WHY THIS BOOK?

This book proposes a comprehensive methodology that guides you through the successful development of large-scale and complex client-server or Web database applications. Doing so, it also explains how to circumvent critical issues and potential dangers that might adversely affect the development of large software systems.

Moreover, this book offers several powerful techniques that can be used by your IT organization to accelerate the construction of large and complex business applications, such as iterative/incremental system development, prototyping, timeboxing, and joint facilitated sessions with users.

Similarly, the book strongly recommends the use of component-based application development architectures to maximize reusability and hence reduce the cycle time associated with the system development process.

WHAT MAKES THIS BOOK UNIQUE?

This book addresses the various sets of technical activities that must be conducted to effectively develop a sound client-server or Web database system. However, rather than concentrating on the “how to,” this publication primarily focuses on the “what to do.”

The book contains numerous guidelines, suggested best practices, hints, and checklists that are based on practical experience and that can be advantageously used by practitioners to successfully develop quality-oriented client-server and Web-based software systems.
This book also emphasizes the importance of enterprises investing efforts in developing and implementing a robust yet flexible enterprise technology infrastructure. An efficient enterprise technology infrastructure is mandatory to effectively support the accelerated development of large software systems. The current enterprise technology infrastructures must be re-engineered or extended to appropriately account for the introduction of new development tools, technologies, and techniques. The new enterprise technology infrastructures must be built on industry-wide standards to accommodate the emergence of new and improved technologies, as time goes by.

WHAT DOESN’T THIS BOOK CONTAIN?

This book does not address the set of project management processes that must be conducted to effectively plan, organize, manage, track, and control the development process of large software systems. It addresses solely the technical activities that are involved in developing large software systems. However, a companion publication will likely be developed in the near future, to address the most critical project management best practices and therefore complement the technical information provided in this current book.

This book is not your typical “one-minute” methodology cookbook to client-server and Web database application development. Its content is at the right level of detail for large software development undertakings. In fact, the book is primarily intended for “real-world” software practitioners who must work in the “trenches,” often with newer technologies, and yet are hard pressed by their customers to deliver workable system solutions in a compressed timeframe.

As such, this book does not promise you silver bullets. It does not attempt to fool you into thinking that large-scale software development efforts are easy and can be achieved overnight, only with the help of the latest RAD tool available on the market, and without the support of a solid information technology infrastructure. Those involved in developing large and complex software systems know better than this the perils of any particular software development process.

WHO SHOULD READ THIS BOOK?

This book should be read by IT executives, managers, project leaders, developers, specific information technology specialists, software testers, and operations staff around the world who are involved in developing large and complex client-server or Web database applications.

The book is also intended for modern IT organizations who are currently in the process of renewing themselves while planning the introduction of new client-server and Web technologies in their enterprise. These organizations will then be interested in reading the chapter titled Client-Server and Web Technology Architecture and Support
Services. This chapter provides the readers with a holistic view of the major issues involved in planning, designing, and implementing the new type of enterprise technology infrastructure that is required to successfully nurture the development, deployment, and support of large client-server or Web database applications.

Finally, the book can be used as a textbook for college or university courses on the development of large client-server and Web database systems.

BOOK ROADMAP

Looking at Figure P-1, the book consists of an introduction chapter, which is then followed by three broad logical sections, identified as sections A, B, and C.

![Book roadmap.](image)

Following is a brief description of the introduction chapter and the three major sections A, B, and C.
INTRODUCTION

The introduction chapter provides an overview of the book and discusses how it is organized. It presents the fundamental concepts that are critical to support the proposed client-server and Web database application development methodology. The process-driven system development methodology that is provided in this book is suited for a variety of business projects of different sizes and complexities. However, it is particularly well-adapted for large and complex client-server or Web database applications.

SECTION A

Section A contains the five core chapters that describe in detail the particular system development phases and set of technical activities that should be performed to successfully build a quality client-server or Web database application.

Table P-1 provides a brief description of these five chapters.

