

*How to Build a Successful  
Application Using Agile Without  
Sacrificing Data Management*

# Building the Agile Database

first edition



*How to Build a Successful  
Application Using Agile Without  
Sacrificing Data Management*

# Building the Agile Database

first edition

Larry Burns



Technics Publications

New Jersey

Published by:

**Technics Publications, LLC**

966 Woodmere Drive  
Westfield, NJ 07090 U.S.A.  
[www.technicspub.com](http://www.technicspub.com)



Edited by Carol Lehn

Cover design by Mark Brye

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

All trade and product names are trademarks, registered trademarks, or service marks of their respective companies, and are the property of their respective holders and should be treated as such.

This book is printed on acid-free paper.

Copyright © 2011 by Larry Burns

ISBN, print ed.

978-1-9355041-5-3

First Printing 2011

Library of Congress Control Number: 2011931258

**ATTENTION SCHOOLS AND BUSINESSES:** Technics Publications books are available at quantity discounts with bulk purchase for educational, business, or sales promotional use. For information, please email Steve Hoberman, President of Technics Publications, at [me@stevehoberman.com](mailto:me@stevehoberman.com).

## Contents at a Glance

---

<i>Acknowledgements</i>	<i>13</i>
<i>Foreword</i>	<i>15</i>
<i>Read me first!</i>	<i>17</i>
<i>Introduction</i>	<i>23</i>
<i>CHAPTER 1: Taking the Business View</i>	<i>29</i>
<i>CHAPTER 2: Agile Explained</i>	<i>43</i>
<i>CHAPTER 3: Agile Data Management</i>	<i>67</i>
<i>CHAPTER 4: Data Management Roles and Responsibilities</i>	<i>87</i>
<i>CHAPTER 5: Managing the Logical-Physical Divide</i>	<i>103</i>
<i>CHAPTER 6: Agile Data Design</i>	<i>119</i>
<i>CHAPTER 7: Agile Database Design</i>	<i>133</i>
<i>CHAPTER 8: Agile Modeling and Documentation</i>	<i>157</i>
<i>CHAPTER 9: Building the Agile Database</i>	<i>169</i>
<i>CHAPTER 10: Refactoring Made Easier</i>	<i>219</i>
<i>CHAPTER 11: Developing an Agile Attitude</i>	<i>239</i>
<i>CHAPTER 12: Sales Option Management Application</i>	<i>247</i>
<i>Afterword: Notes and Resources</i>	<i>263</i>
<i>Acronyms and Terms</i>	<i>265</i>
<i>Index</i>	<i>271</i>



## Contents

---

<b><i>Acknowledgements</i></b>	<b>13</b>
<b><i>Foreword</i></b>	<b>15</b>
<b><i>Read me first!</i></b>	<b>17</b>
Case Study: The Blue Moon Guitar Company	20
<b><i>Introduction</i></b>	<b>23</b>
<b><i>CHAPTER 1: Taking the Business View</i></b>	<b>29</b>
The Importance of Stakeholders	29
The Economics of Application Development	32
The Role of Business Data Management	35
<b><i>CHAPTER 2: Agile Explained</i></b>	<b>43</b>
The Importance of Agility	43
Agile Development Explained	46
Implications of an Agile Approach	47
Critical Issues in Agile Development	49
A Building or a Garden?	55
Agile vs. Software Engineering	60
<b><i>CHAPTER 3: Agile Data Management</i></b>	<b>67</b>
PRISM: The 5 Principles of Data Management	67
Agile Data Management	70
What is a Database?	76
In Defense of the “Intelligent” Database	81

<b>CHAPTER 4: Data Management Roles and Responsibilities</b>	<b>87</b>
The Art of Data Management	87
The Data Manager Role	89
The Data Analyst Role	90
The Database Developer Role	90
The Database Administrator Role	91
The Data Steward Role	91
The Data Development Cycle	91
Data Managers as Intermediaries	92
Database Development Tasks	93
The Evolution of Data Management	96
<b>CHAPTER 5: Managing the Logical-Physical Divide</b>	<b>103</b>
The Object-Relational Impedance Mismatch	104
Sources of Confusion	106
The Virtual Data Layer	108
The Virtual Data Layer in Action	112
<b>CHAPTER 6: Agile Data Design</b>	<b>119</b>
The Focus of Data Design	119
The Cost of Losing Focus	120
How Much Design is “Just Enough”?	123
The Purpose of Normalization in Logical Design	126
When to do Data Design	128
<b>CHAPTER 7: Agile Database Design</b>	<b>133</b>
The Focus of Database Design	133
The Physical Design Process	136
Database Design Deliverables	142



How Much Should You Implement? _____	145
The Purpose of Normalization in Physical Design _____	147
When Should You Denormalize? _____	151
When Shouldn't You Denormalize? _____	152
Normalization "After the Fact" _____	153
When to do Physical Design _____	153
Moving From the Logical to the Physical _____	154
<b>CHAPTER 8: Agile Modeling and Documentation _____</b>	<b>157</b>
What is an "Agile Model"? _____	157
Is the Logical Data Model an "Agile Model"? _____	160
Is the Physical Database Model an "Agile Model"? _____	164
Agile Documentation _____	165
<b>CHAPTER 9: Building the Agile Database _____</b>	<b>169</b>
Abstraction and Encapsulation _____	169
Cohesion and Coupling _____	171
Virtualizing the Database _____	171
Views _____	172
Stored Procedures _____	178
Fundamental Stored Procedures _____	185
Triggers _____	189
Functions _____	194
Work Tables and Materialized Views _____	201
Application Data Objects _____	204
Complex Datatypes _____	205
Standards for Database Coding _____	208
What Should Go Where? _____	211

