Software Engineering For Self Adaptive Systems Lecture Notes In Computer Science Programming And Software Engineering

Partial Contents1: Software Ascents- Components of Adaptive Software Development2: Thriving at the Edge of Chaos- The Adaptive Development Model3: The Project Mission- Identify the Mission- Create Mission Artifacts- Share Mission Values- Focus on Results4: Planning Adaptive Development Cycles- Adaptive Planning Techniques- The Evolving World of Components5: Great Groups and the Ability to Collaborate- Using Complexity Concepts to Improve Collaboration- Joint Application Development6: Learning: Models, Techniques, and Cycle Review Practices- Software Inspections- Project Postmortems7: Why Even Good Managers Cause Projects to Fail- Disruptive Technologies- No Silver Bullet8: Adaptive Management- The Progression from Process to Pattern9: Workstate Life Cycle Management- Managing Workflow in an Adaptive Environment10: Structural Collaboration- Eight Guidelines for Applying Rigor to Project Work11: Managing Project Time Cycles- Plan the Project12: Dawdling, McLuhan, and Thin Air- Organizational Growth- Surviving in Thin AirBibliographyIndex

Cyber-physical systems play a crucial role in connecting aspects of online life to physical life. By studying emerging trends in these systems, programming techniques can be optimized and strengthened to create a higher level of effectiveness. Solutions for Cyber-Physical Systems Ubiquity is a critical reference source that discusses the issues and challenges facing the implementation, usage, and challenges of cyber-physical systems. Highlighting relevant topics such as the Internet of Things, smart-card security, multi-core environments, and wireless sensor nodes, this scholarly publication is ideal for engineers, academicians, computer science students, and researchers that would like to stay abreast of current methodologies and trends involving cyber-physical system progression.

This book constitutes the refereed proceedings of the Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns, REFLECTION 2001, held in Kyoto, Japan in September 2001. The revised eleven long papers, seven short papers, and eight posters presented were carefully reviewed and selected from 44 submissions. The book offers topical sections on reflection and SOC in Java, software adaptation using reflection and SOC techniques, reflective middleware for distributed mobile applications, testing and verification, foundations of reflection and SOC, and software methodologies for SOC.

During the last few years, software evolution research has explored new domains such as the study of socio-technical aspects and collaboration between different individuals contributing to a software system, the use of search-based techniques and meta-heuristics, the mining of unstructured software repositories, the evolution of software requirements, and the dynamic adaptation of software systems at runtime. Also more and more attention is being paid to the evolution of collections of inter-related and inter-dependent software projects, be it in the form of web systems, software product families, software ecosystems or systems of systems. With this book, the editors present insightful contributions on these and other domains currently being intensively explored, written by renowned researchers in the respective fields of software evolution. Each chapter presents the state of the art in a particular topic, as well as the current research, available tool support and remaining challenges. The book is complemented by a glossary of important terms used in the community, a reference list of nearly 1,000 papers and books and tips on additional resources that may be useful to the reader (reference books, journals, standards and major scientific events in the domain of software evolution and datasets). This book is intended for all those interested in software engineering, and more particularly, software maintenance and evolution. Researchers and software practitioners alike will find in the contributed chapters an overview of the most recent findings, covering a broad spectrum of software evolution topics. In addition, it can also serve as the basis of graduate or postgraduate courses on e.g., software evolution, requirements engineering, model-driven software development or social informatics.

