Software Engineering, software development and software reuse are important issues to all organizations. Getting the most out of software packages by ensuring effective development, testing and use can save money and improve business practices. As the implications become more widespread, researchers, practitioners, academicians and information systems managers alike need to have access to the most up-to-date research and practice in software engineering and development. The chapters in this book address the timely topics of auditing software engineering processes, enterprise resource planning and software reuse and other relevant applications and technologies. From academics reporting research findings to developers reporting on best practices, the authors of these chapters are from diverse cultural and industry backgrounds and provide insights from their varied experiences.

Chapter 1 entitled, “Computer Aided Method Engineering” by Ajantha Dahanayake of Delft University of Technology (The Netherlands) discusses a conceptual model to specify the functionality of a support environment. The chapter first presents a review of basic concepts and approaches for deriving models for computer aided Software Engineering (CASE) environments. The chapter then offers an informal description of service component concepts used to derive a generic framework. Finally, the chapter outlines a configuration of service components to support computer aided method engineering (CAME).

Chapter 2 entitled, “Architecture and Implementation Issues” by Ajantha Dahanayake of Delft University of Technology (The Netherlands) concentrates on using a representation formalism to construct a problem specific CAME environment. Such an automated support environment must be provided for the information systems design state in particular for the required UpperCASE tools according to the methods chosen for the problem specific environment.

Chapter 3 entitled, “Future Directions in CASE Repositories” by Ajantha Dahanayake of Delft University of Technology (The Netherlands) reports on how CAME environments provide a fully flexible environment for method specification and integration and can be used for information systems design activities. The
chapter then discusses how this theory can lead to the design of the architecture of such an environment.

Chapter 4 entitled, “Audit of a CASE Environment” by Mario Piattini of the Universidad de Castilla-La Mancha and Jesús García Tomás of Universidad Politecnica de Madrid (Spain) addresses the questions that must be answered when auditing a CASE environment. The chapter reflects upon themes that have been dealt with in the literature from the perspective of an information systems audit. The authors introduce the basic concepts of an information systems audit and analyze the risks that need to be addressed when installing a CASE tool.

Chapter 5 entitled, “Process Model for Round-trip Engineering with Relational Database” by Leszek A. Maciaszek of Macquarie University (Australia) identifies difficult round-trip scenarios and defines the processes needed to handle those scenarios. The processes conform to the current state-of-the practice in forward and revere engineering. The chapter then discusses the limitations of a tool-driven round-trip engineering.

Chapter 6 entitled, “Achieving Effective Software Reuse for Business Systems” by Daniel Brandon, Jr. of Christian Brothers University (USA) reports on software reuse including discussions of both literature research and design/coding research. The chapter further presents an approach for software reuse in the development of business systems. The approach discussed in the chapter is based on object-oriented technology and provides for both the specification and enforcement of software reuse and corporate standards.

Chapter 7 entitled, “The Future of Software Development” by Karen Church and Geoff te Braake of Port Elizabeth Technikon (South Africa) discusses the results of two surveys as they illustrate the trends in software development. The authors look at the history of software development and its evolution. The authors discuss the evolution of programming languages, coding styles and software architecture. It further looks at the growing importance of user interfaces and describes future trends.

Chapter 8 entitled, “Understanding the Role of Use Case in UML: A Review and Research Agenda” by Brian Dobing of the University of Lethbridge and Jeffrey Parsons of Memorial University of New Foundland (Canada) focuses on two components of UML: use cases and class models. The authors consider the appropriateness of use cases as a component of an object-oriented modeling language by examining their role as a tool for communicating with users. The authors further consider the relationship between use cases and the class models that are developed from them. Finally, the authors offer a framework for empirical research to evaluate the value of use cases and their relationship to class models in UML.

Chapter 9 entitled, “Enhancing a Rigorous Reuse Process with Natural Language Requirement Specifications” by L. Felice, C. Leonardi, L. Favre and V.
Mauco of the Universidad Nacional del Centro de la Pcia. de Buenos Aires (Argentina) proposes a systematic reuse approach that integrates natural language requirement specifications with formal specifications in RAISE Specification Language. It addresses the problems associated with reusability techniques, discusses the reusability process and provides a concrete example of the principles discussed.

Chapter 10 entitled, “Extended Spatiotemporal UML: Motivations, Requirements, and Constructs” by Rosanne Price of Monash University (Australia), Nectaria Tryfona and Christian Jensen of Aalborg University (Denmark) presents a conceptual modeling language for spatiotemporal applications that offers built-in support for capturing spatially referenced, time-varying information. Specifically, the well-known object-oriented unified modeling language is extended to capture the semantics of spatiotemporal data. The chapter gives examples to illustrate the simplicity and flexibility of this approach.