Testing and Diagnostics _____	213
Optimizing Performance _____	214
<b>CHAPTER 10: Refactoring Made Easier _____</b>	<b>219</b>
Managing Change _____	220
Determining Where and How to Change _____	223
Refactoring the Logical Model _____	224
Refactoring the Physical Design _____	226
Refactoring the Database Schema _____	229
Refactoring the Virtual Data Layer _____	230
Refactoring Data _____	231
An Example of Multi-Level Refactoring _____	233
<b>CHAPTER 11: Developing an Agile Attitude _____</b>	<b>239</b>
The Three Agile Attitudes _____	239
Agile in the Organization _____	241
<b>CHAPTER 12: Sales Option Management Application _____</b>	<b>247</b>
Description of the Problem _____	247
Pre-Development Activities _____	249
Development Activities _____	253
Post-Development Activities _____	258
<b>Afterword _____</b>	<b>261</b>
<b>Notes and Resources _____</b>	<b>263</b>
<b>Acronyms and Terms _____</b>	<b>265</b>
<b>Index _____</b>	<b>271</b>

*To my loving Aunt, Sarah McPherson, who inspired  
in me a life-long love of books, learning and  
teaching, and to the memory of my parents, Keith G.  
(Bob) and Mary (Peg) Burns.*

*And, always, to Becky for her continual love and  
support.*



## Acknowledgements

---

I would first of all like to acknowledge and thank my mentors in the data management profession, in particular John Dart, Dave Wells, Tim Feetham, Bill Stewart, Roger Paulsen, and Jim Bigej. They gave me my start in this business, and guided me with their wisdom and experience along the way.

I would like to thank my colleagues at DAMA International, in particular Deborah Henderson of the DAMA Foundation and Mark Moseley of the Chicago chapter of DAMA, who gave me the opportunity to contribute to the Data Management Body of Knowledge (DAMA-DMBOK). Also, of course, my friends and colleagues in the Puget Sound chapter of DAMA, who have let me bend their ears and try their patience at numerous monthly meetings, gave me the first opportunity to explore some of the ideas in this book, and provided me with excellent feedback.

I would like to thank all of my colleagues, especially those in the Software Development Center and in the Data Services group. I have learned a lot from working with all of these wonderful people, and many of their thoughts, ideas, suggestions, and comments have found their way into this book.

In particular, I would like to thank Bill Sheboy of the SDC, and Bill Stewart of the Data Services group, for reviewing the manuscript of this book and providing many excellent suggestions. To the extent that I took their advice, this is now a much better book.

I would like to thank Steve Hoberman of Technics Publications for shepherding the manuscript through its various revisions, and helping it see the light of day. His advice was invaluable, and this project would not have seen completion without his guidance.

## **14 Building the Agile Database**

Last but not least, I would like to thank my long-suffering wife Becky who, at many points in the creation of this book, probably wished she had married a long-haul truck driver. Thanks, sweetheart!

I have often wondered how two of the hottest trends in information systems – Agile development and data governance – can coexist and reconcile their differences. In my consulting work, I typically see three scenarios: Agile development supersedes data resource management, data management limits agility, or two groups work at cross-purposes with a high degree of conflict.

*Building the Agile Database* is an unprecedented and much needed book that describes a very pragmatic approach to resolving tension that is common between application development and enterprise data management. From the developer perspective, it seems that data management creates barriers to fast and adaptive development. From the data management point-of-view, Agile development appears to ignore or discard many of the best practices of data asset management. The conflict is driven by different motivations – developers driven by a short-term need for functionality, and data managers motivated by a long-term need for high-quality business data.

This isn't a new problem; it has been with us for decades. The very same dilemma existed back when I was an application developer in the 1980's and 90's. Agile methods didn't exist, and we practiced something called Rapid Application Development (RAD). The premise with RAD was that we could quickly prototype functions when working with a stable data model. In other words, we could be fast once the hard work of database design was finished.

Today the approach is agile development, with development cycles much faster than those of RAD. Business cycles are faster and urgency greater, making agile development an imperative. At the same time, the stakes are raised for data management. Regulatory compliance and business analytics are among the factors that make data quality, continuity, and cohesion critical; and the variety of data and database models is much more extensive – relational, dimensional, columnar, unstructured, and so on. We

## 16 Building the Agile Database

have the same developer-database dilemma in a faster-moving and more complex environment.

Fortunately, Larry Burns answers some of the hard questions that must be tackled to resolve the conflict. Larry begins by making data administrators and database administrators “Agile aware” with understanding of the *what* and *why* of agile development. Next, he describes the principles of data management that are in apparent conflict with agile: Performance, Reusability, Integrity, Security, and Maintainability (PRISM). Now, with the stage set, Larry describes several concepts and techniques for design of agile databases. Many of the techniques may be familiar – abstraction, virtualization, and normalization, for example. New insight is in the application of the techniques and their implications for agility.