This book focuses on the topic of improving software quality using adaptive control approaches. As software systems grow in complexity, some of the central challenges include their ability to self-manage and adapt at run time, responding to changing user needs and environments, faults, and vulnerabilities. Control theory approaches presented in the book provide some of the answers to these challenges. The book weaves together diverse research topics (such as requirements engineering, software development processes, pervasive and autonomic computing, service-oriented architectures, on-line adaptation of software behavior, testing and QoS control) into a coherent whole. Written by world-renowned experts, this book is truly a noteworthy and authoritative reference for students, researchers and practitioners to better understand how the adaptive control approach can be applied to improve the quality of software systems. Book chapters also outline future theoretical and experimental challenges for researchers in this area. Contents:Prioritizing Coverage-Oriented Testing Process — An Adaptive-Learning-Based Approach and Case Study (Fevzi Belli, Mubariz Eminov, Nida Gökçe & W Eric Wong)Statistical Evaluation Methods for V&V of Neuro-Adaptive Systems (Y Liu, J Schumann & B Cukic)Adaptive Random Testing (Dave Towey)Transparent Shaping: A Methodology for Adding Adaptive Behavior to Existing Software Systems and Applications (S Masoud Sadjadi, Philip K McKinley & Betty H C Cheng)Rule Extraction to Understand Changes in an Adaptive System (Marjorie A Darrah & Brian J Taylor)Requirements Engineering Via Lyqpunov Analysis for Adaptive Flight Control Systems (Giampiero Campa, Marco Mammarella, Mario L Fravolini & Bojan Cukic)Quantitative Modeling for Incremental Software Process Control (Scott D Miller, Raymond A DeCarlo & Aditya P Mathur)Proactive Monitoring and Control of Workflow Execution in Adaptive Service-based Systems (Stephen S Yau & Dazhi Huang)Accelerated Life Tests and Software Aging (Rivalino Matias Jr & Kishor S Trivedi) Readership: Students, researchers and practitioners in software engineering, as well as applied optimization and control theory. Keywords:Software Quality;Control;Software Cybernetics

Web services provide systems with great flexibility and easier maintenance which result in better ways to communicate and distribute applications. There are good procedures in place for the design, development, and management of Web services; however, there are areas in which Web service adaptation is required. To preserve the loosely coupled approach of Web services, service adaptations should be implemented appropriately. Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solutions includes current research on the area of Web service adaptation while embarking upon the different aspects related to Web services. This collection provides an overview of existing solutions for service adaption in different development scopes as well as covers a wide variety of challenges which emerge. It aims to keep industry professionals as well as academic researchers up to date with the latest research results.

A concise and practical introduction to the foundations and engineering principles of self-adaptation Though it has recently gained significant momentum, the topic of self-adaptation remains largely under-addressed in academic and technical literature. This book

changes that. Using a systematic and holistic approach, An Introduction to Self-adaptive Systems: A Contemporary Software Engineering Perspective provides readers with an accessible set of basic principles, engineering foundations, and applications of self-adaptation in software-intensive systems. It places self-adaptation in the context of techniques like uncertainty management, feedback control, online reasoning, and machine learning while acknowledging the growing consensus in the software engineering community that self-adaptation will be a crucial enabling feature in tackling the challenges of new, emerging, and future systems. The author combines cutting-edge technical research with basic principles and real-world insights to create a practical and strategically effective guide to self-adaptation. He includes features such as: An analysis of the foundational engineering principles and applications of self-adaptation in different domains, including the Internet-of-Things, cloud computing, and cyber-physical systems End-of-chapter exercises at four different levels of complexity and difficulty An accompanying author-hosted website with slides, selected exercises and solutions, models, and code Perfect for researchers, students, teachers, industry leaders, and practitioners in fields that directly or peripherally involve software engineering, as well as those in academia involved in a class on self-adaptivity, this book belongs on the shelves of anyone with an interest in the future of software and its engineering.

First published in 1998. Routledge is an imprint of Taylor & Francis, an informa company.

Although the self-adaptability of systems has been studied in a wide range of disciplines, from biology to robotics, only recently has the software engineering community recognized its key role in enabling the development of self-adaptive systems that are able to adapt to internal faults, changing requirements, and evolving environments. The 15 carefully reviewed papers included in this state-of-the-art survey were presented at the International Seminar on "Software Engineering for Self-Adaptive Systems", held in Dagstuhl Castle, Germany, in October 2010. Continuing the course of the first book of the series on "Software Engineering for Self-Adaptive Systems" the collection of papers in this second volume comprises a research roadmap accompanied by four elaborating working group papers. Next there are two parts - with three papers each - entitled "Requirements and Policies" and "Design Issues"; part four of the book contains four papers covering a wide range of "Applications".

Cognitive networks can be crucial for the evolution of future communication systems; however, current trends have indicated major movement in other relevant fields towards the integration of different techniques for the realization of self-aware and self-adaptive communication systems. Evolution of Cognitive Networks and Self-Adaptive Communication Systems overviews innovative technologies combined for the formation of self-aware, self-adaptive, and self-organizing networks. By aiming to inform the research community and the related industry of solutions for cognitive networks, this book is essential for researchers, instructors, and professionals interested in clarifying the latest trends resulting in a unified realization for cognitive networking and communication systems.

