Architecture & Patterns for IT

Service Management, Resource Planning, and Governance
Architecture & Patterns for IT
Service Management, Resource Planning, and Governance

Making Shoes for the Cobbler’s Children
Second Edition

Charles T. Betz
To Sue and Keane.
# Table of Contents

<table>
<thead>
<tr>
<th>List of Figures</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>xiii</td>
</tr>
<tr>
<td>Foreword I (Rob Stroud)</td>
<td>xv</td>
</tr>
<tr>
<td>Foreword II (Steve Bell)</td>
<td>xix</td>
</tr>
<tr>
<td>Preface</td>
<td>xxi</td>
</tr>
<tr>
<td>Author Biography</td>
<td>xxxiii</td>
</tr>
</tbody>
</table>

## CHAPTER 1 IT in a World of Continuous Improvement
- What Is “Information Technology”? ................................................. 2
- What Is an IT Service? ................................................................... 3
- What Is Lean? ............................................................................... 4
- What Is IT Value? .......................................................................... 9
- What Is Lean IT? ........................................................................... 13
- Conclusion .................................................................................. 30
- Further Reading .......................................................................... 31

## CHAPTER 2 Architecture Approach
- The Production of IT Services ................................................... 35
- IT Value Chains, Streams, and Processes .................................... 42
- The IT Management Functions .................................................... 70
- The Information Architecture of IT Management ........................ 90
- A Supporting Systems Architecture for IT Management ............... 121
- The Matrices .............................................................................. 138
- Conclusion .................................................................................. 148

