Component Based Product Line Engineering With Uml Component Based Development Series

When people should go to the books stores, search commencement by shop, shelf by shelf, it is really problematic. This is why we present the book compilations in this website. It will definitely ease you to see guide **Component based Product Line Engineering With Uml Component Based Development Series** as you such as.

By searching the title, publisher, or authors of guide you truly want, you can discover them rapidly. In the house, workplace, or perhaps in your method can be all best area within net connections. If you strive for to download and install the Component based Product Line Engineering With Uml Component Based Development Series , it is totally simple then, before currently we extend the link to purchase and make bargains to download and install Component based Product Line Engineering With Uml Component Based Development Series appropriately simple!

Software Modeling and Design - Hassan Gomaa 2011-02-21
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.

Rapid Integration of Software Engineering Techniques - Nicolas Guelfi 2006-05-16
This book constitutes the thoroughly refereed post-proceedings of the Second International Workshop on Rapid Integration of Software Engineering Techniques, RISE 2005. The book presents 19 revised full papers together with the abstract of a keynote paper. Among the topics addressed are modelling safety case evolution, practical approaches in model mapping, context-aware service composition, techniques for representing product line core assets for automation, formal development of reactive fault-tolerant systems, and more.
Software Architecture - Ronald Morrison 2008-09-22
The European Conference on Software Architecture (ECSA) is the premier European conference dedicated to the field of software architecture, covering all architectural features of software engineering. It is the follow-up of a successful series of European workshops on software architecture held in the UK in 2004 (Springer LNCS 3047), Italy in 2005 (Springer LNCS 3527), and France in 2006 (Springer LNCS 4344). It evolved into a series of European conferences whose first edition was ECSA 2007, held in Madrid, Spain during September 24–26, 2007 (Springer LNCS 4758). This year's conference was held at the beautiful Coral

Beach Hotel and Resort near Paphos in Cyprus. As with the previous versions of the conference, ECSA 2008 (Springer LNCS 5292) provided an international forum for researchers and practitioners from academia and industry to present innovative research and to discuss a wide range of topics in the area of software architecture. It focused on formalisms, technologies, and processes for describing, verifying, validating, transforming, building, and evolving software systems. Covered topics included architecture modelling, 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. The conference attracted paper submissions from 29 countries (Australia, Belgium, Brazil, Canada, China, Chile, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Lithuania, Luxembourg, Malta, The Netherlands, Norway, Pakistan, Peru, - land, Portugal, Romania, South Africa, Spain, Turkey, the UK, USA, and Venezuela).

Advanced Topics in Database Research - Keng Siau 2004-01-01
This book presents the latest research ideas and topics on how to enhance current database systems, improve information storage, refine existing database models, and develop advanced applications. It provides insights into important developments in the field of database and database management. With emphasis on theoretical issues regarding databases and database management, the book describes the capabilities and features of new technologies and methodologies, and addresses the needs of database researchers and practitioners. *Note: This book is part of a new series entitled Advanced Topics in Database Research ." This book is Volume Three within this series (Vol. III, 2004).

Component-based Software Development - Kung-Kiu Lau 2004
Component-based software development (CBD) is an emerging discipline that promises to take software engineering into a new era. Building on the achievements of object-oriented software construction, CBD aims to deliver software engineering from a cottage industry into an industrial age for Information Technology, wherein software can be assembled from components, in the manner that hardware systems are currently constructed from kits of parts. This volume provides a survey of the current state of CBD, as reflected by activities that have been taking place recently under the banner of CBD, with a view to giving pointers to future trends. The contributions report case studies - self-contained, fixed-term investigations with a finite set of clearly defined objectives and measurable outcomes - on a sample of the myriad aspects of CBD. The book includes chapters dealing with COTS (commercial off-the-shelf) components; methodologies for CBD; compositionality, i.e. how to calculate or predict properties of a composite from those of its constituents; component software testing; and grid computing.

Software Architecture - Ivica Crnkovic 2011-09-15
This book constitutes the refereed proceedings of the 5th European Conference on Software Architecture, ECSA 2011, held in Essen, Germany, in September 2011. The 13 revised full papers presented together with 24 emerging research papers, and 7 research challenge poster papers were carefully reviewed and selected from over 100 submissions. The papers are organized in topical sections on requirements and software architectures; software architecture, components, and compositions; quality attributes and software architectures; software product line architectures; architectural models, patterns and styles; short papers; process and management of architectural decisions; software architecture run-time aspects; ADLs and metamodels; and services and software architectures.

