Designing Software Product Lines With Uml From Use Cases To Pattern Based Software Architectures

The six volumes LNCS 11619-11624 constitute the refereed proceedings of the 19th International Conference on Computational Science and Its Applications, ICCSA 2019, held in Saint Petersburg, Russia, in July 2019. The 64 full papers, 10 short papers and 259 workshop papers presented were carefully reviewed and selected form numerous submissions. The 64 full papers are organized in the following five general tracks: computational methods, algorithms and scientific applications; high performance computing and networks; geometric modeling, graphics and visualization; advanced and emerging applications; and information systems and technologies. The 259 workshop papers were presented at 33 workshops in various areas of computational sciences, ranging from computational science technologies to specific areas of computational sciences, such as software engineering, security, artificial intelligence and blockchain technologies.

"This book displays how to effectively map and respond to the real-world challenges and purposes which software must solve, covering domains such as mechatronic, embedded and high risk systems, where failure could cost human lives"--Provided by publisher.

This tutorial reference takes the reader from use cases to complete architectures for real-time embedded systems using SysML, UML, and MARTE and shows how to apply the COMET/RTE design method to real-world problems. The author covers key topics such as architectural patterns for distributed and hierarchical real-time control and other real-time software architectures, performance analysis of real-time designs using real-time scheduling, and timing analysis on single and multiple processor systems. Complete case studies illustrating design issues include a light rail control system, a microwave oven control system, and an automated highway toll system. Organized as an introduction followed by several self-contained chapters, the book is perfect for experienced software engineers wanting a quick reference at each stage of the analysis, design, and development of large-scale real-time embedded systems, as well as for advanced undergraduate or graduate courses in software engineering, computer engineering, and software design.

Software product lines are emerging as an important new paradigm for so- ware development. Product lines are enabling organizations to achieve impressive time-to-market gains and cost reductions. In 1997, we at the Software Engine- ing Institute (SEI) launched a Product Line Practice Initiative. Our vision was that product line development would be a low-risk, high-return proposition for the entire software engineering community. It was our hope from the beginning that there would eventually be su?cient interest to hold a conference. The First Software Product Line Conference (SPLC1) was the realization of that hope. Since SPLC1, we have seen a growing interest in software product lines. Companies are launching their own software product line initiatives, product line technical and business practices are maturing, product line tool vendors are emerging, and books on product lines are being published. Motivated by the enthusiastic response to SPLC1 and the increasing number of software product lines and product line researchers and practitioners, the SEI is proud to sponsor this second conference dedicated to software product lines. We were grati?ed by the submissions to SPLC2 from all parts of the globe, from government and commercial organizations. From these submissions we were able to assemble a rich and varied conference program with unique opportunities for software product line novices, experts, and those in between. This collection represents the papers selected from that response and includes research and experience reports.

10th International Joint Conference, ICSOFT 2015, Colmar, France, July 20-22, 2015, Revised Selected Papers

14th International Conference, SPLC 2010, Jeju Island, South Korea, September 13-17, 2010. Proceedings

The Best Industrial Practice in Product Line Engineering

Systems and Software Variability Management

UML, Use Cases, Patterns, and Software Architectures

Software Product-line Engineering

7th International Conference, PROFES 2006, Amsterdam, The Netherlands, June 12-14, 2006, Proceedings

This volume constitutes the refereed proceedings of the 14th International Software Product Line Conference, SPLC 2010, held on Jeju Island, South Korea, in September 2010.

The concept of a data lake is less than 10 years old, but they are already hugely implemented within large companies. Their goal is to efficiently deal with ever-growing volumes of heterogeneous data, while also facing various sophisticated user needs. However, defining and building a data lake is still a challenge, as no consensus has been reached so far. Data Lakes presents recent outcomes and trends in the field of data repositories. The main topics discussed are the data-driven architecture of a data lake; the management of metadata – supplying key information about the stored data, master data and reference data; the roles of linked data and fog computing in a data lake ecosystem; and how gravity principles apply in the context of data lakes. A variety of case studies are also presented, thus providing the reader with practical examples of data lake management.