## CHAPTER 3 Patterns for the IT Processes
- IT Process Principles .................................................................. 153
- Accept Demand Patterns ............................................................. 155
- Execute Project Patterns ............................................................ 167
- Deliver Release Patterns ......................................................... 171
- Complete Change Patterns ....................................................... 174
- Fulfill Service Request Patterns ............................................. 187
- Deliver Transactional Service Patterns .................................... 194
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore Service Patterns</td>
<td>231</td>
</tr>
<tr>
<td>Improve Service Patterns</td>
<td>233</td>
</tr>
<tr>
<td>Retire Service Patterns</td>
<td>240</td>
</tr>
<tr>
<td>Conclusion</td>
<td>241</td>
</tr>
<tr>
<td><strong>CHAPTER 4</strong> Patterns for the IT Lifecycles</td>
<td>243</td>
</tr>
<tr>
<td>The Application Service Lifecycle</td>
<td>244</td>
</tr>
<tr>
<td>The IT Infrastructure Service Lifecycle</td>
<td>278</td>
</tr>
<tr>
<td>The Technology Product Lifecycle</td>
<td>285</td>
</tr>
<tr>
<td>The IT Asset Lifecycle</td>
<td>293</td>
</tr>
<tr>
<td>General Patterns for IT Portfolio Management</td>
<td>299</td>
</tr>
<tr>
<td>Epilogue</td>
<td>311</td>
</tr>
<tr>
<td><strong>APPENDIX A</strong> Extended Definitions for the IT Architectural Catalogs</td>
<td>315</td>
</tr>
<tr>
<td>IT Lifecycle Definitions</td>
<td>315</td>
</tr>
<tr>
<td>IT Process Definitions</td>
<td>320</td>
</tr>
<tr>
<td>IT Function Definitions</td>
<td>331</td>
</tr>
<tr>
<td>IT Data Definitions</td>
<td>352</td>
</tr>
<tr>
<td>IT Management Systems Definitions</td>
<td>375</td>
</tr>
<tr>
<td><strong>APPENDIX B</strong> Fundamentals of Computing for the Business Professional</td>
<td>409</td>
</tr>
<tr>
<td><strong>APPENDIX C</strong> Production and Services</td>
<td>411</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>417</td>
</tr>
<tr>
<td>INDEX</td>
<td>425</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1.1 Systems thinking (from Weinberg) 7
Figure 1.2 The two axes of product value 12
Figure 1.3 The two axes of IT value 12
Figure 1.4 The dynamic tension of IT service 14
Figure 1.5 “TPS House” for Lean IT 26
Figure 2.1 Architectural primitives (the catalogs) 34
Figure 2.2 Computer 35
Figure 2.3 Inside the computer 35
Figure 2.4 Servers 35
Figure 2.5 End-to-end computing 36
Figure 2.6 Transactional value across the stack 37
Figure 2.7 Service lifecycle and transactional value 37
Figure 2.8 Service Lifecycle and Delivery 38
Figure 2.9 IT Service Derived 38
Figure 2.10 IT Service (UML representation) 38
Figure 2.11 Basic architectural elements supporting a business 39
Figure 2.12 Architecture by lines of business 39
Figure 2.13 Enterprise support capabilities 39
Figure 2.14 The IT service organization produces IT services! 40
Figure 2.15 A factory that makes factories? 40
Figure 2.16 The IT value stream produces IT services for itself 41
Figure 2.17 Chevrons 44
Figure 2.18 Relative scale of value chains, streams, processes 45
Figure 2.19 IT as a business: system context 48
Figure 2.20 “Inspire to retire” IT value chain 48
Figure 2.21 Decomposed IT value chain 54
Figure 2.22 The lifecycles are not synchronized 54
Figure 2.23 Application and infrastructure services 55
Figure 2.24 Lean work/wait 58
Figure 2.25 “Four-O” model 59
Figure 2.26 Four-O model to scale 60
Figure 3.3 Demand as precursor 159
Figure 3.4 Various demand routings 160
Figure 3.5 Restore-Resolve-Release 160
Figure 3.6 Restore-Resolve-Demand-Release 160
Figure 3.7 Demand-Restore-Resolve-Release 161
Figure 3.8 Demand Management system integration 164
Figure 3.9 Project management system integrations 169
Figure 3.10 Release management integration 172
Figure 3.11 ITIL® representation of Change/Project/Release 175
Figure 3.12 Alternate representation of Project/Release/Change 176
Figure 3.13 Change Justification 178
Figure 3.14 CI-based risk management 180
Figure 3.15 Configuration and metadata risk management 181
Figure 3.16 Metadata-based Risk management detail 182
Figure 3.17 Change impact (simple) 183
Figure 3.18 Change impact (complex) 183
Figure 3.19 Drift, Incident, and Change 185
Figure 3.20 Change Management System Context 186
Figure 3.21 Service Semantics 188
Figure 3.22 Service chain 189
Figure 3.23 Integrated Service Request Management 192
Figure 3.24 Core transactional systems in context 195
Figure 3.25 Service management system domain 196
Figure 3.26 Core configuration management 199
Figure 3.27 Appropriate data capture level 207
Figure 3.28 Business case for inventory (CMS) consolidation 209
Figure 3.29 Inefficient dependency entry 210
Figure 3.30 Efficient dependency entry 211
Figure 3.31 Configuration iteration 1 213
Figure 3.32 Configuration iteration 2 213
Figure 3.33 Configuration iteration 3 214
Figure 3.34 Configuration iteration 4 214
Figure 3.35 Knowledge management 217
Figure 3.36 Security and configuration management 224
Figure 3.37 Configuration Audit role 226
Figure 3.38 Configuration Audit process 227
Figure 3.39 Configuration Audit and discovery 228
Figure 3.40 IT financial management system context 229
Figure 3.41 Integrated Incident Management 232
Figure 3.42 Capacity Management system context 234
Figure 3.43 Risk Management system context 236
Figure 3.44 Continuous improvement integration 238
Figure 3.45 Service retirement integration 240
Figure 4.1 Value stream key 243
Figure 4.2 The Application Service Lifecycle 245
Figure 4.3 Application Alias and ID 252
Figure 4.4 Reconciliation model 254
Figure 4.5 Application service semantic context 257
Figure 4.6 Project, release, and application 262
Figure 4.7 Project/application direct relationship 262
Figure 4.8 Effort tracking based on portfolio entries 263
Figure 4.9 Metadata Management System Context 267
Figure 4.10 Enterprise architecture portfolio representation 268
Figure 4.11 Simple application association 271
Figure 4.12 Interface system 271
Figure 4.13 System interactions carrying Application and/or Infrastructure Service ID 272
Figure 4.14 IT enablement capability scope and relationships 276
Figure 4.15 IT enablement is to IT as IT is to business 277
Figure 4.16 The Infrastructure Service Lifecycle 279
Figure 4.17 Value chain – application and infrastructure tracks 280
Figure 4.18 The Technology Product Lifecycle 286
Figure 4.19 Technology data model 289
Figure 4.20 Standard technology stack 290
Figure 4.21 Vendor and product master data flows 291
Figure 4.22 The IT Asset Lifecycle 293
Figure 4.23 IT Asset systems context and identifier flows 299
Figure 4.24 Classification taxonomy 307
Figure 4.25 Sample application taxonomy and a key question 309
Figure A.1 Configuration Item taxonomy 358
Figure A.2 Application and boundaries 359
Figure A.3 Configuration Item dependencies 359
Figure A.4 Subtypes of Datastore 360
Figure A.5 Simple data dictionary 361
Figure A.6 Offering versus Service 369
Figure A.7 Service layering: types and Instances 371
Figure A.8 Components and interfaces 372
Figure A.9 Service as API 372
Figure A.10 Layered Application service 373
Figure C.1 Two dimensions of production 413
Figure C.2 Two dimensions of IT production 415
List of Tables

