Потенциальные преимущества и недостатки	240
Процедура обновления схемы	241
Процедура переноса данных	242
Процедура обновления программ доступа	242
Операция рефакторинга “Введение физического удаления”	243
Обоснование	243
Потенциальные преимущества и недостатки	244
Процедура обновления схемы	244
Процедура переноса данных	245
Процедура обновления программ доступа	245
Операция рефакторинга “Введение программного удаления”	246
Обоснование	246
Потенциальные преимущества и недостатки	247
Процедура обновления схемы	247
Процедура переноса данных	249
Процедура обновления программ доступа	249
Операция рефакторинга “Введение триггера для накопления исторических данных”	250
Обоснование	250
Потенциальные преимущества и недостатки	251
Процедура обновления схемы	251
Процедура переноса данных	253
Процедура обновления программ доступа	253
<b>Глава 9. Операции рефакторинга архитектуры</b>	<b>255</b>
Операция рефакторинга “Добавление методов CRUD”	255
Обоснование	256
Потенциальные преимущества и недостатки	257
Процедура обновления схемы	257
Процедура переноса данных	258
Процедура обновления программ доступа	259
Операция рефакторинга “Добавление зеркальной таблицы”	259
Обоснование	259
Потенциальные преимущества и недостатки	260
Процедура обновления схемы	261
Процедура переноса данных	261
Процедура обновления программ доступа	263
Операция рефакторинга “Добавление метода чтения”	263
Обоснование	264
Потенциальные преимущества и недостатки	264
Процедура обновления схемы	265
Процедура переноса данных	266
Процедура обновления программ доступа	266
Операция рефакторинга “Инкапсуляция таблицы в представление”	266
Обоснование	267
Потенциальные преимущества и недостатки	267
Процедура обновления схемы	268
Процедура переноса данных	268

Процедура обновления программ доступа	268
Операция рефакторинга “Введение вычислительного метода”	268
Обоснование	268
Потенциальные преимущества и недостатки	269
Процедура обновления схемы	269
Процедура переноса данных	270
Процедура обновления программ доступа	270
Операция рефакторинга “Введение индекса”	271
Обоснование	271
Потенциальные преимущества и недостатки	271
Процедура обновления схемы	272
Процедура переноса данных	272
Процедура обновления программ доступа	273
Операция рефакторинга “Введение таблицы только для чтения”	273
Обоснование	275
Потенциальные преимущества и недостатки	275
Процедура обновления схемы	276
Процедура переноса данных	277
Процедура обновления программ доступа	279
Операция рефакторинга “Перенос метода из базы данных”	280
Обоснование	280
Потенциальные преимущества и недостатки	282
Процедура обновления схемы	282
Процедура переноса данных	283
Процедура обновления программ доступа	283
Операция рефакторинга “Перенос метода в базу данных”	284
Обоснование	285
Потенциальные преимущества и недостатки	285
Процедура обновления схемы	285
Процедура переноса данных	286
Процедура обновления программ доступа	286
Операция рефакторинга “Замена метода (методов) представлением”	287
Обоснование	288
Потенциальные преимущества и недостатки	288
Процедура обновления схемы	288
Процедура переноса данных	289
Процедура обновления программ доступа	289
Операция рефакторинга “Замена представления методом (методами)”	290
Обоснование	290
Потенциальные преимущества и недостатки	290
Процедура обновления схемы	291
Процедура переноса данных	291
Процедура обновления программ доступа	291
Операция рефакторинга “Использование официально заданного источника данных”	292
Обоснование	292
Потенциальные преимущества и недостатки	294
Процедура обновления схемы	295
Процедура переноса данных	296
Процедура обновления программ доступа	296