Software Reuse: Methods, Techniques, and Tools - Cristina Gacek 2003-08-01
As a result of the open-source movement there is now a great deal of reusable software available in the public domain. This offers significant functionality that commercial software vendors can use in their software projects. Open-source approaches to software development have illustrated that complex, mission critical software can be developed by distributed teams of developers sharing a common goal.

Commercial software vendors have an opportunity to both learn from the op- source community as well as leverage that knowledge for the benefit of its commercial clients. Nonetheless, the open-source movement is a diverse collection of ideas, knowledge, techniques, and solutions. As a result, it is far from clear how these approaches should be applied to commercial software engineering. This paper has looked at many of the dimensions of the open-source movement, and provided an analysis of the different opportunities available to commercial software vendors. References and Notes 1. It can be argued that the open-source community has produced really only two essential 9 products -- Apache (undeniably the most popular web server) and Linux although both are essentially reincarnations of prior systems. Both are also somewhat products of their times: Apache filled a hole in the then emerging Web, at a time no platform vendor really knew how to step in, and Linux filled a hole in the fragmented Unix market, colored by the community s general anger against Microsoft. 2.Evans Marketing Services, Linux Developers Survey, Volume 1, March 2000.
Business Component-Based Software Engineering - Franck Barbier 2012-12-06 Business Component-Based Software Engineering, an edited volume, aims to complement some other reputable books on CBSE, by stressing how components are built for large-scale applications, within dedicated development processes and for easy and direct combination. This book will emphasize these three facets and will offer a complete overview of some recent progresses. Projects and works explained herein will prompt graduate students, academics, software engineers, project managers and developers to adopt and to apply new component development methods gained from and validated by the authors. The authors of Business Component-Based Software Engineering are academic and professionals, experts in the field, who will introduce the state of the art on CBSE from their shared experience by working on the same projects. Business Component-Based Software Engineering is designed to meet the needs of practitioners and researchers in industry, and graduate-level students in Computer Science and Engineering.

Component-based Product Line Engineering with UML - Colin Atkinson 2002
A cutting-edge, UML-based approach to software development and maintenance that integrates component-based and product-line engineering methods. - ripe market: development of component-based technologies is a major growth area - CBD viewed as a faster, more flexible way of building systems that can easily be adapted to meet rapidly-changing business needs and integrate legacy and new applications (e.g. Forrester report in June 1998 predicted that by 2001 "half of packaged apps vendors will deliver component-based apps"; e.g. Butler Group Management Briefing (2000): "Butler Group is now advising that all new-build and significant modification activity should be based on component architectures...Butler Group belives that Component-Based Development is one of the most important events in the evolution of information technology" e.g. Gartner Group estimates that "by 2003, 70% of new applications will be deployed as a combination of pre-assembled and newly created components integrated to form complex business-systems. The book defines, describes and shows how to use a method for component-based product-line engineering, supported by UML. This method aims to dramatically increase the level of reuse in software development by integrating the strengths of both of these approaches. UML is used to describe components during the analysis, design & implementation stages and capture their characteristics and relationships.This method includes two new kinds of extensions to the UML: new stereotypes to capture KobrA-specific concepts and new metamodel elements to capture variabilities. The method makes components the focus of the entire software development process, not just the implementation and deployment phases. The method has grown out of work by two companies in industry (Softlab & Psipenta) and two research organizations (GMD FIRST & Fraunhofer IESE) called the KobrA project. It is influenced by a number of successful existing methods e.g. Fusion method, Cleanroom method, Catalysis & Rational Unified Process, integrated with new ideas in an innovative way. Benefits for the reader: - gain a clear understanding of the product-line and component-based approaches

to software development - learn how to use UML to describe components in analysis, design and implementation of components - learn how to develop and apply component-based frameworks in product-lines - learn how to build new systems from pre-existing components and ensure that components are of a high quality The book also includes: - case studies: library system example running throughout the chapters; ERP/business software system as appendix or separate chapter - bibliography - glossary - appendices covering: UML profiles, concise process description in the form of UML activity diagrams, refinement/translation patterns AUDIENCE Software engineers, architects & project managers. Software engineers working in the area of distributed/enterprise systems who want a method for applying a component-based or product-line engineering approach in practice.

Architecting Systems with Trustworthy Components - Ralf H. Reussner 2006-06-30 This book constitutes the thoroughly refereed post-proceedings of the International Dagstuhl-Seminar on Architecting Systems with Trustworthy Components, held in Dagstuhl Castle, Germany, in December 2004. Presents 10 revised full papers together with 5 invited papers contributed by outstanding researchers. Discusses core problems in measurement and normalization of non-functional properties, modular reasoning over non-functional properties, capture of component requirements in interfaces and protocols, interference and synergy of top-down and bottom-up aspects, and more.