Table 1-1  IT Enablement Themes and Strategies  22
Table 2-1  Supporting Processes by Value Chain  50
Table 2-2  Major Conceptual Entities in the Business of IT  52
Table 2-3  IT Lifecycle Summary Definitions  57
Table 2-4  Value-Add and Non-Value-Add Statuses of IT Value Streams  60
Table 2-5  IT Process Definitions  68
Table 2-6  Lifecycles Mapped to Functional Homes  78
Table 2-7  IT Function Definitions  89
Table 2-8  Roles and Entities  108
Table 2-9  IT Management Data Definitions  119
Table 2-10  IT Management Systems Definitions  136
Table 2-11  Equivalent Matrix  138
Table 2-12  Matrix of Matrices  139
Table 2-13  Matrix: Lifecycle/Process to Function  140
Table 2-14  Matrix: Lifecycle/Process to Data  141
Table 2-15  Matrix: Lifecycle/Process to System  144
Table 2-16  Matrix: Function to Data  145
Table 2-17  Matrix: Function to System  146
Table 2-18  Matrix: System to Data  147
Table 3-1  IT Processes and Service Entry Points  162
Table 3-2  Service Consumers and Entry Points  163
Table 3-3  A Configuration Management Maturity Model  203
Table 3-4  Contrasting RASI Approaches  220
Table 3-5  Configuration Data Responsibilities  222
Table 4-1  Possible Systems of Record for Concept of “Application”  255
Table 4-2  CMS, EA and Metadata Comparison  269
Table 4-3  Hosting Progression  281
Table 4-4  Application versus Infrastructure Tracks, and “Zone of Contention”  282
Table 4-5  IT Asset Traceability Matrix  298
Table A-1  Logical versus Physical Configuration Items  358
Foreword I

Hardly a week goes by where we don’t see an article on the Web or in the newspaper article about an organization experiencing a service outage that is blamed on the failure of Information Technology (IT). These articles are simply another proof point reflecting the growing dependence on IT to run the business, grow market opportunities, and innovate. That said, after many years of involvement in IT, I find it is becoming more and more difficult to discern which investments and resources are adding value commensurate with the business expectations. Pressure to understand the real business value of IT has never been greater and answer to this question may dictate whether business demand will increase or decrease. And if we experience a rapid acceleration in business demand, will the floodgates of IT funding open?

In the past 2 years, with the economy in trouble, all of us in IT have made severe budget cuts, automated process, invested in virtualization, delayed or cancelled much innovation, and reduced staffing. Now the requirements from the business are changing as we are moving out of recession and the business is mandating that we start to innovate again to grow the business. This is good, isn’t it? The challenge is, how do we do it with our current resource levels, operational commitments, and ever-changing business needs? All this in a time when the value chain delivering IT is becoming more complex! IT can no longer take years to deliver innovation; we must do it rapidly. The window of competitive advantage is significantly shorter, so IT goods and services must be developed with increased agility, yet still delivered with high service levels.

In order to drive value to the business, we have invested not only in new technology but also in new methods for delivery where rather than building all components organically we are leveraging a supply chain with a focus on driving business outcomes. The new supply chain now includes internally delivered components, outsourcers, cloud computing, and the growing knowledge of the digital native community entering the workforce.
I recall sitting in a diner in Minneapolis with Charlie just a few short months ago discussing the totally risk adverse nature of IT. In my opinion, this posture can no longer prevail. IT must be able to quantify and accept risk where the business mandates and additionally efficiently and effectively run the business. Now if this on its own is not enough, as IT is now delivering the business processes, it too inherits much of the requirement for compliance.

Furthermore, we are seeing the business in many organizations allowed to choose between traditionally centralized IT organizations, building more closely held internal capabilities, or even going directly to the cloud. Clearly, if IT is to survive, it must be world class!

I was speaking to the CIO of a large insurance organization recently, and in the discussion, she mentioned that her IT budget had grown by 2% this year and at the same time the business mandate was to innovate reflecting the strategy of growth and transactions were expected to grow 5%. It was clear that simply automating processes was not enough. The plan included the requirement to make cost savings through application rationalization, outsourcing, and cloud adoption. These savings will provide relief this financial year, but the real sustainable growth must come from the ability to understand business strategy, prioritize resources, and meet demand more quickly.

The methodology to drive this out in a sustainable fashion effectively managed and governed mandates the adoption of Service and Portfolio Management (SPM). My personal definition of SPM is not one out of a framework or standard. I define SPM as “aggregating all demand and then balancing investments against organizational requirements, allowing the business to determine priorities allowing IT to make the appropriate sourcing decisions.”

To assist you in your endeavors, there are many frameworks and standards in place, as Charlie covers in the book. Through application of Lean and enterprise architecture principles, he then builds a nontechnical reference model showing with clarity how the high-level principles espoused by these frameworks can be implemented in an effective integrated system of process, data, and technology.