Software product lines (SPLs) allow managing the variability that arises in families of related software models due to varying customer needs. While designing changes to them, engineers need to consider many alternative SPL designs. However, without complete information about the desired quality requirements of the final SPL, engineers face uncertainty about how to make the appropriate design choices. Existing formalisms and techniques are not well suited to modelling and reasoning about the two dimensional space defined by variability and design choices. We propose an approach for modelling design uncertainty in SPLs and for analyzing and understanding the impact of design choices in the quality of SPLs, expressed as properties. We formally define Software Product Lines with Design Choices (SPLDCs) and outline a procedure for analyzing them and providing appropriate feedback to engineers, based on the partial order of SPLDC property categories. We illustrate the applicability of our approach using a fully worked out example, that shows the kind of nuanced feedback necessary for meaningful analysis of SPLs in the presence of design choices. To evaluate the scalability of our approach we use our approach over many SPLDCs and record runtimes.

The success of product line engineering techniques in the last 15 years has popularized the use of software variability as a key modeling approach for describing the commonality and variability of systems at all stages of the software lifecycle. Software product lines enable a family of products to share a common core platform, while allowing for product specific functionality being built on top of the platform. Many companies have exploited the concept of software product lines to increase the resources that focus on highly differentiating functionality and thus improve their competitiveness with higher quality and reusable products and decreasing the time-to-market condition. Many books on product line engineering either introduce specific product line techniques or include brief summaries of industrial cases. From these sources, it is difficult to gain a comprehensive understanding of the various dimensions and aspects of software variability. Here the editors address this gap by providing a comprehensive reference on the notion of variability modeling in the context of software product line engineering, presenting an overview of the techniques proposed for variability modeling and giving a detailed perspective on software variability management. Their book is organized in four main parts, which guide the reader through the various aspects and dimensions of software variability. Part 1 which is mostly written by the editors themselves introduces the major topics related to software variability modeling, thus providing a multi-faceted view of both technological and management issues. Next, part 2 of the book comprises four separate chapters dedicated to research and commercial tools. Part 3 then continues with the most practical viewpoint of the book presenting three different industry cases on how variability is managed in real industry projects. Finally, part 4 concludes the book and encompasses six different chapters on emerging research topics in software variability like e.g. service-oriented or dynamic software product lines, or variability and aspect orientation. Each chapter briefly summarizes "What you will learn in this chapter", so both expert and novice readers can easily locate the topics dealt with. Overall, the book captures the current state of the art and best practices, and indicates important open research challenges as well as possible pitfalls. Thus it serves as a reference for researchers and practitioners in software variability management, allowing them to develop the next set of solutions, techniques and methods in this complicated and yet fascinating field of software engineering.

Data Lakes

Aspect-Oriented, Model-Driven Software Product Lines

Software Product Lines

Software Modeling and Design

Software Engineering Research, Management and Applications

Design and Use of Software Architectures

Model Driven Engineering Languages and Systems

This book constitutes the refereed proceedings of the 7th International Conference on Product-Focused Software Process Improvement, PROFES 2006, held in Amsterdam, June 2006. The volume presents 26 revised full papers and 12 revised short papers together with 6 reports on workshops and tutorials. The papers constitute a balanced mix of academic and industrial aspects, organized in topical sections on decision support, embedded software and system development, measurement, process improvement, and more.

Mobile devices are rapidly developing into the primary technology for users to work, socialize, and play in a variety of settings and contexts. Their pervasiveness has provided researchers with the means to investigate innovative solutions to ever more complex user demands. Tools for Mobile Multimedia Programming and Development investigates the use of mobile platforms for research projects, focusing on the development, testing, and evaluation of prototypes rather than final products, which enables researchers to better understand the needs of users through image processing, object recognition, sensor integration, and user interactions. This book benefits researchers and professionals in multiple disciplines who utilize such techniques in the creation of prototypes for mobile devices and applications. This book is part of the Advances in Wireless Technologies and Telecommunication series collection.

This book covers research into the most important practices in product line organization. Contributors offer experience-based knowledge on the domain and application engineering, the modeling and management of variability, and the design and use of tools to support the management of product line-related knowledge.