This handbook provides a unique and in-depth survey of the current state-of-the-art in software engineering, covering its major topics, the conceptual genealogy of each subfield, and discussing future research directions. Subjects include foundational areas of software engineering (e.g. software processes, requirements engineering, software architecture, software testing, formal methods, software maintenance) as well as emerging areas (e.g., self-adaptive systems, software engineering in the cloud, coordination technology). Each chapter includes an introduction to central concepts and principles, a guided tour of seminal papers and key contributions, and promising future research directions. The authors of the individual chapters are all acknowledged experts in their field and include many who have pioneered the techniques and technologies discussed. Readers will find an authoritative and concise review of each subject, and will also learn how software engineering technologies have evolved and are likely to develop in the years to come. This book will be especially useful for researchers who are new to software engineering, and for practitioners seeking to enhance their skills and knowledge.

The 18 revised full papers presented in this book together with an introductory survey were carefully reviewed and constitute the documentation of the Second International Workshop on Self-adaptive Software, IWSAS 2001, held in Balatonfüred, Hungary in May 2001. Self-adaptive software evaluates its own behavior and changes it when the evaluation indicates that the software does not accomplish what it is intended to do or when better functionality or better performance is possible. The self-adaptive approach in software engineering builds on well known dynamic features familiar to Lisp or Java programmes and aims at improving the robustness of software systems by gradually adding new features of self-adaption or autonomy.

Traditionally, research on model-driven engineering (MDE) has mainly focused on the use of models at the design, implementation, and verification stages of development. This work has produced relatively mature techniques and tools that are currently being used in industry and academia. However, software models also have the potential to be used at runtime, to monitor and verify particular aspects of runtime behavior, and to implement self-* capabilities (e.g., adaptation technologies used in self-healing, self-managing, self-optimizing systems). A key benefit of using models at runtime is that they can provide a richer semantic base for runtime decision-making related to runtime system concerns associated with autonomic and adaptive systems. This book is one of the outcomes of the Dagstuhl Seminar 11481 on models@run.time held in November/December 2011, discussing foundations, techniques, mechanisms, state of the art, research challenges, and applications for the use of runtime models. The book comprises four research roadmaps, written by the original participants of the Dagstuhl Seminar over the course of two years following the seminar, and seven research papers from experts in the area. The roadmap papers provide insights to key features of the use of runtime models and identify the following research challenges: the need for a reference architecture, uncertainty tackled by runtime models, mechanisms for leveraging runtime models for self-adaptive software, and the use of models at runtime to address assurance for self-adaptive systems.

This text is written with a business school orientation, stressing the how to and heavily employing CASE technology throughout. The courses for which this text is appropriate include software engineering, advanced systems analysis, advanced topics in information systems, and IS project development. Software engineer should be familiar with alternatives, trade-offs and pitfalls of methodologies, technologies, domains, project life cycles, techniques, tools CASE environments, methods for user involvement in application development, software, design, trade-offs for the public domain and project personnel skills. This book discusses much of what should be the ideal software engineer's project related knowledge in order to facilitate and speed the process of novices becoming experts. The goal of this book is to discuss project planning, project life cycles, methodologies, technologies, techniques, tools, languages, testing, ancillary technologies (e.g. database) and CASE. For each topic, alternatives, benefits and disadvantages are discussed.

This book provides formal and informal definitions and taxonomies for self-aware computing systems, and explains how self-aware computing relates to many existing subfields of computer science, especially software engineering. It describes architectures and

algorithms for self-aware systems as well as the benefits and pitfalls of self-awareness, and reviews much of the latest relevant research across a wide array of disciplines, including open research challenges. The chapters of this book are organized into five parts: Introduction, System Architectures, Methods and Algorithms, Applications and Case Studies, and Outlook. Part I offers an introduction that defines self-aware computing systems from multiple perspectives, and establishes a formal definition, a taxonomy and a set of reference scenarios that help to unify the remaining chapters. Next, Part II explores architectures for self-aware computing systems, such as generic concepts and notations that allow a wide range of self-aware system architectures to be described and compared with both isolated and interacting systems. It also reviews the current state of reference architectures, architectural frameworks, and languages for self-aware systems. Part III focuses on methods and algorithms for self-aware computing systems by addressing issues pertaining to system design, like modeling, synthesis and verification. It also examines topics such as adaptation, benchmarks and metrics. Part IV then presents applications and case studies in various domains including cloud computing, data centers, cyber-physical systems, and the degree to which self-aware computing approaches have been adopted within those domains. Lastly, Part V surveys open challenges and future research directions for self-aware computing systems. It can be used as a handbook for professionals and researchers working in areas related to self-aware computing, and can also serve as an advanced textbook for lecturers and postgraduate students studying subjects like advanced software engineering, autonomic computing, self-adaptive systems, and data-center resource management. Each chapter is largely self-contained, and offers plenty of references for anyone wishing to pursue the topic more deeply.