Personally I have experienced that most frameworks and standards are sometimes used in isolation by zealots who have the framework, standard, or simply process in mind and have lost focus on the business outcome, the cornerstone for IT. Fundamentally the book will guide you through the applicability and appropriateness of each of these frameworks and standards, not only individually but in terms of the business synergies they drive when being used collectively.
In this economic climate with the focus on innovation with the complexities of the supply chain, the delivery of this book is both timely and appropriate and I am sure you, like I, will find this a godsend to driving your personal and organization success for which I wish you well.

Enjoy the book as have I.

Robert E. Stroud, CGEIT
International Vice President ISACA
Former Director itSMF International
Evangelist, Governance, Cloud Computing and Service Management CA Technologies
Architect, in the subtest application of the word, describes one able to engage and arrange all elements of an environment to create a harmonious whole.

From a business perspective, enterprise architecture bridges the operational, technical, managerial, and social elements of the organization. Within IT, enterprise architecture encompasses the people, physical, and intellectual assets across many domains of technical specialization, enabling enterprise-wide systems and processes. When one considers the frequent and tumultuous changes that confront the global, internet-enabled enterprise every day, creating an effective, agile IT ecosystem is a tall order.

The goal of enterprise architecture is to foster an environment that is stable, providing consistent quality and performance, while enabling agility—the flexibility to adapt to quickly changing circumstances without excessive cost or disruption. Oliver Wendell Holmes once said, “I would not give a fig for the simplicity this side of complexity, but I would give my life for the simplicity on the other side of complexity.” This simplicity beyond complexity requires a fine sense of balance that cannot be achieved through rigidity—like an Aikido master, fluid balance is realized through continuous microadjustments that adapt to the changing environment, while maintaining a sense of center.

Today’s marketplace is so dynamic that what works today in one environment may not work tomorrow in another—or even the same—environment. Add to this challenge the breakneck evolution of new and relatively untested technology tools and techniques. Thus, enterprise architecture cannot be a prescriptive discipline. So to deliver value, a book on enterprise architecture must focus on the big picture, offering guiding principles, practices, and models that can be applied in any situation. A book on enterprise architecture must live beyond the short lifetime of any particular technology configuration if it is going to empower us to effectively guide our future decisions.

In this book, Charles Betz navigates these challenges, painting a holistic view of the enterprise from business and technical perspectives. He provides guidance
on navigating and managing the many participants and assets that must work together to deliver real IT value.

Charles has been a longtime proponent of “ERP for IT”—not suggesting that the entire IT organization and its activities can or should be controlled centrally, but rather that IT operations may be organized as interdependent elements, being coordinated and managed through consistent processes. This has led him on many inquiries as follows: What is the true value of IT? Why are IT expenditures ranging into the billions not treated as value streams? Why are the silos in IT among the hardest to break in the modern enterprise? Why do we not have a truly integrated view into IT demand? Building on these inquiries, his perspective of IT applications and services as distinct value streams is an important contribution to the design of a Lean enterprise architecture.

Perhaps most important, this second edition of Architecture and Patterns for IT Service Management, Resource Planning, and Governance: Making Shoes for the Cobbler’s Children places an even greater emphasis on the essential role of continuous improvement. For enterprise architecture to fulfill its purpose, it should strive to create something more closely resembling an organism than an organization: continuously learning, sensing, and adapting to immediate threats and opportunities. By focusing on rapid cycles of discovery and learning, eliminating root causes and practicing “quality at the source,” this book highlights the essential ingredient to successful enterprise architecture: highly engaged people that solve the problems of today, while innovating and creating the vision for tomorrow.

Steve Bell

Author of Lean Enterprise Systems, Using IT for Continuous Improvement
Coauthor of Lean IT: Enabling and Sustaining Your Lean Transformation
Recipient of the 2011 Shingo Prize for Operational Excellence, Research Award
Lean Enterprise Institute faculty
Preface

AN ARCHITECTURE FOR IT ITSELF

The development and delivery of information technology itself must be seen as a system of value.

This is an exciting time for enterprise information technology.

A day’s worth of my Twitter feed shows:

- Leaders at an influential IT Service Management consultancy engaging with and advocating Lean management for IT
- A Prominent young software engineering author grappling with the writings of famous industrial author W. E. Deming (founder of the Quality movement) for the first time
- Fascinating case studies in applying the idea of kanban (a Lean technique from Toyota) to software development and operations

What do all of these have in common? They evidence an increasing awareness that the development and delivery of information technology itself must be seen as a system of value – that the high walls between architecture, development, operations, and risk must be eliminated so that the IT capability delivers unified value to its stakeholders.

This book presents an “outside-in” view of IT as a value system. It is for anyone who wonders how the pieces fit together in the large, enterprise-supporting IT organization. And it is especially intended for anyone who needs architectural precision and detail in this understanding.

Years ago, when I was leading an application team, I met a senior business executive. The conversation went something like this:

Exec: “So, what do you do?”
Me: “I’m building a metadata repository.”