This book constitutes the refereed proceedings of the 9th International Conference on Software Reuse, ICSR 2006, held in Torino, Italy, in June 2006. The book presents 27 revised full papers and 13 revised short papers, carefully reviewed and selected from numerous submissions. The Coverage includes COTS selection and integration; product lines, domain analysis, and variability; reengineering maintenance; programming languages and retrieval; aspect-oriented software development; approaches and models; and components.

Model-Driven Domain Analysis and Software Development: Architectures and Functions

Software Technologies

9th International Conference on Software Reuse, ICSR 2006, Torino, Italy, June 12-15, 2006, Proceedings

Fundamental Approaches to Software Engineering

Advanced Topic

Practices and Patterns: Practices and Patterns

Concepts, Tools and Experiences

Many approaches to creating Software Product Lines have emerged that are based on Model-Driven Engineering. This book introduces both Software Product Lines and Model-Driven Engineering, which have separate success stories in industry, and focuses on the practical combination of them. It describes the challenges and benefits of merging these two software development trends and provides the reader with a novel approach and practical mechanisms to improve software development productivity. The book is aimed at engineers and students who wish to understand and apply software product lines and model-driven engineering in their activities today. The concepts and methods are illustrated with two product line examples: the classic smart-home systems and a collection manager information system.

This book constitutes the refereed proceedings of the 6th International Symposium on Search-Based Software Engineering, SSBSE 2014, held in Fortaleza, Brazil. The 14 revised full papers presented together with 2 keynote addresses, 1 invited talk, 1 short paper, 3 papers of the graduate track, and 4 challenge track papers were carefully reviewed and selected from 51 submissions. Search Based Software Engineering (SBSE) studies the application of meta-heuristic optimization techniques to various software engineering problems, ranging from requirements engineering to software testing and maintenance.

This book constitutes the thoroughly refereed proceedings of the 10th International Joint Conference on Software Technologies, ICSOFT 2015, held in Colmar, France, in July 2015. The 23 revised full papers presented were carefully reviewed and selected from 117 submissions. The papers are organized around the following conference tracks: enterprise software technologies; software project management; software engineering methods and techniques; distributed and mobile software systems.

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.

Computational Science and Its Applications – ICCSA 2019

Software Architecture

19th International Conference, Saint Petersburg, Russia, July 1–4, 2019, Proceedings, Part V

Foundations, Principles and Techniques

Model-Driven and Software Product Line Engineering

Second International Conference, SPLC 2, San Diego, CA, USA, August 19-22, 2002. Proceedings

High Confidence Software Reuse in Large Systems

This book constitutes the proceedings of the 16th International Conference on Fundamental Approaches to Software Engineering, FASE 2013, held as part of the European Joint Conference on Theory and Practice of Software, ETAPS 2013, which took place in Rome, Italy, in March 2013. The 25 papers presented in this volume were carefully reviewed and selected from 112 s are organized in topical sections named: model-driven engineering; verification and validation; software comprehension; analysis tools; model-driven engineering: applications; model transformations; and testing.

A practical guide to designing and implementing software architectures.

The authors outline a systematic method for rapid software production through the family-oriented abstraction, specification, and translation (FAST) process. FAST uses practical domain engineering to decrease the time and effort necessary to develop, deliver, and maintain software. Any software development projects using C, C++, or Java can incorporate the FAST model. Th contains a FAST PASTA browser and a simulator for a floating weather station. Annotation copyrighted by Book News, Inc., Portland, OR

Domain engineering is a set of activities intended to develop, maintain, and manage the creation and evolution of an area of knowledge suitable for processing by a range of software systems. It is of considerable practical significance, as it provides methods and techniques that help reduce time-to-market, development costs, and project risks on one hand, and helps improve sy performance on a consistent basis on the other. In this book, the editors present a collection of invited chapters from various fields related to domain engineering. The individual chapters present state-of-the-art research and are organized in three parts. The first part focuses on results that deal with domain engineering in software product lines. The second part describes how d languages are used to support the construction and deployment of domains. Finally, the third part presents contributions dealing with domain engineering within the field of conceptual modeling. All chapters utilize a similar terminology, which will help readers to understand and relate to the chapters content. The book will be especially rewarding for researchers and students of s engineering methodologies in general and of domain engineering and its related fields in particular, as it contains the most comprehensive and up-to-date information on this topic.