This book discusses the problems and challenges in the interdisciplinary research field of self-adaptive software systems. Modern society is increasingly filled with software-intensive systems, which are required to operate in more and more dynamic and uncertain environments. These systems must monitor and control their environment while adapting to meet the requirements at runtime. This book provides promising approaches and research methods in software engineering, system engineering, and related fields to address the challenges in engineering the next-generation adaptive software systems. The contents of the book range from design and engineering principles (Chap. 1) to control–theoretic solutions (Chap. 2) and bidirectional transformations (Chap. 3), which can be seen as promising ways to implement the functional requirements of self-adaptive systems. Important quality requirements are also dealt with by these approaches: parallel adaptation for performance (Chap. 4), self-adaptive authorization infrastructure for security (Chap. 5), and self-adaptive risk assessment for self-protection (Chap. 6). Finally, Chap. 7 provides a concrete self-adaptive robotics operating system as a testbed for self-adaptive systems. The book grew out of a series of the Shonan Meetings on this ambitious topic held in 2012, 2013, and 2015. The authors were active participants in the meetings and have brought in interesting points of view. After several years of reflection, they now have been able to crystalize the ideas contained herein and collaboratively pave the way for solving some aspects of the research problems. As a result, the book stands as a milestone to initiate further progress in this promising interdisciplinary research field.

There is a strong synergy between the concepts of evolution and adaptation in software engineering: software adaptation refers to both the current software being adapted and to the evolution process that leads to the new adapted software. Evolution changes for the purpose of adaptation are usually made at development or compile time, and are meant to handle predictable situations in the form of software change requests. On the other hand, software may also change and adapt itself based on the changes in its environment. Such adaptive changes are usually dynamic, and are suitable for dealing with unpredictable or temporary changes in the software's operating environment. A promising solution for software adaptation is to develop self-adaptive software systems that can manage changes dynamically at runtime in a rapid and reliable way. One of the main advantages of self-adaptive software is its ability to manage the complexity that stems from highly dynamic and nondeterministic operating environments. If a self-adaptive software system has been engineered and used properly, it can greatly improve the cost-effectiveness of software change through its lifespan. However, in practice, many of the existing approaches towards self-adaptive software are rather expensive and may increase the overall system complexity, as well as subsequent future maintenance costs. This means that in many cases, self-adaptive software is not a good solution, because its development and maintenance costs are not paid off. The situation is even worse in the case of making current (legacy) systems adaptive. There are several factors that have an impact on the cost-effectiveness and usability of self-adaptive software; however the main objective of this thesis is to make a software system adaptive in a cost-effective way, while keeping the target adaptive software generic, usable, and evolvable, so as to support future changes.

This book discusses how model-based approaches can improve the daily practice of software professionals. This is known as Model-Driven Software Engineering (MDSE) or, simply, Model-Driven Engineering (MDE). MDSE practices have proved to increase efficiency and effectiveness in software development, as demonstrated by various quantitative and qualitative studies. MDSE adoption in the software industry is foreseen to grow exponentially in the near future, e.g., due to the convergence of software development and business analysis. The aim of this book is to provide you with an agile and flexible tool to introduce you to the MDSE world, thus allowing you to quickly understand its basic principles and techniques and to choose the right set of MDSE instruments for your needs so that you can start to benefit from MDSE right away. The book is organized into two main parts. The first part discusses the foundations of MDSE in terms of basic concepts (i.e., models and transformations), driving principles, application scenarios, and current standards, like the well-known MDA initiative proposed by OMG (Object Management Group) as well as the practices on how to integrate MDSE in existing development processes. The second part deals with the technical aspects of MDSE, spanning from the basics on when and how to build a domain-specific modeling language, to the description of Model-to-Text and Model-to-Model transformations, and the tools that support the management of MDSE projects. The second edition of the book features: a set of completely new topics, including: full example of the creation of a new modeling language (IFML), discussion of modeling issues and approaches in specific domains, like business process modeling, user interaction modeling, and enterprise architecture complete revision of examples, figures, and text, for improving readability, understandability, and coherence better formulation of definitions, dependencies between concepts and ideas addition of a complete index of book content In addition to the contents of the book, more resources are provided on the book's website http://www.mdse-book.com, including the examples presented in the book.