Exec: “Hmm, that sounds like a business we shouldn’t be in.”

Demoralizing? Yes! Yet this interaction, and others like it, sowed the seeds of this book.

Why would an enterprise spend a million dollars or more on building and running something so obscure? Change management systems, asset systems, risk management systems, e-records management, configuration management databases, capacity management systems, metadata repositories – all are so far away from the bottom line. Making the business case was always an uphill struggle.

Yet these tools are built and supported in large information technology (IT) organizations, and significant markets exist for vendors of such software. Why? Because they are an essential part of an overall IT system of value.

My epiphany occurred as I read a 2003 interview with Ralph Szygenda, then Chief Information Officer at General Motors. In that interview, he called for a more integrated, systemic approach to IT management, calling it “Enterprise Resource Planning for IT” (IT-ERP).

In many ways, that single passage inspired the next eight years of my career, including both editions of this book.

I had spent years building ERP systems as a consultant. An ERP system is a large enterprise-spanning system managing one or more major functional areas. Often, it is oriented around a “value chain” like “procure to pay” or “hire to retire.” And as my career moved into focusing on internal IT systems, the question kept coming up – why didn’t all this expensive and often troubled IT activity have a similar approach? I’d already worked in organizations with IT budgets approaching one billion dollars, and had witnessed the ongoing struggles with failed projects and operational outages. There was clearly a value chain and associated processes . . . not working too well.

Szygenda’s call therefore hit me like a thunderbolt and started my quest. My metadata repository had to be part of some value chain. What was it?

Answering this question led me to the major IT frameworks – systematically organized “best” or “good practice” collections, the best known of which are the Information Technology Infrastructure Library (ITIL®), Control Objectives for Information Technology (COBIT®), and the Capability Maturity Model–Integrated (CMMI®). I dove into all of these, studying and comparing them with my daily experiences. (There were interesting contradictions, some of which I discuss in this book.)
This education was invaluable, as it gave me a framework for better understanding the purpose of the systems I was developing. Being “outsourced” to a large consulting firm provided further insight, as I observed the strategy and priorities of that consulting firm’s incoming IT leadership and their concern for “IT demand management.”

I continued my practical education with six years as senior enterprise architect and vice president for one of the largest U.S. banks, focusing not on the business of banking, but the dynamics of IT across a $6 billion spend. During this work, it became clear that of all the major functional areas in any enterprise (sales, marketing, supply chain, outbound logistics, finance, HR, and so forth), IT itself was the least automated! It was and is the least mature in terms of process integration, common data, and centralized systems to bring the diverse actors and concerns together. There was not, and arguably still is not, the equivalent of an ERP system for the IT function itself.

The cynical might say, “So? ERP systems are dreadful products, and there’s been so many failures of ERP, that’s probably a good thing!” But I disagree. ERP has also succeeded for companies that satisfied its formidable culture change requirements, and the market for ERP systems is in the tens of billions of dollars.

Yet this is not a call for an ERP system per se. That’s a solution to a problem we don’t understand well enough yet.

There are more fundamental matters. ERP succeeded where core problems like production scheduling, materials forecasting, and their information requirements were clearly stated and understood, in ways that led to effective automation.

And – the efforts of the IT frameworks notwithstanding – I don’t think we have done that sufficiently for large-scale enterprise IT. Not at the level of rigor required to manage IT holistically as a subsystem of the modern enterprise. Areas like enterprise architecture, IT portfolio management, IT service management, application lifecycle management, and IT governance are all expanding into each other’s domains, and practical guidance on how to coordinate and integrate these evolving areas is still lacking.

That is why I wrote this book. It is the analysis and high-level design of an IT value system.

It is not a book about particular platforms, technologies, or programming languages. You won’t find anything here about virtualization, or Java, or PHP, or networking.

It is an enterprise architecture for the business of IT itself, so it both is, and is not, “technical.” IT is technical in what it manages, but one does not need a
deep technical background to understand the practical processes that go into managing the business of IT.

The book first defines the fundamentals of IT value, identifying the largest, longest-lived flows of IT activity, understanding the major processes directing those flows, and considering the information and automated systems required.

It treats major IT industry frameworks and related literature as a statement of requirements. Process, information, and distributed systems modeling techniques were applied to derive an integrated, vendor-neutral structure, an architectural effort no different from applying those techniques to supply chain, customer relationship management, or human resources.

The approach is inspired by Toyota’s great Lean thought leader Taiichi Ohno and his call to “study the work.” As a consulting enterprise architect for the “business of IT,” I have had direct visibility into large IT organizations for extended periods of time, with the responsibility of investigating many matters large and small across the development and operation of IT services. Out of this experience, my intent has been to produce a next step book for those saying:

“Okay – let’s ‘Run IT Like A Business.’ Now what?”