<b>Глава 10. Операции рефакторинга методов</b>	<b>299</b>
10.1. Операции рефакторинга, которые приводят к изменению интерфейса	299
Операция рефакторинга “Добавление параметра”	300
Операция рефакторинга “Параметризация метода”	300
Операция рефакторинга “Удаление параметра”	301
Операция рефакторинга “Переименование метода”	302
Операция рефакторинга “Переупорядочение параметров”	302
Операция рефакторинга “Замена параметра явно заданными методами”	304
10.2. Операции рефакторинга внутренней организации методов	305
Операция рефакторинга “Консолидация условного выражения”	305
Операция рефакторинга “Декомпозиция условного выражения”	306
Операция рефакторинга “Извлечение метода”	307
Операция рефакторинга “Введение переменной”	310
Операция рефакторинга “Удаление флажка управления”	311
Операция рефакторинга “Удаление посредника”	311
Операция рефакторинга “Переименование параметра”	312
Операция рефакторинга “Замена подстановки литерала поиском в таблице”	312
Операция рефакторинга “Замена вложенного условного выражения защитными конструкциями”	314
Операция рефакторинга “Разбиение временной переменной”	314
Операция рефакторинга “Подстановка алгоритма”	315
<b>Глава 11. Преобразования</b>	<b>317</b>
Преобразование “Вставка данных”	317
Обоснование	317
Потенциальные преимущества и недостатки	318
Процедура обновления схемы	319
Процедура переноса данных	319
Процедура обновления программ доступа	319
Преобразование “Введение нового столбца”	321
Обоснование	322
Потенциальные преимущества и недостатки	322
Процедура обновления схемы	322
Процедура переноса данных	322
Процедура обновления программ доступа	323
Преобразование “Введение новой таблицы”	323
Обоснование	324
Потенциальные преимущества и недостатки	324
Процедура обновления схемы	324
Процедура переноса данных	325
Процедура обновления программ доступа	325
Преобразование “Введение представления”	325
Обоснование	325
Потенциальные преимущества и недостатки	327
Процедура обновления схемы	327
Процедура переноса данных	327
Процедура обновления программ доступа	328
Преобразование “Обновление данных”	329

	<b>Содержание</b>	<b>15</b>
Обоснование		330
Потенциальные преимущества и недостатки		330
Процедура обновления схемы		331
Процедура переноса данных		331
Процедура обновления программ доступа		331
<b>Приложение А. Обозначения языка моделирования данных UML</b>		<b>333</b>
<b>Приложение Б. Глоссарий</b>		<b>339</b>
<b>Приложение В. Литература</b>		<b>345</b>
<b>Приложение Г. Список операций рефакторинга и операций преобразования</b>		<b>347</b>
<b>Приложение Д. Отзывы</b>		<b>353</b>
<b>Предметный указатель</b>		<b>356</b>

*Посвящается Беверли, моей прекрасной молодой невесте.*

СКОТТ

*Посвящаю женщинам, которых я люблю больше всего, Рупали и  
нашей дочери Аруле.*

ПРАМОД

# Об авторах

**Скотт В. Эмблер** — консультант в области усовершенствования программных процессов (Software Process Improvement — SPI), проживающий недалеко от Торонто. Скотт создал методологии AM (Agile Modeling) ([www.agilemodeling.com](http://www.agilemodeling.com)), AD (Agile Data) ([www.agiledata.org](http://www.agiledata.org)), EUP (Enterprise Unified Process) ([www.enterpriseunified-process.com](http://www.enterpriseunified-process.com)) и AUP (Agile Unified Process) ([www.ambysoft.com/unifiedprocess](http://www.ambysoft.com/unifiedprocess)), а также возглавил работы по их усовершенствованию. Скотт — автор нескольких книг, в том числе *Agile Modeling* (John Wiley & Sons, 2002), *Agile Database Techniques* (John Wiley & Sons, 2003), *The Object Primer 3rd Edition* (Cambridge University Press, 2004), *The Enterprise Unified Process* (Prentice Hall, 2005) и *The Elements of UML 2.0 Style* (Cambridge University Press, 2005). Скотт работает в качестве редактора с правами публикации в коммерческом журнале *Software Development* ([www.sdmagazine.com](http://www.sdmagazine.com)), участвует и выступает с программными речами на различных международных конференциях, включая *Software Development*, *UML World*, *Object Expo*, *Java Expo* и *Application Development*. Скотт закончил Университет Торонто, получив степень магистра информатики. В свободное время Скотт изучает стили каратэ — годзюрю и кобудо.

**Прамодкумар Дж. Садаладже** — консультант компании ThoughtWorks, которая занимается разработкой и интеграцией приложений для предприятия. Прамод стал инициатором внедрения методов и процессов эволюционного проектирования и рефакторинга баз данных в 1999 году, работая над крупным приложением J2EE с использованием методологии экстремального программирования (Extreme Programming — XP). С тех пор Прамод, принимая участие во многих проектах, неизменно применяет эти методы и процессы. Прамод охотно делится своими знаниями в области администрирования баз данных в эволюционных проектах, применения эволюционных процессов к базам данных и влияния эволюционных подходов на администрирование базы данных, и в письменном, и в устном виде, чтобы дать возможность всем желающим освоить эволюционные методы проектирования баз данных. Свое свободное время Прамод проводит с женой и дочерью, совершая прогулки и бегая трусцой.