The carefully reviewed papers in this state-of-the-art survey describe a wide range of approaches coming from different strands of software engineering, and look forward to future challenges facing this ever-resurgent and exacting field of research.

The increasing complexity of systems and the growing uncertainty in their operational environments have created a critical need to develop systems able to improve their operation, adapt to change, and recover from failures autonomously. This situation has led

to recent advances in self-adaptive systems able to reconfigure their structure and modify their behavior at run-time to adapt to environmental changes. Despite these advances, one key aspect of self-adaptive systems that remains to be tackled in depth is "assurances": the provision of evidence that the system satisfies its stated functional and non-functional requirements during its operation in the presence of self-adaptation. This book is one of the outcomes of the ESEC/FSE 2011 Workshop on Assurances for Self-Adaptive Systems (ASAS), held in Szeged, Hungary, in September 2011. It contains extended versions of some of the papers presented during the workshop, as well as invited papers from recognized experts. The 12 refereed papers were thoroughly reviewed and selected. The book consists of four parts: formal verification, models and middleware, failure prediction, and assurance techniques.

As the complexity of today’s networked computer systems grows, they become increasingly difficult to understand, predict, and control. Addressing these challenges requires new approaches to building these systems. Adaptive, Dynamic, and Resilient Systems supplies readers with various perspectives of the critical infrastructure that systems of networked computers rely on. It introduces the key issues, describes their interrelationships, and presents new research in support of these areas. The book presents the insights of a different group of international experts in each chapter. Reporting on recent developments in adaptive systems, it begins with a survey of application fields. It explains the requirements of such fields in terms of adaptation and resilience. It also provides some abstract relationship graphs that illustrate the key attributes of distributed systems to supply you with a better understanding of these factors and their dependencies. The text examines resilient adaptive systems from the perspectives of mobile, infrastructure, and enterprise systems and protecting critical infrastructure. It details various approaches for building adaptive, dynamic, and resilient systems—including agile, grid, and autonomic computing; multi-agent-based and biologically inspired approaches; and self-organizing systems. The book includes many stories of successful applications that illustrate a diversified range of cutting-edge approaches. It concludes by covering related topics and techniques that can help to boost adaptation and resilience in your systems.

Software Engineering for Self-Adaptive Systems III. AssurancesInternational Seminar, Dagstuhl Castle, Germany, December 15-19, 2013, Revised Selected and Invited PapersSpringer

The book intends to cover various problematic aspects of emerging smart computing and self-adapting technologies comprising of machine learning, artificial intelligence, deep learning, robotics, cloud computing, fog computing, data mining algorithms, including emerging intelligent and smart applications related to these research areas. Further coverage includes implementation of self-adaptation architecture for smart devices, self-adaptive models for smart cities and self-driven cars, decentralized self-adaptive computing at the edge networks, energy-aware AI-based systems, M2M networks, sensors, data analytics, algorithms and tools for engineering self-adaptive systems, and so forth. Acts as guide to Self-healing and Self-adaptation based fully automatic future technologies Discusses about Smart Computational abilities and self-adaptive systems Illustrates tools and techniques for data management and explains the need to apply, and data integration for improving efficiency of big data Exclusive chapter on the future of self-stabilizing and self-adaptive systems of systems Covers fields such as automation, robotics, medical sciences, biomedical and agricultural sciences, healthcare and so forth This book is aimed researchers and graduate students in machine learning, information technology, and artificial intelligence.

Self Adaptive, Self Organizing Systems, Cloud Computing, Autonomic Computing

Self-adaptive software evaluates its own behavior and changes its behavior when the evaluation indicates that the software does not accomplish what it is intended to do or when better functionality or better performance is possible. The self-adaptive approach in software engineering builds on well-known features like the use of errors and the handling of exceptions in languages like Lisp or Java and aims at improving the robustness of software systems by gradually adding new features of self-adaption and autonomity. This book originates from the First International Workshop on Self-Adaptive Software, IWSAS 2000, held in Oxford, UK in April 2000. The revised full papers presented in the volume together with an introductory survey by the volume editors assess the state of the art in this emerging new field and set the scene for future research and development work.