The result is a practical reference model for structuring your IT processes and capabilities, improving their supporting data and tools, and making decisions about acquiring and integrating new IT process automation capabilities.

You’ll get better value from your existing IT support systems and perhaps find that you don’t need to buy new ones! And by understanding IT itself as a system – a system that engineers and operates other systems – you will get better value from IT for your entire enterprise.

I sincerely hope you find it useful.

Why, and for whom, this book was written

For CIOs, it has been like trying to run a business before the invention of bookkeeping.

Howard Rubin, Meta Group

Do you need to understand a significant Information Technology capability as a system? If so, this book is for you.

---

1CIO Magazine 2004.
This book was written to provide a practical framework for organizations to understand their IT management infrastructure, its complexity and key management areas, in order to improve IT operations, maturity, reliability, and effectiveness.

This book is meant to provide a systemic, architectural overview of enterprise IT management, identifying how value flows within the IT organization, with specific attention to process, data, and enabling systems for the “business of IT” itself.

It is intended for computing and information systems professionals working for large enterprises (governmental, nonprofit, and for-profit). These are professionals employed in IT/IS as a support function, not a primary line of business.

Particularly, the book is aimed at the managers and staff of internally facing IT capabilities:

- IT strategic planning
- Enterprise architecture
- IT portfolio management
- IT process management
- Service management and support
- Project management office
- Risk management
- Security management
- IT audit
- IT quality assurance/continuous improvement

In particular, the book is aimed towards anyone who is trying to design and integrate solutions for these areas.

There is no shortage of guidance on the subject of IT management and governance systems and approaches, but because this guidance comes from a variety of sources and can be highly general, it presents the would-be user with challenges.

Major frameworks such as ITIL® or COBIT® from the Information Security Audit and Control Association require substantial interpretation in order to implement.

Consultants, analyst firms, and vendors seek to fill this void, but their material can be expensive and overly influenced by their business models and incentives.

This practitioner-authored book provides a reference model inspired by both Lean principles and the IT frameworks’ general guidance, to support more
detailed analysis of the specific data, tools, and technologies that enable IT service and portfolio management and governance.

It’s debatable whether the world needs another book on IT project management or IT operations, but it does need more detail on how these two areas relate to each other.

The book is a workable template that will help you reduce waste and redundancy in IT governance, and increase IT agility and transparency. It can be read cover to cover or be used simply as a desk reference. (Many readers of the first edition have indicated that’s how they use it.)

Don’t confuse this work with any of the excellent books on enterprise architecture per se. This book is an application of enterprise architecture principles (process, data, and systems analysis) back upon IT itself (including, paradoxically, the organizations, tools, and processes supporting enterprise architecture in the IT organization).

It is also not a book on Enterprise Resource Planning systems, such as SAP or Oracle. The “ERP for IT” or “IT Resource Planning” theme underlying this book is evocative and provocative, but true “ERP for IT” products are still immature (even as of the second edition of this book).

**DEFINITION: IT ENABLEMENT**

IT Enablement in this book is defined not as the enablement of business processes by IT, but rather the enablement (often automated) of IT’s own business processes.

**Reading this book**

**The Lean house**

This house-like icon is well recognized as a symbol of Lean management. Originally termed the “Toyota Production System House,” TPS now is variously interpreted as “Thinking Person’s System.” This icon will appear when Lean approaches or perspectives are at the forefront of discussion.

**Structure**

IT management is a complex system. It is inherently difficult to structure a linear narrative about it – where to start? What order to tell the story? Where to end? All such decisions are arbitrary, and any narrative requires backwards and forwards references, making the reader’s job difficult.

This book draws on well-established architectural principles in covering the material as a series of “views”:

It is inherently difficult to structure a linear narrative about IT.
Requirements  
Process analysis  
Data model  
System architecture  
Patterns

As The Open Group Architecture Framework states, architecture at the highest level is catalogs, matrices, and diagrams. Catalogs of reusable elements of interest – be they processes, functions, data entities, or systems – are critical building blocks and are best stated as unified lists, at least in summary form.

Matrices are also helpful, as a comprehensive statement of all the dependencies that may be considered between objects of a given type or objects of differing types. For example, a matrix may show the relationship between two functions or the relationship between a process and a data entity.

In general, this book therefore divides itself into three major sections:

- The problem statement, and the themes of Lean IT
- The architecture approach, including the catalogs and matrices
- The patterns – now greatly expanded

The patterns discussion encompasses the larger part of the book and is structured thus:

1. True IT processes and their associated functions
2. Long-lived IT lifecycles

This organization is consistent with the author’s experience and observations in a variety of IT contexts. In general, IT starts with the IT professionals whose positions are funded. These people delineate domains of responsibility, and so the bias towards functional management begins. The applications teams write code and the operations teams run it, and all curse the immature “over-the-wall” relationship.