When I first met Larry Burns, he was an application developer at the University of Washington. Today he is a data management professional with more than a decade of experience. Larry is the right person to author a book of this kind. He has walked both sides of the street; he sees the conflict from both perspectives; and he offers pragmatic solutions born of experience as a practitioner in the field.

*Building the Agile Database* doesn’t have all of the answers, but it makes big steps in the right direction. Every database administrator, every application developer, and every information systems manager will learn from this book.

Thanks, Larry, for stepping up to this long-standing problem.

Dave Wells  
Information Management Consultant and Mentor



This book is meant for a broad audience that includes all of the application development stakeholders whose roles are explained in Chapter 1: application developers, database developers, database administrators (DBAs), data managers, data analysts, Quality Assurance (QA) managers, QA testers, project managers, business managers, enterprise architects, operations managers, and application support managers.

The early chapters of the book will focus on general, business-oriented principles that will help all stakeholders understand the roles and needs of the other groups, and explain how they can work together effectively, in the context of Agile development. The middle chapters explore the book's central theme of separating the work of data design (the logical, or business requirements view of data) from the work of database design (the physical, or implementation-specific view of data), and introduce the important concept of data virtualization. The final chapters show how these principles can be practically applied to Agile database development, and will be of interest primarily to application developers, database developers, and DBAs. The book concludes with a section explaining how to develop an "Agile Attitude" that helps people on Agile development teams work together more effectively. Everyone involved in the process of application software development will find something of value in this book!

The following outline briefly describes each chapter's content:

- **Chapter 1: Taking the Business View.** This chapter explains the importance of taking a business-focused, stakeholder-oriented view of application development, summarizes the economics of application development, and explains the importance of data – and of good data management practices – to the enterprise. This chapter should be read by all readers, as it provides much of the justification for the approach this book recommends.

## 18 Building the Agile Database

- **Chapter 2: Agile Explained.** Explains how Agile has transformed software development, examines the implications of an Agile approach, and explores some of the critical issues in Agile Development. Data management professionals will want to be sure to read this chapter, as it describes the justification for an Agile approach. Even Agile practitioners will want to read this chapter, since it explains the difficulty that people in some roles will have in adapting their processes to an Agile model.
- **Chapter 3: Agile Data Management.** In this chapter, an Agile approach to data management is described, based on the five PRISM principles. Different ways of looking at what a database is are examined, and a defense of the “intelligent” database is proposed. The information in this chapter is central to the book, and should be read by everybody.
- **Chapter 4: Data Management Roles and Responsibilities.** Describes the roles and responsibilities of data management professionals in application development, enterprise data management, and operational support; explains how these roles are evolving, and describes how their work is affected by an Agile approach. The information in this chapter is essential to understanding data management work, and should be read by everybody.
- **Chapter 5: Managing the Logical-Physical Divide.** Examines the root cause of what Agile practitioners call the “object-relational impedance mismatch”, the disconnect between the logical (requirements) and physical (implementation) views of data. The consequences of confusing these two views of data are explored, and the central concept of the Virtual Data Layer is introduced and explained. At this point, the focus of the book shifts to material that will be mostly of interest to application developers, data analysts, and DBAs, although I would recommend that project managers,

application architects, and data managers read and understand this chapter as well.

- **Chapter 6: Agile Data Design.** Explores the logical side of the data divide to determine how best to do requirements modeling in the context of an Agile project. Since Agile emphasizes a minimum of design work, how and when can data modeling be done? The role of normalization in logical data modeling is explained. The material in this chapter will be of great value to data analysts, and to anyone involved in understanding and modeling data requirements for an application or business process.
- **Chapter 7: Agile Database Design.** Explores the physical side of the data divide to determine how best to do database design and implementation in the highly-iterative context of an Agile project. The role of normalization in physical database design is explained. This material will be of value to DBAs, application database developers, and anyone involved in designing and creating data structures for applications.
- **Chapter 8: Agile Modeling and Documentation.** Examines the concept of an “Agile Model” to determine how much – and what kind – of documentation can and should be produced from the logical and physical data activities. Agile developers, data analysts, DBAs, Scrum Masters and project managers will find this chapter useful.
- **Chapter 9: Building the Agile Database.** Examines different approaches to data virtualization, both relational and object-oriented and describes how to implement data requirements in the database. This chapter will be of value primarily to DBAs and application developers.
- **Chapter 10: Refactoring Made Easier.** Explains the Agile concept of refactoring, and how refactoring is applied to the logical model, physical database design, database schema,

virtual data layer, and data. The emphasis is on minimizing costly refactorings at the database schema level. This chapter will be of value primarily to DBAs and application developers, but will also be of interest to QA (Quality Assurance) and Application Support people.