Experience and Research Directions

Product Lines, Languages, and Conceptual Models

Adopting and Evolving a Product-line Approach

Tools for Mobile Multimedia Programming and Development

Product Lines for Digital Information Products

Architectures and Functions

Software Product Line Engineering

Software product lines provide a systematic means of managing variability in a suite of products. They have many benefits but there are three major barriers that can prevent them from reaching their full potential. First, there is the challenge of scale: a large number of variants may exist in a product line context and the number of interrelationships and dependencies can rise exponentially. Second, variations tend to be systemic by nature in that they affect the whole architecture of the software product line. Third, software product lines often serve different business contexts, each with its own intricacies and complexities. The AMPLE (http://www.ample-project.net/) approach tackles these three challenges by combining advances in aspect-oriented software development and model-driven engineering. The full suite of methods and tools that constitute this approach are discussed in detail in this edited volume and illustrated using three real-world industrial case studies.

Software product lines are emerging as a critical new paradigm for software development. Product lines are enabling organizations to achieve impressive time-to-market gains and cost reductions. With the increasing number of product lines and product-line researchers and practitioners, the time is right for a comprehensive examination of the issues surrounding the software product line approach. The Software Engineering Institute at Carnegie Mellon University is proud to sponsor the first conference on this important subject. This book comprises the proceedings of the First Software Product Line Conference (SPLC1), held August 28-31, 2000, in Denver, Colorado, USA. The twenty-seven papers of the conference technical program present research results and experience reports that cover all aspects of software product lines. Topics include business issues, enabling technologies, organizational issues, and life-cycle issues. Emphasis is placed on experiences in the development and fielding of product lines of complex systems, especially those that expose problems in the design, development, or evolution of software product lines. The book will be essential reading for researchers and practitioners alike.

This open access book includes contributions by leading researchers and industry thought leaders on various topics related to the essence of software engineering and their application in industrial projects. It offers a broad overview of research findings dealing with current practical software engineering issues and also pointers to potential future developments. Celebrating the 20th anniversary of adesso AG, adesso gathered some of the pioneers of software engineering including Manfred Broy, Ivar Jacobson and Carlo Ghezzi at a special symposium, where they presented their thoughts about latest software engineering research and which are part of this book. This way it offers readers a concise overview of the essence of software engineering, providing valuable insights into the latest methodological research findings and adesso's experience applying these results in real-world projects.

MODELS2008wasthe11theditionoftheseriesofconferencesonModel-Driven Engineering Languages and Systems. The conference was held in Toulouse, France, during the week of September 28 to October 3, 2008. The local arran- ments were provided by the Institut de Recherche en Informatique de Toulouse (IRIT). The conference program included three keynote presentations, technical - per presentations, two panels, and several workshops and tutorials. The invited keynote speakers were Don Batory (University of Texas, USA), Je? Kramer (Imperial College London, UK), and Patrick Rauhut (Airbus, Germany). Thisvolumecontainsthe?nalversionsofthepapersacceptedforpresentation attheconference.Thepaperscoverawiderangeoftopicsfromthe?eldincluding model transformation, model management, domain-speci?c modeling, modeling language semantics, model analysis, and applications. We received a record number of 271 full paper submissions from 40 di?erent countries. Of these, 43 papers were submitted by authors from more than one country. The top three countries submitting papers were France (40), Germany (38), and Canada (24). A total of 58 papers were accepted for inclusion in the proceedings. The acceptance rate was therefore 21%, which is somewhat lower than those of the previous MODELS conferences. At least three Program Committee or Expert Reviewer Panel members - viewed each paper. Reviewing wasthorough,and most authors received detailed comments on their submissions. Con?icts of interest were taken very seriously. No-oneparticipatedinany wayin the decisionprocessofanypaper wherea c- ?ict of interest was identi?ed. In particular, PC members who submitted papers did not have access to information concerning the reviews of their papers.