At some point, the functions realize that they must coordinate systematically, and thus cross-functional processes begin to formalize. Project and Change Management become key governance activities, situated in functional homes but exerting considerable matrix influence via their processes that touch the vast majority of IT activities in the enterprise. Other processes such as Release and Request Management soon follow.

Finally, as the organization approaches its maximum maturity, it becomes clear that all along, certain very long-lived concepts have been operating. This second edition asserts that information technology management and governance, at the largest scale, can be understood in terms of four primary lifecycles.
Here is the revised book structure:

- IT in a world of continuous improvement
- Architectural approach
- The inventories
  - IT lifecycles and processes
  - IT functions
  - IT data
  - IT management systems
- Patterns for IT Processes
  - Accept Demand
  - Execute Project
  - Deliver Release
  - Complete Change
  - Fulfill Service Request
  - Deliver Transactional Service
  - Restore Service
  - Improve Service
  - Retire Service
- Patterns for IT Lifecycles
  - The Application Service Lifecycle
  - The Infrastructure Service Lifecycle
  - The IT Asset Lifecycle
  - The Technology Product Lifecycle
- Conclusion

In this way, the reader seeking a clear discussion of the necessities for any particular major IT lifecycle, process, or function can understand how it works in terms of its activities, its semantics and concepts, the systems and tools needed to run it, and finally benefit from discussion of conceptual patterns that the author has encountered.

In more detail, here is a synopsis of the chapters.

**Chapter 1, “IT in a World of Continuous Improvement”**
This chapter provides a detailed discussion of IT fundamentals and brings in the concepts of Lean IT, IT value, and related topics.

**Chapter 2, “Architecture Approach”**
In the architecture chapter, the book starts with the end in mind, so that the reader has a complete view of the objectives and the fundamental tools used to develop the rest of the book: process, data, and systems architecture and their design patterns.
A simple IT application example is elaborated and discussed in terms of the business architecture that produces and runs it. The paradox of “IT for IT” is explained, and then further detail is provided on the principles and meanings of the architecture views.

Using entity lifecycle analysis, the four major lifecycles are derived, elaborated with nine cross-cutting, shorter-lived processes (both lifecycles and processes are rigorously countable), and contrasted with a functional view derived from current IT frameworks. Data and systems models, inventories, and matrices are presented at a high level with concise definitions. (More detailed definitions are in the appendices.)

**Chapter 3, “Patterns for the IT Processes”**

Though the IT lifecycles may last years, they are crossed by a series of IT processes that last shorter periods. Based on entity lifecycle analysis, the countable true processes are:

- Accept Demand
- Execute Project
- Deliver Release
- Complete Change
- Fulfill Service Request
- Deliver Transactional Service
- Restore Service
- Improve Service
- Retire Service

As organizations mature from a purely functional view to a true process-oriented view, the necessity of systems and data integration becomes clear. Processes cannot cross the functions without a sound basis in normalized, integrated data and systems. Patterns showing this are presented in detail.

**Chapter 4, “Patterns for the IT Lifecycles”**

The four primary lifecycles are:

- The Application Service Lifecycle
- The Infrastructure Service Lifecycle
- The IT Asset Lifecycle
- The Technology Product Lifecycle

Chapter 4 discusses the complex cross-cutting nature of processes and the longer-lived lifecycles; the complexity of IT stems in large part from these unpredictable interactions.
The Application Service Lifecycle
This section focuses on the primary value delivered by production IT services: the “Application Service.” Covering demand, requirements, design, build, release, maintenance, and retirement, this section examines the extended software development lifecycle and the data and systems used to support it (e.g., requirements and issue management, release and deployment, and more).

The Infrastructure Service Lifecycle
This chapter focuses on the necessary technical services that underpin application delivery. Again following an end-to-end lifecycle more focused on technical configuration and less on developing novel business functionality, the various classes of infrastructure services and the necessary processes, data, and infrastructure used to deliver them are discussed.

The Technology Product Lifecycle
Technology products are building blocks – “Oracle 11g,” “Dell PowerEdge R710,” or “64-bit Unix” as opposed to any particular license, asset, or installation. Technology products go through stages of demand, supply, and retirement the same way as the other lifecycles, but are also subject to unique concerns such as patching all known instances of a given product, governing acceptable configurations, or finding a functional replacement for a product whose vendor has discontinued it.

The IT Asset Lifecycle
This chapter focuses on the management of the IT Asset, from demand through disposition. (Notice that Asset and Service are fundamentally distinguished.) It is a briefer chapter than the others, because of the extensive material available in the industry on supply chain management – for physical IT assets at least, the problems and approaches here are very well understood. (Software Asset Management is less well understood.)