- **Chapter 11: Developing an Agile Attitude.** The biggest determinant for the success of an Agile project is the attitude of the people involved. Each person must make an effort to understand the roles of everyone involved, what they contribute to the effort, and what they need to be successful. Then, everyone must commit themselves to the success of the undertaking. The three Agile attitudes of commitment, cooperation, and communication are explained. This chapter should be read and understood by anyone involved in an Agile project – or in any other sort of application development effort.
- **Chapter 12: Case Study – A Sales Option Management Application.** This case study describes how Agile principles were applied in the development of an application that maintains sales option data for a manufacturing company. The case study is not intended to describe how Agile ought to be practiced, but to illustrate how data management can be done in the context of an Agile project, and how Agile projects can be done in the context of well-defined project management processes and stakeholder groups. This chapter is essential to the book, and should be read by everybody.

### ***Case Study: The Blue Moon Guitar Company***

---

To illustrate the application of the concepts in this book, I've created a fictional company called "Blue Moon Guitar Company". The company designs and builds custom guitars and other wooden stringed instruments to order; that is, it does not mass-manufacture instruments. When ordering an instrument, the customer can specify a number of customization options, such as size, finish, type of tuning pegs and bridge, pick guard design, size and location of the sound hole, type(s) of wood used, type and

placement of electronics (if any), and so on. Each model of instrument is associated with a set of these customization options, and each option specified contributes to the total list price of the instrument.

The Sales Option Management (SOM) application will enable the company to input and manage these options, and their relationship to the models of instruments offered by the company. The application will also enable Blue Moon's sales representatives to help customers design their dream instrument, and give them an accurate price quote when all of the options have been chosen.



In the decade or so that I've spent working in the field of data management (after many years as an application developer), I've experienced first-hand the challenges of implementing effective data management processes in the context of a fast-moving IT organization at a large global Fortune 500 company.

Most of the existing literature on application development, including much of what you'll read in the online developer forums, highlights the difficulty of getting application developers and data managers to work together effectively, especially in the context of Agile development. In many large organizations, there is often a considerable amount of contention between application developers, who need to get their applications written and deployed as quickly as possible, and data managers, enterprise architects, and other "big picture" folks, who are trying to focus on the long-term needs of the business. Although Agile development methodologies have been around for a decade, we seem to be no closer to resolving this contention. In fact, the rift between the two groups has become so wide at some companies, developers are advocating doing away with databases altogether!

In this book, I intend to demonstrate, in a practical way, how application developers and database developers can work together to satisfy both application (functional) requirements and larger-scale business data requirements. My complaint about much of the current Agile literature is that it is written solely from the point of view of Agile developers committed to the cause of Agile Development. There is no explanation, for non-Agile people, of why Agile is an important, even crucial, methodology (and not just the application methodology *du jour*). Nor is there any explanation, or exploration, of the roles and needs of other stakeholders in the application development effort.

Instead of exploring alternative approaches that would enable *all* stakeholders to do what they feel they need to do, most of the literature on Agile Development simply says, "The Agile approach

## 24 Building the Agile Database

is the correct one, and everyone else will have to accept it and change.” In this book, I hope to educate both application developers and data professionals about how to approach Agile projects in a way that meets the needs of the *entire* organization.

Everybody involved in an application development effort, including business users, project managers, QA testers, DBAs, and developers, is a stakeholder in the outcome, with legitimate interests and needs. To create an effective application development strategy, it is essential to understand what each of these groups brings to the table, and what each of them needs in order to do their jobs effectively. Each group has to learn as much as possible about what the other group is trying to do, and why, and then commit themselves to working together in a positive and professional manner, putting aside preconceived ideas and personal antagonisms in pursuit of the common enterprise goal.

In addition to taking a more inclusive approach to this topic, I also want to define an approach that is more business-focused. Both application developers and data managers need to understand that it’s not their money that is being spent and that, personal interests aside, the important goal is to satisfy both the short-term and long-term needs of the business. This requires a complete understanding of the economics of application development, the trade-offs involved in various development approaches, and a commitment to achieving the maximum ROI (Return On Investment) from each project.

I also believe that in any effective approach, the less prescription, the better. Instead of telling people, “This is what you must do, and this is how to do it!”, it is better to simply establish some fundamental goals and principles, and let people chart their own course. This is why, for example, I won’t be writing anything about the “rules” of normalization (normally *de rigueur* for a book of this type). Instead, I’ll talk about what normalization is, how it works (in both the logical and physical realms), and what the trade-offs are in the context of various types of data structures. The rest is up to you.



I'm also going to try to forsake ideology for the sake of practicality. Both developers and data managers tend to have very strong opinions about what is the “right” thing to do, which is why conflicts between the two groups tend to take on the tone of religious disputes. Developers tend to be focused on ease and speed of coding, while data managers search for the Holy Grail of “a single source of the truth”. Reality is usually much more complicated than this. As we'll see, getting an application out the door quickly may be the *least* cost-effective thing to do, and an organization's data needs may encompass several views of multiple truths, occurring in different business contexts.

However, my major source of dissatisfaction with most of the books I've read on the subject of application database development is that they don't properly address the critical distinction between the *logical* and *physical* views of data. The common assumption is that a (relational) database will be modeled in third-normal form, and implemented in third-normal form; in other words, there is no distinction made between the logical design (the data model) and the physical schema of the database. In this book, I will show why logical modeling and database design must be treated as two separate activities, each with its own purpose and goals, and its own set of rules, and how failure to do this leads to many common database problems, including what has been referred to as the “object-relational impedance mismatch”.