11th International Conference, MoDELS 2008, Toulouse, France, September 28 - October 3, 2008, Proceedings

Second International Conference, ECSA 2008 Paphos, Cyprus, September 29-October 1, 2008 Proceedings

Feature-Oriented Software Product Lines

Applied Software Product Line Engineering

Designing Software Product Lines with UML

Software Product Lines in Action

The Essence of Software Engineering

Over the last decade, software product line engineering (SPLE) has emerged as one of the most promising software development paradigms for increasing productivity in IT-related industries. Detailing the various aspects of SPLE implementation in different domains, Applied Software Product Line Engineering documents best practices with regard to system development. Expert contributors from academia and industry come together and focus on core asset development, product development, and management, addressing the process, technical, and organizational issues needed to meet the growing demand for information. They detail the adoption and diffusion of SPLE as a primary software development paradigm and also address technical and managerial issues in software product line engineering. Providing an authoritative perspective of the latest research and practice in SLPE, the text: Presents in-depth discussions and many industry / case studies Covers applications in various domains including automotive, business process management, and defense Organized according to the organizational, process, and technical aspects of software product lines within an organization Provides the expertise of a distinguished panel of global contributors Ever-increasing global competition coupled with a fragile world economy means that the pressure is on for software engineers and software process improvement professionals to find ways to meet the needs of expanding markets—with greater efficiency and effectiveness. This book arms readers with the insight needed to harness the power of SPLE to increase productivity, reduce time to market, and to handle the growing diversity in the quickly evolving global marketplace.

This book constitutes the refereed proceedings of the 9th International Software Product Line Conference, SPLC 2005, held in Rennes, France in September 2005, emanating from the merger of the former events SPLC (Software Product Line Conference started 2000 in the USA) and PFE (Product Family Engineering started 1996 in Europe). The 17 revised full technical papers presented together with 3 short research papers and 2 keynote talks were carefully reviewed and selected for inclusion in the book. The papers are organized in topical sections on keynotes, feature modelling, re-engineering, short papers, strategies, panels, validation, scoping and architecture, and product derivation.

While standardization has empowered the software industry to substantially scale software development and to provide affordable software to a broad market, it often does not address smaller market segments, nor the needs and wishes of individual customers. Software product lines reconcile mass production and standardization with mass customization in software engineering. Ideally, based on a set of reusable parts, a software manufacturer can generate a software product based on the requirements of its customer. The concept of features is central to achieving this level of automation, because features bridge the gap between the requirements the customer has and the functionality a product provides. Thus features are a central concept in all phases of product-line development. The authors take a developer's viewpoint, focus on the development, maintenance, and implementation of product-line variability, and especially concentrate on automated product derivation based on a user's feature selection. The book consists of three parts. Part I provides a general introduction to feature-oriented software product lines, describing the product-line approach and introducing the product-line development process with its two elements of domain and application engineering. The pivotal part II covers a wide variety of implementation techniques including design patterns, frameworks, components, feature-oriented programming, and aspect-oriented programming, as well as tool-based approaches including preprocessors, build systems, version-control systems, and virtual separation of concerns. Finally, part III is devoted to advanced topics related to feature-oriented product lines like refactoring, feature interaction, and analysis tools specific to product lines. In addition, an appendix lists various helpful tools for software product-line development, along with a description of how they relate to the topics covered in this book. To tie the book together, the authors use two running examples that are well documented in the product-line literature: data management for embedded systems, and variations of graph data structures. They start every chapter by explicitly stating the respective learning goals and finish it with a set of exercises; additional teaching material is also available online. All these features make the book ideally suited for teaching – both for academic classes and for professionals interested in self-study.

This book constitutes the refereed proceedings of the Second European Conference on Software Architecture, ECSA 2008, held in Paphos, Cyprus, in September/October 2008. The 12 revised full papers presented together with 2 keynote abstracts, 4 experience papers, 7 emerging research papers, and 12 research challenge poster papers were carefully reviewed and selected from 83 submissions. The papers focus on formalisms, technologies, and processes for describing, verifying, validating, transforming, building, and evolving software systems. Topics include architecture modeling, architecture description languages, architectural aspects, architecture analysis, transformation and synthesis, architecture evolution, quality attributes, model-driven engineering, built-in testing and architecture-based support for component-based and service-oriented systems.