Disclaimer
This book’s discussion is broad and cross-disciplinary. However, as an architectural book, it does not discuss important matters of culture and organizational evolution. Other scoping decisions had to be made as well. Apologies are tendered in advance to any domain experts who feel their subject matter is short-changed or ill-treated. Please feel free to forward corrections, comments, or criticisms to the publisher.

Acknowledgements
Many have contributed to the two editions of this book.

First, to Gene Kim, for sending me a copy of The Goal in 2007. I didn’t understand why it was important until 2009 or so; sorry for being a slow learner.
I thank my second-edition reviewers, including Aale Roos, Machteld Meijer, David Moskowitz, Steve Bell, Mehmet Orun, Bob Treadway, Greg Sallee, David Zaucha, and Jez Humble. Thanks also to Steve and Jez for contributing sections to the manuscript.

I pride myself on being first a practitioner, and as such I acknowledge my Wells Fargo colleagues, including Greg Sallee, John Price, Hemant Birari, Kathy Dunn, Wendy Janney, Maureen Backfield, Kathy Daigle, Jim Wallace, Greg Dome, Steve Riley, Bob Carasik Audra Kneer, Mike Gerdes, Roy Taylor, Jane Snyder, Marnie Ellison, Cheri Richardson, Mark Tiggas, John Frisk, Tom Grob, Anita Sukur, Jeff Lamb, Jennifer McArthur, Jeanne Rardon, Chuck Anesi, Yonas Yohannes, Anne Feider, Bob MacFarlane, Heidi Slane, Sherrie Littlejohn, Pat Brennan, and many others. This book reflects 6 years of lessons I learned from and with you in one of the most demanding IT environments on the planet.

I also acknowledge new professional contacts I’ve made in the past years, via blogging and Twitter: Rob England (The IT Skeptic), James Finister, Ken Gonzalez, Rob Stroud, Mike Rosen of Cutter Consortium, Terry Doerscher, Oliver Sims, Nick Gall, Brad Appleton Troy DuMoulin, and many others – you have all influenced this work in some measure.


I also thank the many readers and correspondents I have gained through my weblog, www.erp4it.com.

And of course my first editor Diane Cerra, for seeing merit in what must have seemed like a somewhat far-fetched proposal, and my current editorial team Danielle Miller, Robyn Day and Rachel Roumeliotis.
Charles Betz is Research Director for IT Portfolio Management at Enterprise Management Associates.

From 2005 to 2011, he worked at Wells Fargo as Senior Enterprise Architect and Vice President for IT Portfolio Management and Systems Management. In this role, he guided IT portfolio management and IT service management efforts, as well as consulting broadly across the bank’s $6 billion IT operations regarding process, data, and systems for the “business of IT.”

These efforts included extensive work on IT portfolio management systems; application lifecycle management; the enterprise Configuration Management System; incident, change, and configuration management processes, infrastructure service provisioning and hosting; service catalog, availability processes, and infrastructure; IT financial management, capacity management, SOA and messaging; governance, risk, audit, and compliance; security, enterprise architecture methodology, and many other areas.

As part of the Wells Fargo-Wachovia merger, he established the architectural protocols for the rationalization and orderly dismantling of over 1,500 production applications.

Aligning IT processes via solving master data management problems and enabling IT management system integration were key priorities throughout this work.

Previously, Charlie has held application management, architect, and software engineer positions for Best Buy, Target, and Accenture, specializing in IT governance, ERP systems, enterprise application integration, data architecture and metadata systems, and configuration management. He served as IT manager for the College of Pharmacy at the University of Minnesota, and has also worked in the nonprofit sector.

Charlie holds a summa cum laude B.A. in Political Science (1989) and a Master of Science in Software Engineering (2003), both from the University of
Minnesota. He is an active member of the professional community, belonging to ISACA, IEEE, ACM, ITSMF, and DAMA. He presents frequently both locally and nationally to professional associations and conferences.

He is the sole author of the popular www.erp4it.com weblog.

Charlie lives in Minneapolis, Minnesota, with his wife Sue (a Ph.D. hydrogeologist) and son Keane. His interests include writing, music, cooking, and enjoying Minnesota’s great outdoors.

**Note on Capitalization**
The reader may notice considerable inconsistency in whether terms are capitalized or not. The author and his editors struggled with this question and it became apparent that there are two “modes” that the book has. The first is more narrative and informal, and in this mode there is less capitalization. The second mode is architectural, based on structured vocabularies (“catalogs”) and in this mode the elements of those catalogs are capitalized. Hope this helps.

**Note on Method**
Some topics in IT management are controversial, and the reader may well disagree with specific representations of IT management (e.g. the concept of “application service,” the approach to incident management, particular functional and systems decompositions, and other contentious issues). In such cases, please consider the benefit that the book’s structured approach is furnishing in clarifying disagreements. By using explicit process, data, and system semantics, we can at least be more specific about these controversies and hopefully achieve greater clarity in such industry debates.