This disconnect becomes most apparent when examining the most common approach to Agile database development – continuous “refactoring” (reworking) of the database schema. Not only is this a difficult, sometimes dangerous, and often unnecessary thing to do from a database perspective, it's also, ironically, one of the *least* Agile approaches from an application development perspective! In Agile Development projects that I personally have worked on, the schema-refactoring approach has resulted in significant rework, project delays, and frayed tempers in both the application and database groups. To make matters worse, many Agile practitioners also recommend performing “normalization after the fact”, i.e., starting with denormalized data structures and normalizing

incrementally as needed. Unfortunately, this approach often fails when it is discovered that the data needed to populate the new normalized key attributes is missing or corrupted, due to a lack of data integrity constraints.

An alternative approach, which I intend to advocate in this book, is to do as much of the refactoring and denormalization as possible in one or more *virtual* data layers residing above the database schema. This approach accomplishes a number of very important things: from a data management perspective, it helps preserve the critical distinction between the logical (application-independent) and physical (application-specific) views of data, enabling a single database to support multiple applications and multiple uses of business data. From an application development perspective, it helps reduce the coupling that rigidly ties applications to an unyielding and over-normalized view of the data, and enables iterative changes to be made quickly, easily, and painlessly. This approach, which our organization has used in a number of Agile Development projects, has resulted in much faster development and implementation times, less “scrap and rework”, more data and code reuse, lower maintenance costs, and greater cooperation between the data and development groups.

Too often, in the past, developers have said to data managers, “You MUST DO this!”; while data managers have said to developers, “You CAN’T DO that!”. Nothing is really going to change until both developers and data managers understand that everybody NEEDS TO DO certain things for certain reasons. Any methodology that denigrates the valid business needs of people or groups who must support it is doomed from the start.

As a database developer in good standing (I helped write the chapters on Database Development and Database Operations Management for DAMA International’s Data Management Body of Knowledge [DMBOK] and frequently speak at DAMA meetings and conferences), I am committed to an approach to data management that helps companies maximize the business value of their data. As a long-time application developer, I am also committed to helping software developers create high-quality,

reusable code that maximizes the return on our company's IT investments. I do not believe that these goals are mutually exclusive; indeed, I don't believe that either can be achieved without the other. My hope is that this book will inspire a broader vision of what both developers and data people can accomplish if they work together.

Larry Burns  
Kent, WA



## CHAPTER 1

### Taking the Business View

---

Most books on application and database development focus on IT (Information Technology) activities: choosing the application architecture, designing and coding the application, testing, etc. What often gets overlooked is that all IT activities exist (or, at least, *should* exist) to add value to a company's business activities, in pursuit of that company's business goals. At each step in the application development process, we need to ask ourselves: are we doing the right thing, in the right way, to help our company achieve its goals? Are our activities adding value to processes that are streamlined and effective? Are the activities of critical stakeholders in these processes being helped or hindered? An approach that helps one set of stakeholders is ultimately valueless if it impedes the constructive work of others, and ultimately costs the company time, money, and opportunity.

#### ***The Importance of Stakeholders***

---

One of the distinguishing characteristics of my approach to data management is my insistence on a stakeholder view. Most books on software development (especially Agile development) are written totally from the developer's point of view – nothing is more important than getting software out the door on schedule. Most books on data management are written totally from the data management point of view – nothing is more important than “a single view of the truth”. But in the real world, it's not as simple as that. All organizations and organizational processes involve a large number of concerned stakeholders, whose interests must be considered in any endeavor.

For companies to be successful over the long term, they must consider the interests and well-being of their customers, employees, and suppliers, as well as their stockholders and directors. David R. Vincent, whose seminal work, *The Information-Based Corporation*, is an excellent introduction to the concept of

stakeholder economics, goes so far as to include communities, competitors, and regulators in his list of stakeholders.<sup>1</sup> Failure to consider the environmental impact of corporate activities on surrounding communities, for example, has cost many companies millions of dollars in lawsuits and settlements. Similarly, many companies have lost millions of dollars fighting – and losing – anti-trust and patent infringement suits.<sup>2</sup> Other companies have created value for their shareholders by entering into cooperative ventures with their competitors; a consortium of insurance companies in Florida, for example, have pooled customer data in order to expedite the processing of third-party and multi-party insurance claims, resulting in benefits for both the companies and their customers.

In the arena of software development, there are many stakeholders at various levels who have a legitimate interest in the outcome of the development effort. These include:

- *Software developers*, whose livelihoods depend on being able to create useful software applications quickly and cost-effectively.
- *Project managers*, whose jobs depend on being able to complete projects on time and under budget.
- *Business users*, who want to be able to do their jobs as effectively as possible.
- *Product owners*, who are responsible for making sure the application meets the needs of the business.

---

<sup>1</sup> Vincent, David R. *The Information-Based Corporation: Stakeholder Economics and the Technology Investment* (Dow Jones-Irwin, 1990).

