The constantly changing landscape of Software Design and Development makes it challenging for experts and practitioners to stay apprized of the field’s most up-to-date research. That is why Information Science Reference is pleased to offer this four-volume reference collection that will empower students, researchers, and academicians with a strong understanding of critical issues within Software Design and Development by providing both broad and detailed perspectives on cutting-edge theories and developments in the field. This collection is designed to act as a single reference source on conceptual, methodological, technical, organizational, and managerial issues, as well as provide insight into emerging trends and future opportunities within the discipline.

*Software Design and Development: Concepts, Methodologies, Tools, and Applications* is organized into eight distinct sections that provide comprehensive coverage of important topics. The sections are (1) Fundamental Concepts and Theories, (2) Development and Design Methodologies, (3) Tools and Technologies, (4) Organizational and Social Implications, (5) Utilization and Application, (6) Managerial Impact, (7) Critical Issues, and (8) Emerging Trends. The following paragraphs provide a summary of what readers may expect from this invaluable reference tool.

Section 1, “Fundamental Concepts and Theories”, provides an overview of both traditional and recently developed paradigms in Software Design and Development. The first chapter in this section, *A Roadmap for Software Engineering for the Cloud* by Abhishek Sharma and Frank Maurer serves as an effective introduction in this regard, presenting readers with an in-depth look at how the emerging—and increasingly prevalent—field of cloud computing, in conjunction with classic software development policies and practices, has led to an evolved understanding of the subject. The remaining chapters in this section introduce a wide variety of functional areas and applications within the field of software engineering, notably *Agile, Lean, and Service-Oriented Development, Continuum, or Chasm* by Juha Rikkilä; *Model-Driven Engineering, Services and Interactive Real-Time Applications* by Luis Costa, Neil Loughran, and Roy Gronno; and *A Software Engineering Framework for Context-Aware Service-Based Processes in Pervasive Environments* by Zakwan Jaroucheh, Xiaodong Liu, and Sally Smith, topics which are explored in greater detail in later sections.

Section 2, “Development and Design Methodologies”, discusses techniques and best practices in Software Design and Development. In particular, the chapters in this section explore model-based design practices, such as *Model-Driven Development of Mobile Information Systems* by Ralf Bruns and Jürgen Dunkel and *A Method for Model-Driven Information Flow Security* by Fredrik Seehusen and Ketil Stølen. Additionally, this section includes information pertaining to topics introduced in the first section, including *SaaS Requirements Engineering for Agile Development* by Asif Qumer Gill and Deborah Bunker, *Data Mining Techniques for Software Quality Prediction* by Bharavi Mishra and K. K. Shukla, and *Tool Based Integration of Requirements Modeling and Validation into Business Process*
Modeling by Sven Feja, Sören Witt, and Andreas Speck. The final chapter in this section, Robust Network Services with Distributed Code Rewriting by Thomas Meyer and Christian Tschudin, serves as an effective transition into the third section with its focus on programming languages, the fundamental building blocks of software development.

Section 3, “Tools and Technologies”, elaborates on the previous foundational chapters to explore the basic building blocks of Software Design and Development. The initial chapters of this section describe domain-specific languages (DSLs) and how they are used in software engineering processes. In particular, the first chapter, MoDSEL by Ersin Er and Bedir Tekinerdogan introduces readers to a model-driven software development tool, and DSLs in Action with Model Based Approaches to Information System Development by Ivan Lukovic, Vladimir Ivancevic, Milan Celikovic, and Slavica Aleksic investigates the use of DSLs in the development of Information Systems. Other chapters in this section describe additional tools integral in the development of successful software applications, topics ranging from alternative programming models and languages such as Unified Modeling Language, debugging and testing tools, and mobile applications. This section’s final chapter, Towards an Integrated Personal Software Process and Team Software Process Supporting Tool by Ho-Jin Cho, Sang-Hun Lee, Syed Ahsan Fahmi, Ahmad Ibrahim, Hyun-II Shin, and Young-Kyu Park, segues into the next section, which deals with social software and software development teams.

Section 4, “Organizational and Social Implications”, explores the impact of Software Design and Development on users and developers. The chapters in this section focus primarily on two main topics: software development teams and social networking systems. Both include discussions of the implications of cloud computing on the life cycle of software systems. Trust Building Process for Global Software Development Teams by Adrián Hernández-López, Ricardo Colomo-Palacios, Ángel García-Crespo, and Pedro Soto-Acosta investigates the evolving landscape of software development, where programming teams are commonly scattered across the globe and often work remotely. Similarly, Establishing Ethos on Proprietary and Open Source Software Websites by Kevin Brock presents the user viewpoint of software stored and shared in the cloud. Perhaps the most important topic in this section, however, is the consideration of security in software systems, whether online or off. Security Risks in Cloud Computing by Belén Cruz Zapata and José Luis Fernández Alemán acts as a touchstone for these chapters, providing readers with a survey of some of the critical threats to software development in the cloud.