Chapter 11 entitled, “A Design Method for Real-Time Object-Oriented Systems Using Communicating Real Time State Machines by Eduardo B. Fernandez, Jie Wu and Debera R. Hancock of Florida Atlantic University (USA) proposes an object-oriented analysis and design methodology that augments the traditional Unified Modeling Language dynamic model with real-time extensions based on high-level parallel machines and communication notations from Communicating Real-Time State Machines. The chapter also provides an example of the proposed methodology as it applies to an automated passenger train system.

Chapter 12 entitled, “Java Integrated Development Environments’ Support for Reuse-Oriented Software Development” by Jenni Ristonmaa, Jarmo Ahonen and Marko Forsell of the University of Jyväskylä (Finland) reports on the authors’ study of three Java IDEs and how they support reuse-oriented software development. The authors derived the evaluation criteria from a known reuse model. They conclude that current Java IDEs need to improve their support for the reuse process.

Chapter 13 entitled, “Information Modeling and Method Engineering: A Psychological Perspective” by Keng Siau of the University of Nebraska-Lincoln (USA) proposes the use of cognitive psychology as a reference discipline for information modeling and method engineering. The chapter reviews theories in cognitive psychology and applies them to information modeling and method engineering.

Chapter 14 entitled, “Load-Testing of Web Site Applications: Analysis and Recommendations” by Vijay Raghavan of Northern Kentucky University (USA) discusses the need and benefits of load testing. The author provides criteria for developing a metrics program for load testing Web site applications. Finally, the
chapter concludes that it is critical for organizations deploying Web sites to develop a load-testing plan that includes all aspects of site development.

Chapter 15 entitled, “Component-Based ERP Design in a Distributed Object Environment” by Bonn-Oh Kim of Seattle University and Ted Lee of Memphis State University (USA) outlines strategic steps needed to wield a dominant power in the future Enterprise Resource Planning (ERP) market. The steps discussed are: knowledge modeling, componentization of domain knowledge, implementation of componentized domain knowledge, and marketing strategies for domain knowledge components.

Chapter 16 entitled, “Knowledge and Object-Oriented Approach for Interoperability of Heterogeneous Information Management Systems” by Chin-Wan Chung and Chang-Ryong Kim of the Korea Advanced Institute of Science and Technology (Korea) and Son Dao of Hughes Research Laboratory (USA) incorporates concepts and constructs associated with the knowledge and object-oriented paradigms with abstract views, procedures, encapsulation, inheritance and class composition hierarchies to resolve problems.

Chapter 17 entitled, “A Recursive Approach to Software Development” by Shirley Becker of the Florida Institute of Technology and Alan Jorgensen of Advanced Engineering Technology (USA) proposes that a recursive software development process be used as a means of managing the complexity of today’s software systems. The authors advocate that the recursive approach has the flexibility needed to perform development activities in any order to ensure that systems requirements are met.

Chapter 18 entitled, “Adding Alternative Access Paths to Abstract Data Types” by Xavier Franch and Jordi Marco of the Universitat Politecnica de Catalunya (Spain) presents a proposal for developing efficient programs in the abstract data type programming framework, keeping the modular structure of programs and without violating the information hiding principle. The proposal focuses in the concept of shortcut as an efficient way of accessing data, an alternative to using primitive operations of ADT.

Chapter 19 entitled, “Relational Data Modeling for Geographic Information Systems” by Lawrence West, Jr. of the University of Central Florida and Brian Mennecke of East Carolina University (USA) addresses data modeling problems inherent in the use of geographic information systems that are not adequately covered by traditional modeling techniques. This chapter proposes relational modeling techniques that document organizational data integrity rules when systems that include spatial data are developed for more widespread use.

Chapter 20 entitled, “Software Process Models are Software Too: A Domain Class Model for Software Process Models” by Daniel Turk of Colorado State University and Vijay Vaishnavi of Georgia State University (USA) focuses on the
domain class model as an example of one type of model that could be produced if an approach such as the Unified Process were used in the process modeling domain. While identifying the conceptual needs of process modeling systems, these models leave open the choice of how to formalize and implement actual solutions. The authors develop a domain class model for process models as an example.

Chapter 21 entitled, “A Process Model for Certification of Product and Process” by Hareton Leung and Vincent Li of Hong Kong Polytechnic University (Hong Kong) identifies two process models, one for process certification and another for product certification. The authors then propose a certification process for Commercial Off the Shelf (COTS) product and its development process. Finally, the authors develop a model of certification process for both product certification and development process certification.

As businesses seek to improve their use of software, the chapters in this book will provide insightful theoretical discussion as well as practical examples and case studies illustrating the concepts discussed. Researchers, academician, students, or software engineers will find the information contained herein invaluable as a starting point or a supplement to their research and practice. From how to improve reuse techniques to how to more efficiently develop and use models, this book contains practical and theoretical information which is essential to those seeking to fully understand the emerging field of software engineering.

IRM Press
January 2002