# Предисловия

Ниже в данном разделе приведены предисловия, написанные авторитетными специалистами.

- Десятилетие назад термин “рефакторинг”<sup>1</sup> был известен лишь ограниченному числу профессионалов, главным образом принадлежащих к сообществу пользователей языка Smalltalk. Но было очень интересно наблюдать за тем, как постепенно увеличивается количество желающих использовать рефакторинг в качестве формального и эффективного способа модификации работающего кода. В конечном итоге в наши дни многие рассматривают рефакторинг кода как важную составляющую разработки программного обеспечения.

Сферой моей деятельности является разработка приложений для предприятий, причем значительная часть такой разработки связана с обеспечением взаимодействия с базами данных. Впервые выпустив книгу по рефакторингу, я отметил базы данных (БД) как основную проблемную область при проведении рефакторинга, поскольку при попытке применить операции рефакторинга к БД приходится сталкиваться с новым кругом проблем. Острота этих проблем усугубляется тем заслуживающим сожаления барьером, который обнаруживается в сфере разработки программного обеспечения для предприятий, поскольку специалисты в области баз данных и разработчики программного обеспечения отделены друг от друга стеной взаимного непонимания и недовольства.

В работе Скотта и Прамода мне особенно понравилось то, что они стремятся каждый по-своему найти точки соприкосновения между этими категориями специалистов и объединить их усилия. Скотт неизменно пытается преодолеть указанный разрыв, подготавливая все новые и новые книги по базам данных, а написанное им по объектно-реляционному отображению оказало большое влияние на мои собственные сочинения, посвященные архитектуре приложений предприятия. Прамод, возможно, менее известен, но оказал на мои взгляды не меньшее влияние. Когда он начинал работу вместе со мной над одним проектом в компании ThoughtWorks, нам говорили, что проведение операций рефакторинга баз данных — невыполнимая задача. Прамод не прислушался к этим предостережениям, взял за основу некоторые наброски идей и превратил их в формализованную программу, позволяющую привести схему базы данных в постоянное, но управляемое движение. Благодаря этому разработчики приложений также получили возможность использовать методы эволюционного проектирования в своем коде. С тех пор Прамод опробовал эти методы на предприятиях многих наших клиентов, распространил их среди наших коллег из компании ThoughtWorks, поэтому, по край-

---

<sup>1</sup> Интересной аналогией, которая могла бы способствовать усвоению понятия “рефакторинг”, является то, что одно из толкований английского слова refactoring — перестановка мест слагаемых. — *Примеч. пер.*

ней мере для нас, базы данных навсегда исключены из списка объектов, не поддающихся развитию на основе непрерывного проектирования.

В этой книге сконцентрирован опыт двух человек, которые научились выживать на ничейной земле между приложениями и данными, а также приведены рекомендации, касающиеся того, как использовать методы рефакторинга для усовершенствования баз данных. Специалисты, знакомые с таким подходом, как рефакторинг, смогут отметить такую важную особенность описанных методов, что они позволяют обеспечить непрерывный перенос самих данных, а не только модификацию программ и структур данных. В этой книге показано, как достичь такой цели, и все, что в ней описано, подтверждается опытом проектирования (и положительным, и отрицательным), который был накоплен этими двумя специалистами.

Несмотря на то что я в восторге от появления этой книги, я также надеюсь, что это — только первый шаг. Меня очень радует то, что после выхода в свет моей книги по рефакторингу появилось много новых сложных инструментальных средств, позволяющих автоматизировать значительную часть операций рефакторинга. Я надеюсь, что то же самое произойдет применительно к базам данных, и мы увидим, как поставщики инструментальных средств предлагают инструменты, позволяющие упростить задачу непрерывного переноса схемы и данных для каждого. Но прежде чем это произойдет, вы сможете воспользоваться данной книгой, чтобы создать собственные процессы и инструменты для упрощения своей работы, а в дальнейшем эта книга будет оставаться для вас ценным источником знаний, позволяющим успешно использовать указанные инструментальные средства рефакторинга баз данных.