<sup>2</sup> Microsoft, for example, paid no stockholder dividends in 2002, and its stock lost 16% of its value, because much of their available cash was tied up in lawsuits and settlements (Source: *The Seattle Times*, November 6, 2002, p. E1).

- *Business analysts*, who understand the needs of the business and can communicate business requirements to the development team in technical terms.
- *Business managers*, who want to obtain the most value for their IT investment.
- *Enterprise architects*, who want to ensure the maximum reuse of enterprise assets at the lowest possible cost.
- *Data managers*, a subset of enterprise architects, who want to ensure the maximum reuse of enterprise data assets at the lowest possible cost.
- *Data analysts*, who are responsible for determining and documenting (modeling) business data requirements.
- *Database developers*, who must design and build databases (and other data structures) that satisfy both business and application requirements.
- *Database administrators*, who must maintain the databases (and the database software they run on), ensure that data can be accessed quickly and easily, and safeguard data against loss, theft, and harm.
- *Infrastructure administrators*, including server and network administrators, who oversee the day-to-day maintenance, testing, installation, and operation of the enterprise's IT infrastructure.
- *Quality Assurance (QA) managers*, who want to make sure that business users get the maximum benefit from new software with the least amount of pain and disruption to the business.
- *QA Testers*, whose job it is to make sure the software satisfies the user requirements and provides a positive user experience.
- *Operations managers*, who want to make sure that the day-to-day operations of the business run smoothly, with no (or minimal) disruptions. They also want to make sure that

## 32 Building the Agile Database

business operations can be quickly and effectively restored if a disruption occurs.

- *Application Support managers*, who are responsible for both the day-to-day support of IT applications, and for the maintenance and enhancement that enables these applications to maintain their value for the organization over time.
- *Portfolio managers*, who oversee and administer a set of applications supporting a specific area of the business.
- *Resource managers*, who are responsible for allocating the resources (people) needed for the project to succeed. Most of these resources will be divided between several different projects.

Each of these roles contributes some measure of value to the application development effort, and helps ensure, to some extent, the ultimate success of the project and its value to the business. It's important to understand that, while ignoring the needs of one set of stakeholders may provide some short-term advantage to another, the greatest long-term advantage to the enterprise is served by creating processes that enable all stakeholders to accomplish what they need to accomplish for the good of the organization. Taking a stakeholder view gives all affected groups a stake in the successful outcome of a given endeavor. Failure to do this encourages one or more groups to back-pedal, to slow down or obstruct the effort of others. I would argue that most of the internal conflicts that cause IT projects to come in late and over budget can be traced back to a failure to sufficiently consider the interests – and enlist the support – of all the groups whose active participation and cooperation is essential for success.

### ***The Economics of Application Development***

I also believe that application development and data management issues should be looked at from a broader economic perspective. This means taking a hard look at the economics of both application development and data management to determine what processes



and practices will deliver the maximum business value (ROI) for our IT investment. While it may be true, as Scott Ambler says, that “working software is the primary measure of progress”<sup>3</sup> in an application development effort, it is not the sole measure of value. It may also be true, from an application development standpoint, that “the highest priority is to satisfy the customer through early and continuous delivery of valuable software”;<sup>4</sup> but developers, and even end users, are not the only or even the principal stakeholders in the development effort. After all, it’s not their money that’s being spent! The ultimate goal of any application development effort is not just to put functionality into the hands of end users, but to increase the value and profitability of the company’s activities. So, we need to look at the whole “value equation” of application development, which includes:

- The initial cost of development.
- The incremental cost of subsequent changes and enhancements.
- The cost of maintenance over the life of the application.
- The cost of missed opportunities if the application is not deployed in a timely fashion, OR if it is deployed but fails to work as required, OR if the functionality of the application doesn’t meet current or future business needs, OR if the time and money spent developing the application could have been better spent on something else.
- The value returned to the company by the application.
- The value returned to the company from subsequent reuses of the application, application components, data structures, and/or application data.

---

<sup>3</sup> Ambler, Scott W. *Agile Database Techniques* (Wiley Publishing, Inc., 2003), p. 9.

<sup>4</sup> Ambler, op. cit.

## 34 Building the Agile Database

- The value returned to the company from subsequent reuses of any design artifacts, such as an architecture model, logical data model, or OOA class model.
- The value returned to the company from tools, skills, processes, methodologies, or knowledge created or acquired in the course of the development effort.

This is not a comprehensive list, but you get the idea. The basic objective of Agile Development, to get working applications into the hands of business users quickly enough to take advantage of business opportunities, is a laudable and correct one, but care must be taken to ensure that this is done in a way that doesn't sacrifice the company's long-term best interests on the altar of short-term goals. For example, we will see later on that there is a difference between "time to market" (the standard measure of software project success) and "time to money": the amount of time it takes for new – and often bug-ridden – software to start having a positive impact on a company's balance sheet.

What I'm advocating, of course, is that companies take a somewhat broader view of the application development equation than is usually found in Agile methodologies. I'm going to try not to be too prescriptive here, but would suggest that a more economically viable approach to application development might include the following:

- Establishing a consistent architectural framework of approved platforms, vendors, tools, processes, products, and languages that can support current and future development efforts across the enterprise. It is unnecessary (and economically counter-productive) to adopt a different set of architectural standards for every application or business unit.
- Creating a project management process to ensure steady progress toward the project's goals, anticipate problems, keep the project aligned with business objectives, ensure an adequate return on investment (ROI), and make sure that

lessons learned from one project are used to improve the process for future projects.