Research Issues in Engineering and Management

A Family-based Software Development Process

The AMPLE Way

Software Product Line

Modelling and Reasoning with Software Product Lines with Design Choices

9th International Conference, SPLC 2005, Rennes, France, September 26-29, 2005, Proceedings

Reuse of Off-the-Shelf Components

This book takes a unique HCI approach to the concept of Software Product Line (SPL) and discusses the peculiarities of human-computer interaction not usually addressed in more traditional approaches. SPL is based on industrial practices for defining a range of software products. SPL design identifies commonalities and differences between the various software versions, modelling and managing the software variability. Recent research has focused on reconciling the different viewpoints of SPL and HCI, and in particular emphasizing the specific variability of HCI and the management of complex SPL models that could benefit from HCI in terms of representation, manipulation and visualization. This edited volume includes research that addresses the SPL for HCI and HCI for SPL. In putting together

This book constitutes the refereed proceedings of the 10th International Conference on Software Reuse, ICSR 2008, held in Beijing, China, in May 2008. The 40 revised full papers presented together with 5 workshop summaries and 5 tutorials were carefully reviewed and selected from numerous submissions. The papers are organized in topical sections on architecture and reuse approaches, high confidence and reuse, component selection and reuse repository, product line, domain models and analysis, service oriented environment, components and services, reuse approaches and frameworks, as well as reuse approaches and methods.

This book covers all you need to know to model and design software applications from use cases to software architectures in UML and shows how to apply the COMET UML-based modeling and design method to real-world problems. The author describes architectural patterns for various architectures, such as broker, discovery, and transaction patterns for service-oriented architectures, and addresses software quality attributes including maintainability, modifiability, testability, traceability, scalability, reusability, performance, availability, and security. Complete case studies illustrate design issues for different software architectures: a banking system for client/server architecture, an online shopping system for service-oriented architecture, an emergency monitoring system for component-based software architecture, and an automated guided vehicle for real-time software architecture. Organized as an introduction followed by several short, self-contained chapters, the book is perfect for senior undergraduate or graduate courses in software engineering and design, and for experienced software engineers wanting a quick reference at each stage of the analysis, design, and development of large-scale software systems.

The 6th ACIS International Conference on Software Engineering, Research, Management and Applications (SERA 2008) was held in Prague in the Czech Republic on August 20 – 22. SERA '08 featured excellent theoretical and practical contributions in the areas of formal methods and tools, requirements engineering, software process models, communication systems and networks, software quality and evaluation, software engineering, networks and mobile computing, parallel/distributed computing, software testing, reuse and metrics, database retrieval, computer security, software architectures and modeling. Our conference officers selected the best 17 papers from those papers accepted for presentation at the conference in order to publish them in this volume. The papers were chosen based on review scores submitted by members or the program committee, and underwent further rounds of rigorous review.

Consolidation of Customized Product Copies into Software Product Lines

16th International Conference, FASE 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013, Proceedings

Product-Focused Software Process Improvement

From Use Cases to Pattern-based Software Architectures

Real-Time Software Design for Embedded Systems

Concepts and Implementation

Domain Engineering

Software product lines represent perhaps the most exciting paradigm shift in software development since the advent of high-level programming languages. Nowhere else in software engineering have we seen such breathtaking improvements in cost, quality, time to market, and developer productivity, often registering in the order-of-magnitude range. Here, the authors combine academic research results with real-world industrial experiences, thus presenting a broad view on product line engineering so that both managers and technical specialists will benefit from exposure to this work. They capture the wealth of knowledge that eight companies have gathered during the introduction of the software product line engineering approach in their daily practice.