Мартин Фаулер, редактор серии; руководитель исследовательских работ компании ThoughtWorks

- С того времени, когда я начал свою карьеру в области разработки программного обеспечения, изменились до неузнаваемости многие аспекты производства и технологии. Но не изменилось одно — фундаментальный характер разработки программного обеспечения. Задача создания хотя бы какого-то программного обеспечения никогда не была особенно сложной — достаточно лишь получить в свое распоряжение компьютер и начать выдавать “на-гора” код. Но создать хорошее программное обеспечение всегда было трудно, и тем более было очень нелегко разработать превосходное программное обеспечение. Эта ситуация не изменилась и сегодня. В наши дни стало удобнее создавать все более крупные и сложные программные системы, собирая воедино фрагменты программ, полученных из различных источников, возможности инструментальных средств разработки программного обеспечения еще больше расширились, и мы гораздо лучше стали разбираться в том, какие подходы к осуществлению процесса разработки программного обеспечения принесут или не принесут результаты. Тем не менее основная часть программного обеспечения все еще продолжает оставаться ненадежной, а борьба за достижение допустимых уровней качества кажется нескончаемой. Возможно, причиной такого положения дел является то, что создаваемые системы становятся все крупнее и сложнее, или, возможно, то, что в используемых методиках все еще не преодолены какие-то фундаментальные недостатки. Я счи-

таю, что разработка качественного программного обеспечения и в наши дни остается столь же трудоемкой, как и прежде, из-за совместного проявления этих двух факторов. К счастью, время от времени появляются новые технологии и методики, внушающие надежду. Среди этих достижений редко встречаются новинки, которые позволяли бы резко повысить нашу способность реализовать тот потенциал, который внушал нам надежду при запуске большинства проектов. Методы, применяемые в рефакторинге, наряду со связанными с ними адаптивными методологиями, относятся именно к таким редкостным достижениям. Работа, результаты которой изложены в этой книге, расширяют возможности появления перспективных технологических достижений в очень важном направлении.

*Рефакторинг* — это контролируемый метод безопасного улучшения проекта кода без изменения его функциональной семантики. Возможность добиться усовершенствования кода дана любому разработчику, но подход, основанный на проведении рефакторинга, предусматривает применение формального метода безопасного внесения изменений (с помощью тестов) и распространение полученных при этом знаний, накопленных сообществом по разработке программного обеспечения (в виде формальных определений операций рефакторинга). С того времени, как была издана оригинальная книга Фаулера по этой теме, нашли широкое распространение операции рефакторинга и были приняты на вооружение инструментальные средства, позволяющие обнаружить потенциальные возможности проведения операций рефакторинга и применить эти операции к коду. Но применительно к тому уровню в многоуровневой структуре приложений, на котором используются данные, оказалось, что проведение операций рефакторинга является намного более затруднительным. Как показано в этой книге, связанные с этим проблемы отчасти обусловлены применением сложившихся ранее подходов, но не менее важно то, что еще не был четко определен процесс и набор операций рефакторинга, применимых к уровню данных. Этот фактор действительно оказывал неблагоприятное воздействие, поскольку низкое качество проекта на уровне данных почти всегда становится причиной возникновения проблем на более высоких уровнях, как правило, вызывая применение целого ряда неудачных проектных решений, принятых для осуществления бесполезных усилий стабилизировать шаткий фундамент. Более того, невозможность обеспечить развитие уровня данных, то ли из-за отрицания необходимости в этом, то ли из-за страха перед изменениями, препятствует реализации способностей всех остальных участников разработки создать наилучшее возможное программное обеспечение. Работа, проделанная авторами, столь важна потому, что позволяет решить именно эти проблемы; мы теперь имеем организационные и технологические возможности итеративного внесения усовершенствований в проекты, применяемые в этой жизненно важной области.

Меня очень обрадовало появление данной книги, и я надеюсь, что она станет стимулом к созданию инструментальных средств поддержки описанных в ней методик. В настоящее время наблюдается интересный этап развития индустрии программного обеспечения, связанный с постоянно растущей значимостью программного обеспечения с открытым исходным кодом и тех средств совместной разработки кода, которые с ним связаны. Такие проекты, как Eclipse Data Tools Platform, становятся естественным центром притяжения для всех тех разработчи-