Software product line engineering has proven to be the methodology for developing a diversity of software products and software intensive systems at lower costs, in shorter time, and with higher quality. In this book, Pohl and his co-authors present a framework for software product line engineering which they have developed based on their academic as well as industrial experience gained in projects over the last eight years. They do not only detail the technical aspect of the development, but also an integrated view of the business, organisation and process aspects are given. In addition, they explicitly point out the key differences of software product line engineering compared to traditional single software system development, as the need for two distinct development processes for domain and application engineering respectively, or the need to define and manage variability.

A comprehensive collection of influential articles from one of IEEE Computer magazine’s most popular columns This book is a compendium of extended and revised publications that have appeared in the “Software Technologies” column of IEEE Computer magazine, which covers key topics in software engineering such as software development, software correctness and related techniques, cloud computing, self-managing software and self-aware systems. Emerging properties of software technology are also discussed in this book, which will help refine the developing framework for creating the next generation of software technologies and help readers predict future developments and challenges in the field. Software Technology provides guidance on the challenges of developing software today and points readers to where the best advances are being made. Filled with one insightful article after another, the book serves to inform the conversation about the next wave of software technology advances and applications. In addition, the book: Introduces the software landscape and challenges associated with emerging technologies Covers the life cycle of software products, including concepts, requirements, development, testing, verification, evolution, and security Contains rewritten and updated articles by leaders in the software industry Covers both theoretical and practical topics Informative and thought-provoking throughout, Software Technology is a valuable book for everyone in the software engineering community that will inspire as much as it will teach all who flip through its pages.

"This book provides an estimable global view of the most up-to-date research on the strategies, applications, practice, and implications of complex adaptive systems, to better understand the various critical systems that surround human life. Researchers will find this book an indispensable state-of-art reference"--Provided by publisher.

As software systems become more and more ubiquitous, the issues of dependability become more and more critical. Given that solutions to these issues must be planned at the beginning of the design process, it is appropriate that these issues be addressed at the architectural level. This book is inspired by the ICSE 2002 Workshop on Architecting Dependable Systems; it is devoted to

current topics relevant for improving the state of the art for architecting dependability. Some of the 13 peer-reviewed papers presented were initially presented at the workshop, others were invited in order to achieve competent and complete coverage of all relevant aspects. The papers are organized in topical sections on - architectures for dependability - fault tolerance in software architectures - dependability analysis in software architectures - industrial experience.

A collective autonomic system consists of collaborating autonomic entities which are able to adapt at runtime, adjusting to the state of the environment and incorporating new knowledge into their behavior. These highly dynamic systems are also known as ensembles. To ensure correct behavior of ensembles it is necessary to support their development through appropriate methods and tools which can guarantee that an autonomic system lives up to its intended purpose; this includes respecting important constraints of the environment. This State-of-the-Art Survey addresses the engineering of such systems by presenting the methods, tools and theories developed within the ASCENS project. ASCENS was an integrated project funded in the period 2010-2015 by the 7th Framework Programme (FP7) of the European Commission as part of the Future Emerging Technologies Proactive Initiative (FET Proactive). The 17 contributions included in this book are organized in four parts corresponding to the research areas of the project and their concrete applications: (I) language and verification for self-awareness and self-expression, (II) modeling and theory of self-aware and adaptive systems, (III) engineering techniques for collective autonomic systems, and last but not least, (IV) challenges and feedback provided by the case studies of the project in the areas of swarm robotics, cloud computing and e-mobility.

Software architecture—the conceptual glue that holds every phase of a project together for its many stakeholders—is widely recognized as a critical element in modern software development. Practitioners have increasingly discovered that close attention to a software system’s architecture pays valuable dividends. Without an architecture that is appropriate for the problem being solved, a project will stumble along or, most likely, fail. Even with a superb architecture, if that architecture is not well understood or well communicated the project is unlikely to succeed. Documenting Software Architectures, Second Edition, provides the most complete and current guidance, independent of language or notation, on how to capture an architecture in a commonly understandable form. Drawing on their extensive experience, the authors first help you decide what information to document, and then, with guidelines and examples (in various notations, including UML), show you how to express an architecture so that others can successfully build, use, and maintain a system from it. The book features rules for sound documentation, the goals and strategies of documentation, architectural views and styles, documentation for software interfaces and software behavior, and templates for capturing and organizing information to generate a coherent package. New and improved in this second edition: Coverage of architectural styles such as service-oriented architectures, multi-tier architectures, and data models Guidance for documentation in an Agile development environment Deeper treatment of documentation of rationale, reflecting best industrial practices Improved templates, reflecting years of use and feedback, and more documentation layout options A new, comprehensive example (available online), featuring documentation of a Web-based service-oriented system Reference guides for three important architecture documentation languages: UML, AADL, and SySML

This graduate-level text offers a thorough understanding of the global stability properties essential to designing adaptive systems. Its self-contained, unified presentation includes detailed case studies and numerous problems. 1989 edition.