- Creating a quality assurance process and QA team to make sure that product quality requirements are met, that all functional requirements have been satisfied, that user expectations have been met, and that total cost of ownership (TCO) is minimized.
- Formalizing an implementation/turnover process to ensure that both end users and IT support and maintenance staff have adequate training and documentation, and will be able to use and support the application with a minimum of cost and effort.

This might, at first glance, appear to be smothering the development methodology with a lot of process, and it's true that taking this broader approach to application development involves more project stakeholders and more communication (including, probably, more formal documentation). But it needs to be recognized that these activities involve legitimate company stakeholders, whose interests and activities directly affect the company's profitability. There is no point to a software development methodology that leaves critical stakeholders dissatisfied, and ultimately costs the company money.

### ***The Role of Business Data Management***

Much has been written about the so-called "Information Economy", but most people do not understand the economic changes that are driving the new economy. At the heart of these changes are two driving forces: *commoditization* and *globalization*. Commoditization means that everything is basically a commodity; something to be produced as cheaply as possible, used, and then thrown away. Traditionally, manufacturers of goods make profits by lowering costs, creating more efficient manufacturing processes, improving quality, or producing newer or better products. They've spent most of the last three decades reengineering their businesses to create more efficient processes (laying off millions of employees along the way) and moving their manufacturing plants overseas to take advantage

of cheaper labor and production costs. Today, most manufactured products are virtually indistinguishable from one another; any given product, regardless of brand, is made in the same way, probably in the same overseas factory, in the cheapest and most cost-effective way possible. Thus, it has become very difficult, if not impossible, for businesses to increase their profits in the traditional manner.

How then, can companies increase value for their customers (and profits for their shareholders)? David R. Vincent makes the point that in the new global economy, business value is created by establishing and nurturing *relationships* with customers, suppliers, and dealers. He makes the further point that the essence of effective relationships lies in *empowerment*; in giving people the ability to do more things for themselves.<sup>5</sup> For example, from the comfort and convenience of my office, I can do all my banking, manage my investments, schedule my travel arrangements, and buy virtually anything anywhere in the world. Thanks to the power of the Internet and the various software applications it supports, we are moving from a service economy to what might be called a *self-service* economy.

To quote columnist Ian Shoales, we live in a data-driven world.<sup>6</sup> Businesses use CRM (Customer Relationship Management) software to identify problem areas and potential new business opportunities in their relationships with their customers; they use ERP (Enterprise Resource Planning) software to supply up-to-date financial information to workers at all levels of the business; they use data mining technology to identify patterns, trends, and possible opportunities for growth; they use intranets to improve their relationships with their employees; and they use extranets to improve their relationships with their suppliers and vendors. All of

---

<sup>5</sup> Vincent, op. cit.

<sup>6</sup> Ian Shoales' column "Data-Driven World" appeared monthly in *Intelligent Enterprise*, a trade magazine for database programmers, from 1997 to 2005. Ian Shoales is the alter ego of writer/performer Merle Kessler.

these different technologies have one thing in common: *they use the power of information to empower people to create value at the lowest possible level.*

Bill Gates, in his book *Business @ the Speed of Thought*,<sup>7</sup> echoes this idea of improving business processes by using information to empower individuals, and cites several examples from his struggling but moderately successful software company. Both Michael Dertouzos<sup>8</sup> of MIT and Thomas Davenport<sup>9</sup> of Harvard use the phrase “information marketplace” to describe a scenario in which economic value is generated through the exchange of information among individuals, much as the traditional marketplace generates value through the exchange of goods and services. They cite examples such as American Airline’s SABRE reservations system, which enables travel agents to locate and book flights on any airline. This enabled American to develop a preferred (and highly profitable) relationship with travel agents. Now, of course, online providers such as Expedia and Travelocity have given individual travelers the ability to make their own travel, hotel, and rental car reservations, bypassing the use of travel agents entirely. Vincent gives the example of American Hospital Supply (AHS), which made its order entry system available to its customers, allowing hospitals serviced by AHS to order all their supplies through the AHS system. This not only improved AHS’s relationship with its customers, but also gave it an advantageous bargaining position with its suppliers!

---

<sup>7</sup> William H. Gates III. *Business @ the Speed of Thought: Using a Digital Nervous System* (New York, NY: Warner Books, 1999).

<sup>8</sup> Michael Dertouzos. *What Will Be: How the New World of Information Will Change Our Lives* (San Francisco: HarperCollins, HarperEdge, 1997).

<sup>9</sup> Thomas Davenport and Laurence Prusak. *Working Knowledge: How Organizations Manage What They Know* (Boston: Harvard Business School Press, 1998).