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 2: Survey Phase</td>
<td>The preliminary business needs that point toward a client-server/Web application system solution are identified and documented during this stage. A sketch of the initial scope and boundaries of the proposed system are outlined jointly with the customers. Similarly, the development team assesses the major project assumptions, risks, and constraints. The survey phase technical activities can be conducted either through a series of joint facilitated sessions, with the active participation of the users, or in a more traditional system development approach.</td>
</tr>
<tr>
<td>Chapter 3: Analysis Phase</td>
<td>A blueprint of the major application data and functional requirements is quickly captured with the direct involvement of the customers. A series of joint facilitated sessions can be used specifically for this purpose. The preliminary data and process application distribution requirements are also documented at this stage, along with additional system requirements, such as the security, performance, and hardware/software/networking application needs. If necessary, the system is partitioned into smaller chunks, where core functions can be implemented more rapidly by the development team, either in sequence or in parallel. Moreover, a first stab at defining the system data conversion, training, and testing strategies is done with the users during this phase.</td>
</tr>
</tbody>
</table>
Finally, a live system prototype is quickly constructed to demonstrate the graphical user interface and some of the core functionality elements of the proposed client-server or Web database application. Reusable GUI classes or application system templates are used to speed up the delivery of the initial application graphical interface. This fully interactive prototype is used to firm up the initial set of critical user requirements that center on the application graphical interface.

Chapter 4: Design Phase

During this phase, the design of the client-server or Web-based system solution is gradually refined and augmented with all the necessary automated solution details. The application reusable GUI classes and system templates may be created or extended, where applicable.

The sets of reusable functions that can be effectively reutilized across the application are developed first, prior to developing the application custom functions.

The detailed design of the database(s) that is(are) required to support the system information needs is finalized, including stored procedures and triggers.

The data and process distribution design solutions are revisited and expanded with more detailed information. Similarly, the system training, data conversion, and testing strategies are refined and augmented with more detailed information. Based on the design specifications, high-level test cases are defined by the developers toward the end of the design phase.

Chapter 5: Construction Phase

The detailed functional components of the client-server or Web application are coded, unit tested, and then gradually integrated into a complete and robust application system.

In parallel to the coding and testing activities, the development team gradually sets up the initial user training and data conversion system environments.

Chapter 6: Implementation Phase

During this phase, user training is completed. Final user acceptance tests are performed by selected groups of user representatives.

Finally, the new client-server or Web system is transferred into a production environment, along with its converted data. The maintenance environment is closely monitored, and the client-server system evolution plan is activated.

During a short period of time, the application is closely monitored and fine-tuned to achieve optimal performance.

A post-mortem study is launched to identify what went wrong and what went well during the project. The outcome of the post-mortem study is used to revise and confirm the best internal software development practices.

Table P–1  Section A (cont.)
SECTION B

Section B logically consists of Chapters 8, 9, and 11. The technical information provided in each of these chapters directly supports and/or reinforces the system development methodology proposed in Section A.

Table P-2 provides a brief description of these three chapters.

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 8: Client-Server and Web Testing</td>
<td>The development of large and complex client-server or Web database applications emphasizes the necessity for a flexible yet effective testing process, specifically adapted to these newer technology paradigms. This chapter proposes a software-test life cycle that promotes the development of quality-oriented client-server and Web database systems. Some of the new components that need to be tested in a client-server and/or Web environment include Graphical User Interfaces (GUI), servers, networks, and a variety of middleware technologies. Besides the more traditional types of testing scenarios, I discuss in this chapter new categories of test cases that are specifically intended to verify client-server and Web database systems. These include a variety of test cases such as client configuration testing, server or client-workstation load testing, database server testing, and Web browser application interface testing. I also discuss the relevance to implement or not automated testing tools in a client-server or Web development environment in this chapter.</td>
</tr>
<tr>
<td>Chapter 9: Client-Server and Web Technology Architecture &amp; Support Services</td>
<td>This chapter presents a holistic view of the major technology infrastructure components that must be put in place to successfully enable and support the rapid development and deployment of large client-server or Web database applications across and sometimes beyond the enterprise. I also discuss in this chapter the most critical enterprise systems management activities and support services that must be implemented to efficiently manage and support the distributed client-server and Web technology infrastructures. I also include a brief overview of the most desirable characteristics that an integrated software development tool suite must possess to construct effective Web database applications.</td>
</tr>
</tbody>
</table>
SECTION C

Section C consists of Chapters 7, 10, and 12. Both Chapters 7 and 10 present some specific concepts and techniques that can be used to help accelerate the system development process proposed in this book, such as joint facilitated sessions and software reusability. Chapter 12 covers some fundamental graphical user interface design concepts.