"Designing Software Product Lines with UML is well-written, informative, and addresses a very important topic. It is a valuable contribution to the literature in this area, and offers practical guidance for software architects and engineers." --Alan Brown Distinguished Engineer, Rational Software, IBM Software Group "Gomaa"s process and UML extensions allow development teams to focus on feature-oriented development and provide a basis for improving the level of reuse across multiple software development efforts. This book will be valuable to any software development professional who needs to manage across projects and wants to focus on creating software that is consistent, reusable, and modular in nature." --Jeffrey S Hammond Group Marketing Manager, Rational Software, IBM Software Group "This book brings together a good range of concepts for understanding software product lines and provides an organized method for developing product lines using object-oriented techniques with the UML. Once again, Hassan has done an excellent job in balancing the needs of both experienced and novice software engineers." --Robert G. Pettit IV, Ph.D. Adjunct Professor of Software Engineering, George Mason University "This breakthrough book provides a comprehensive step-by-step approach on how to develop software product lines, which is of great strategic benefit to industry. The development of software product lines enables significant reuse of software architectures. Practitioners will benefit from the well-defined PLUS process and rich case studies." --Hurley V. Blankenship II Program Manager, Justice and Public Safety, Science Applications International Corporation "The Product Line UML based Software engineering (PLUS) is leading edge. With the author"s wide experience and deep knowledge, PLUS is well harmonized with architectural and design pattern technologies." --Michael Shin Assistant Professor, Texas Tech University Long a standard practice in traditional manufacturing, the concept of product lines is quickly earning recognition in the software industry. A software product line is a family of systems that shares a common set of core technical assets with preplanned extensions and variations to address the needs of specific customers or market segments. When skillfully implemented, a product line strategy can yield enormous gains in productivity, quality, and time-to-market. Studies indicate that if three or more systems with a degree of common functionality are to be developed, a product-line approach is significantly more cost-effective. To model and design families of systems, the analysis and design concepts for single product systems need to be extended to support product lines. Designing Software Product Lines with UML shows how to employ the latest version of the industry-standard Unified Modeling Language (UML 2.0) to reuse software requirements and architectures rather than starting the development of each new system from scratch. Through real-world case studies, the book illustrates the fundamental concepts and technologies used in the design and implementation of software product lines. This book describes a new UML-based software design method for product lines called PLUS (Product Line UML-based Software engineering). PLUS provides a set of concepts and techniques to extend UML-based design methods and processes for single systems in a new dimension to address software product lines. Using PLUS, the objective is to explicitly model the commonality and variability in a software product line. Hassan Gomaa explores how each of the UML modeling views--use case, static, state machine, and interaction modeling--can be extended to address software product families. He also discusses how software architectural patterns can be used to develop a reusable component-based architecture for a product line and how to express this architecture as a UML platform-independent model that can then be mapped to a platform-specific model. Key topics include: Software product line engineering process, which extends the Unified Development Software Process to address software product lines Use case modeling, including modeling the common and variable functionality of a product line Incorporating feature modeling into UML for modeling common, optional, and alternative product line features Static modeling, including modeling the boundary of the product line and information-intensive entity classes Dynamic modeling, including using interaction modeling to address use-case variability State machines for modeling state-dependent variability Modeling class variability using inheritance and parameterization Software architectural patterns for product lines Component-based distributed design using the new UML 2.0 capability for modeling components, connectors, ports, and provided and required interfaces Detailed case studies giving a step-by-step solution to real-world product line problems Designing Software Product Lines with UML is an invaluable resource for all designers and developers in this growing field. The information, technology, and case studies presented here show how to harness the promise of software product lines and the practicality of the UML to take software design, quality, and efficiency to the next level. An enhanced online index allows readers to quickly and easily search the entire text for specific topics.

The Software Product Line (SPL) is an emerging methodology for developing software products. Currently, there are two hot issues in the SPL: modelling and the analysis of the SPL. Variability modelling techniques have been developed to assist engineers in dealing with the complications of variability management. The principal goal of modelling variability techniques is to configure a successful software product by managing variability in domain-engineering. In other words, a good method for modelling variability is a prerequisite for a successful SPL. On the other hand, analysis of the SPL aids the extraction of useful information from the SPL and provides a control and planning strategy mechanism for engineers or experts. In addition, the analysis of the SPL provides a clear view for users. Moreover, it ensures the accuracy of the SPL. This book presents new techniques for modelling and new methods for SPL analysis.

Software Product Lines: Going Beyond

Human Centered Software Product Lines

6th International Symposium, SSBSE 2014, Fortaleza, Brazil, August 26-29, 2014, Proceedings

Search-Based Software Engineering

10th International Conference on Software Reuse, ICSR 2008, Bejing, China, May 25-29, 2008