This book constitutes the refereed proceedings of the 23rd International Conference on Advanced Information Systems Engineering, CAiSE 2011, held in London, UK, in June 2011. The 42 revised full papers and 5 revised short papers presented were carefully reviewed and selected from 320 submissions. In addition the book contains the abstracts of 2 keynote speeches. The contributions are organized in topical sections on requirements; adaptation and evolution; model transformation; conceptual design; domain specific languages; case studies and experiences; mining and matching; business process modelling; validation and quality; and service and management.

The book is about a very active research field in software engineering. In modern society, the fact of the world's high reliance on software requires the system's robustness, i.e., continual availability and satisfactory service quality. This requirement gives rise to the popularity of the research on the self-adaptive software in open environment. There are some academic conferences dedicated to this field. But there is a lack of monographs about the topic. We believe such need is unmet in marketplace. By publishing the book, it can help bridge the gap and bring benefits to readers thereof. Key Features: The topic is well-motivated, interesting and actively studied worldwide The research represents as the state-of-the-art in the field The technical part of the book is rigidly evaluated The theoretical part of the book is sound and proved The organization and presentation of the book will be double-checked by professional scholars

A major challenge for modern software systems is to become more cost-effective, while being versatile, flexible, resilient, energy-efficient, customizable, and configurable when reacting to run-time changes that may occur within the system itself, its environment or requirements. One of the most promising approaches to achieving such properties is to equip the software system with self-adaptation capabilities. Despite recent advances in this area, one key aspect that remains to be tackled in depth is the provision of assurances. Originating from a Dagstuhl seminar held in December 2013, this book constitutes the third volume in the series “Software Engineering for Self-Adaptive Systems”, and looks specifically into the provision of assurances. Opening with an overview chapter on Research Challenges, the book presents 13 further chapters written and carefully reviewed by internationally leading researchers in the field. The book is divided into topical sections on research challenges, evaluation, integration and coordination, and reference architectures and platforms.

Managing Trade-Offs in Adaptable Software Architectures explores the latest research on adapting large complex systems to changing requirements. To be able to adapt a system, engineers must evaluate different quality attributes, including trade-offs to balance functional and quality requirements to maintain a well-functioning system throughout the lifetime of the system. This comprehensive resource brings together research focusing on how to manage trade-offs and architect adaptive systems in different business contexts. It presents state-of-the-art techniques, methodologies, tools, best practices, and guidelines for developing adaptive systems, and offers guidance for future software engineering research and practice. Each contributed chapter considers the practical application of the topic through case studies,

experiments, empirical validation, or systematic comparisons with other approaches already in practice. Topics of interest include, but are not limited to, how to architect a system for adaptability, software architecture for self-adaptive systems, understanding and balancing the trade-offs involved, architectural patterns for self-adaptive systems, how quality attributes are exhibited by the architecture of the system, how to connect the quality of a software architecture to system architecture or other system considerations, and more. Explains software architectural processes and metrics supporting highly adaptive and complex engineering Covers validation, verification, security, and quality assurance in system design Discusses domain-specific software engineering issues for cloud-based, mobile, context-sensitive, cyber-physical, ultra-large-scale/internet-scale systems, mash-up, and autonomic systems Includes practical case studies of complex, adaptive, and context-critical systems

Professionals in the interdisciplinary field of computer science focus on the design, operation, and maintenance of computational systems and software. Methodologies and tools of engineering are utilized alongside computer applications to develop efficient and precise information databases. Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications is a comprehensive reference source for the latest scholarly material on trends, techniques, and uses of various technology applications and examines the benefits and challenges of these computational developments. Highlighting a range of pertinent topics such as utility computing, computer security, and information systems applications, this multi-volume book is ideally designed for academicians, researchers, students, web designers, software developers, and practitioners interested in computer systems and software engineering.