To put this another way, we are moving from an economy based on physical value chains (e.g., from supplier to wholesaler to dealer to customer) to one based on what Jeffrey Rayport and John J. Sviokla have called the *virtual value chain*.<sup>10</sup> In the virtual *marketspace* (as distinguished from a physical marketplace), competitive advantage goes to those who can find innovative ways to use information to deliver goods and services. Sporting goods retailer Cabela's, for example, automatically emails discount offers to its customers on goods that are being closed out of inventory. I'm constantly getting emails from Amazon.com: "Dear Mr. Burns: People who have ordered *Fundamentals of Data Analytics* have also purchased Vladimir Nabokov's *Lolita*." (Not true). (As far as I know).

In contrast to physical value chains, which are discrete and linear, with defined points of input and output, virtual value chains are non-linear and virtually limitless – "a matrix of potential inputs and outputs that can be accessed and distributed through a wide variety of channels".<sup>11</sup> As Rayport and Sviokla point out, a customer not interested in a new compact disk by the Rolling Stones may still choose to download it from iTunes, listen to it on Pandora, or sit in on a chat session with them in the Internet's Voodoo Lounge.<sup>12</sup>

What fuels the information processes that make up the virtual value chain is *data*. Data assets, as Vincent points out, are a special type of circulating (as opposed to fixed) asset. But they are very different from other types of circulating assets such as cash and inventory. They do not disappear when consumed. They can be reused almost indefinitely (unless they become out-of-date), and in an almost infinite variety of ways. And, most importantly, because

---

<sup>10</sup> Rayport, Jeffrey F. and John J. Sviokla. *Exploiting the Virtual Value Chain*. Harvard Business Review, November-December 1995, pp. 75-85.

<sup>11</sup> Ibid, p. 83.

<sup>12</sup> Ibid, p. 77.

value can be created using data assets at minimal cost, the traditional barriers to entry and expansion for new or small businesses are eliminated, allowing almost anyone to become an entrepreneur.

Even for traditional businesses like the global Fortune 500 manufacturer I work for, the power of data assets can be harnessed to reduce costs, improve quality, expand markets, and fuel innovation. For these companies, information can contribute to business value in the following ways.<sup>13</sup>

- Eliminating repetitive or redundant tasks (automation)
- Enabling people to do more work, or to work more effectively (e.g., collaboration and workflow software that lets people work in distributed ad-hoc teams without frequent meetings)
- Designing improved business processes (e.g., Ford was able to shorten its design-to-market time by over a year, enabling it to take significant market share from its chief rival, General Motors. Also, by automating their design-to-manufacture process, Ford's management was able to accept bigger design risks, enabling Ford to break from its conservative tradition and produce car designs that had more customer appeal)
- Enabling business to be conducted globally

---

<sup>13</sup> The examples cited in this section, along with many others, can be found in the article *How Information Gives You Competitive Advantage*, by Michael E. Porter and Victor E. Millar (Harvard Business Review, July-August 1985, pp. 149-160), the article *Exploiting the Virtual Value Chain* by Jeffrey Rayport and John Sviokla (Harvard Business Review, November-December 1995, pp. 75-85), and David R. Vincent's book *The Information-Based Corporation: Stakeholder Economics and the Technology Investment* (Dow Jones-Irwin, 1990).



## 40 Building the Agile Database

- Reducing intermediaries (i.e., flattening hierarchies and giving more responsibility to empowered individuals and entrepreneurial teams)
- Adding intelligence to their products (e.g., PACCAR Inc can produce trucks that can communicate their location, fuel consumption, and status to the driver and the driver's dispatcher; if necessary, the driver will be directed to the nearest service dealership, where an appointment for service will have been automatically made)
- Lowering costs (Canon, for example, has built a low-cost manufacturing process around an automated parts selection and materials handling system, delivering parts "just in time" to workers on the assembly line)
- Enhancing differentiation (American Express has developed differentiated travel services for its corporate customers, using information systems to search for the lowest airline fares, hotels, and rental cars, and tracking travel expenses for each cardholder).

The point I'm trying to make here is that, in the Information Economy, competitive advantage is dictated by the quality and reusability of a company's data assets. To maximize its usefulness, data must be managed in a way that ensures its accuracy, timeliness, and business relevance. The data must be easily accessible, consumable, and transformable in a variety of ways. Imagine if, for example, the data in Cabela's inventory system was inaccessible to distributed applications, or if it couldn't easily be consumed and transformed by an email application, or if its customers were being offered discounts on products that didn't exist or weren't available (or that shouldn't be discounted).

It is in this context, the importance of managing high-quality, reusable, and easily consumable data assets to fuel innovative business information processes, that the rest of the material in this book should be considered.



### **Key Points**

- The end purpose of all IT activities (including application development and data management) is to add value to a company's business activities, in pursuit of that company's business goals.
  - All business activities should be conducted with the broadest possible view of the stakeholders involved. Enlisting the support of all affected stakeholders is essential to the success of any endeavor.
  - All business activities have an economic "value equation" that determines whether the activity will be profitable for the company. In any undertaking, the entire value equation must be considered.
  - In the new global economy, companies create value by using the power of information to nurture relationships, empower stakeholders, improve products and processes, and create new markets.
  - Companies rely on high-quality, reusable, and easily consumable data assets to quickly create or modify the information processes (the "virtual value chain") that enable them to be competitive in the global marketplace.
  - Proper management of business data assets is key to a company's success in the Information Economy.
-