Table P-3 provides a brief description of these three chapters.

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 7: Software Reusability</td>
<td>This chapter discusses the merits and challenges of implementing a software development process that encourages the construction of reusable software components. It also provides some basic guidelines to help construct software reusable components across the enterprise. Finally, I discuss briefly the major characteristics of three popular distributed software component architectures, namely: DCOM/ActiveX, CORBA, and JavaBeans for Enterprise.</td>
</tr>
<tr>
<td>Chapter 10: Joint Facilitated Sessions</td>
<td>This chapter describes the set of guidelines that are recommended throughout this book to plan, prepare, and successfully conduct effective joint facilitated sessions. The joint facilitated session process is frequently used during the analysis and design phases of large projects to dynamically capture, with the active participation of the users, the major external requirements of the system. You can also use joint facilitated sessions advantageously during the design, construction, and implementation phases.</td>
</tr>
</tbody>
</table>
DISCLAIMER

Despite the great attention provided by the author to ensure up-to-date and accurate technical information, this publication could contain technical inaccuracies or typographical errors. All Web site addresses provided in this publication were carefully verified for completeness and accuracy purposes. However, due to the dynamic nature of the Web, no one can guarantee that these Web site addresses will remain unchanged over time.

The views and opinions expressed in this publication are solely those of the author and do not represent the views of anyone else. The names of the vendors and/or products mentioned in this publication should not be interpreted as a recommendation from the author. All references of vendor products and/or services in this book are done as a means of providing background information to discussions that relate to the use of different tools and technologies, and are for illustrative purposes only. Any reference to vendor-related products and/or services made in this publication should not be viewed or construed as either positive or negative comment on those products and/or services, neither should the inclusion of a product or vendor, or the omission of a product or vendor on the part of the author.

TRADEMARK ACKNOWLEDGMENT

The following list recognizes the commercial and intellectual property of the trademark holders whose products and/or services are mentioned in this book. All other products, services, or company names mentioned herein are claimed as trademarks and trade names by their respective companies. Any omission from this list is inadvertent.

The browser-based screen-prints presented in this publication were made using Netscape Communicator, with the permission of Netscape Communications Corporation. Netscape Communicator is a trademark of Netscape Communications Corpo-
ration. Netscape Communications Corporation has not authorized, sponsored, or endorsed, or approved this publication and is not responsible for its content. Netscape and the Netscape Communications Corporate Logos are trademarks and trade names of Netscape Communications Corporations. All other product names and/or logos are trademarks of their respective owners.

The clip-art illustrations used in this publication are extracted from: Masterclips, which is a trademark of IMSI Inc. and Corel Gallery, which is a trademark of Corel Corporation.

ActiveX is a trademark of Microsoft Corporation.
CORBA is a trademark of Object Management Group.
DB2 is a trademark of IBM Corporation.
DCOM is a trademark of Microsoft Corporation.
Encina is a trademark of IBM Corporation.
IPX/SPX is a trademark of Novell Corporation.
Java and JavaBeans are trademarks of Sun Microsystems.
JavaScript is a trademark of Netscape Communications Corporation.
Lotus Notes and Domino are trademarks of IBM Corporation.
NetWare is a trademark of Novell Corporation.
ODBC and OLE are trademarks of Microsoft Corporation.
Oracle7/Oracle8 are trademarks of Oracle Corporation.
UNIX is a registered trademark licensed exclusively through X/Open Company.
Windows is a trademark of Microsoft Corporation.