Section 5, “Utilization and Application”, follows the theoretical first half of this reference with chapters on practical applications of Software Design and Development. A field as pervasive as modern software engineering will naturally impact a diverse array of disciplines. In particular, this section covers applications in language learning (Design and Development Considerations for a Multilingual Digital Library by Anne R. Diekema), genetic engineering (A Comparative Analysis of Software Engineering Approaches for Sequence Analysis by Muneer Ahmad, Low Tang Jung, and Noor Zaman), robotics (A Hierarchically Structured Collective of Coordinating Mobile Robots Supervised by a Single Human by Choon Yue Wong, Gerald Seet, Siang Kok Sim, Wee Ching Pang), education (Reengineering The Portal to Texas HistorySM by Kathleen Murray, Mark Phillips, William Hicks, Neena Weng, and Dreanna Belden), and more. The final two chapters bridge the gap between this section and the next with information on software process improvement, ending with The Development of International Standards to Facilitate Process Improvements for Very Small Entities by Claude Laporte and Edgardo Palza Vargas.

Section 6, “Managerial Impact”, addresses common concerns of managers and leaders of Software Design and Development teams. The first chapter, Software Process Improvement for Small and Very Small Enterprises by Mohammad Zarour, Alain Abran, and Jean-Marc Desharnais, picks up where the
previous section left off by exploring how organizations can evaluate and build upon existing software systems and processes. An additional focus in this section is on optimization of software services, a topic explored by Jonathan Lee, Shang-Pin Ma, Shin-Jie Lee, Chia-Ling Wu, and Chiung-Hon Leon Lee in their chapter Towards a High-Availability-Driven Service Composition Framework and by Salah Merad, Rogério de Lemos, and Tom Anderson in A Game Theoretic Solution for the Optimal Selection of Services. The section concludes with three chapters on managing software development teams, notably Managing Software Projects with Team Software Process (TSP) by Salmiza Saul Hamid, Mohd Hairul Nizam Md Nasir, Shamsul Sahibuddin, and Mustaffa Kamal Mohd Nor, which aims to avoid failure of software projects through effective project management and teambuilding.

Section 7, “Critical Issues”, objectively evaluates many of the methods and strategies commonly employed in Software Design and Development environments. The chapters in this section apply to the gamut of software development processes, from pre-coding through final testing and launch. The first chapter, Quality-Driven Software Development for Maintenance by Iwona Dubielewicz, Bogumila Hnatkowska, Zbigniew Huzar, and Lech Tuzinkiewicz, argues that while software maintenance is traditionally viewed as a post-launch task, it should really be considered throughout the software development process, particularly in the early stages. The central chapter, Business Intelligence and Agile Methodology for Risk Management in Knowledge-Based Organizations by Muhammad Mazen Almustafa and Dania Alkhalili discusses methods for mitigating risk and evaluating software development processes. Finally, the section concludes with How Much Automation can be Done in Testing? by Izzat Alsmaidi, an evaluation of best practices in the latter stages of development just prior to launch, ensuring the highest quality with each completed product.

Section 8, “Emerging Trends”, peers into the future of Software Design and Development, using current advances to extrapolate new innovations, applications, and possibilities. The primary focus of these chapters is the reuse and security of software systems. Adaptive Web Services for Modular and Reusable Software Development by Javier Cubo and Ernesto Pimentel proposes a framework for reusing components and services of software systems through public interfaces. Later in this section, Innovative Strategies for Secure Software Development by Punam Bedi, Vandana Gandotra, and Archana Singhal discusses proactive threat management techniques for software systems. Additional topics covered in this final section include human-computer interaction, mobile software systems, model-based architectures, object-oriented programming, and DSLs. The final chapter in this extensive multi-volume reference, Service Composition Verification and Validation by Manuel Palomo-Duarte, brings the text full circle with a discussion of web services, systems security, and Software Design and Development in a cloud-based society.

As a comprehensive collection of research on current findings related to the development of interdisciplinary technologies, Software Design and Development: Concepts, Methodologies, Tools, and Applications provides researchers, administrators, and all audiences with a complete understanding of the latest advances, applications, and concepts in Software Design and Development. Although the primary organization of the contents in this multi-volume work is based on its eight sections, offering a progression of coverage on the important concepts, methodologies, technologies, social issues, applications, administrative considerations, critical concerns, and emerging trends, the reader can also identify specific content by utilizing the extensive indexing system found at the end of each volume. Given the vast number of issues concerning usage, successes and failures, policies, strategies, and applications of Software Design and Development in countries around the world, Software Design and Development: Concepts, Methodologies, Tools, and Applications addresses the demand for a resource that encompasses the most pertinent research on the technologies being employed to globally bolster the knowledge and implementation of Software Design and Development.