Component-Based Software Testing with UML - Hans-Gerhard Gross 2005 Component-based software development regards software construction in terms of conventional engineering disciplines where the assembly of systems from readily-available prefabricated parts is the norm. Because both component-based systems themselves and the stakeholders in component-based development projects are different from traditional software systems, component-based testing also needs to deviate from traditional software testing approaches. Gross first describes the specific challenges related to component-based testing like the lack of internal knowledge of a component or the usage of a component in diverse contexts. He argues that only built-in contract testing, a test organization for component-based applications founded on building test artifacts directly into components, can prevent catastrophic failures like the one that caused the now famous ARIANE 5 crash in 1996. Since building testing into components has implications for component development, built-in contract testing is integrated with and made to complement a model-driven development method. Here UML models are used to derive the testing architecture for an application, the testing interfaces and the component testers. The method also provides a process and guidelines for modeling and developing these artifacts. This book is the first comprehensive treatment of the intricacies of testing component-based software systems. With its strong modeling background, it appeals to researchers and graduate students specializing in component-based software engineering. Professionals architecting and developing component-based systems will profit from the UML-based methodology and the implementation hints based on the XUnit and JUnit frameworks.

Applied Software Product Line Engineering - Kyo C. Kang 2009-12-22 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.

Designing Enterprise Information Systems - Boris Shishkov 2019-11-21
This book brings together enterprise modeling and software specification, providing a conceptual background and methodological guidelines that concern the design of enterprise information systems. In this, two corresponding disciplines (enterprise engineering and software engineering) are considered in a complementary way. This is how the widely recognized gap between domain experts and software engineers could be effectively addressed. The content is, on the one hand, based on a conceptual invariance (embracing concepts whose essence transcends the barriers between social and technical disciplines) while on the other, the book is featuring a modeling duality, by bringing together social theories (that are underlying with regard to enterprise engineering) and computing paradigms (that are underlying as it concerns software engineering). In addition, the proposed approach as well as its guidelines and related notations further foster such enterprise-software modeling, by facilitating modeling generations and transformations. Considering unstructured business information in the beginning, the modeling process would progress through the methodological construction of enterprise models, to reach as far as a corresponding derivation of software specifications. Finally, the enterprise-software alignment is achieved in a component-based way, featuring a potential for re-using modeling constructs, such that the modeling effectiveness and efficiency are further stimulated. For the sake of grounding the presented studies, a case study and illustrative examples are considered. They are not only justifying the idea of bringing together (in a component-based way) enterprise modeling and software specification but they are also demonstrating various strengths and limitations of the proposed modeling approach. The book was mainly written for researchers and graduate students in enterprise information systems, and also for professionals whose work involves the specification and realization of such systems. In addition, researchers and practitioners entering these fields will benefit from the blended view on enterprise modeling and software specification, for the sake of an effective and efficient design of enterprise information systems.

Reuse of Off-the-Shelf Components - Maurizio Morisio 2006-07-25
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.

UML and the Unified Process - Liliana Favre 2003-01-01
"Unified Modeling Language (UML), Unified Process (UP), and other information modeling methods are addressed in this scholarly consideration of the analysis, design, and development of web-based and enterprise applications. The most current research on conceptual, theoretical, and empirical issues of modeling for online business and static information is provided."

Testing Commercial-off-the-Shelf Components and Systems - Sami Beydeda 2005-12-05
Industrial development of software systems needs to be guided by recognized engineering principles. Commercial-off-the-shelf (COTS) components enable the systematic and cost-effective reuse of prefabricated tested parts, a characteristic approach of mature engineering disciplines. This reuse necessitates a thorough

test of these components to make sure that each works as specified in a real context. Beydeda and Gruhn invited leading researchers in the area of component testing to contribute to this monograph, which covers all related aspects from testing components in a context-independent manner through testing components in the context of a specific system to testing complete systems built from different components. The authors take the viewpoints of both component developers and component users, and their contributions encompass functional requirements such as correctness and functionality compliance as well as non-functional requirements like performance and robustness. Overall this monograph offers researchers, graduate students and advanced professionals a unique and comprehensive overview of the state of the art in testing COTS components and COTS-based systems.

NODe 2004 - Mathias Weske 2004-09-23 This book constitutes the refereed proceedings of the 5th Annual International Conference on Object-Oriented and Internet-Based Technologies, Concepts and Applications for a Networked World, Net. Object Days 2004, held in Erfurt, Germany, in September 2004. The 15 revised full papers presented together with an invited paper were carefully reviewed and selected from inclusion in the book. The papers are organised in topical sections on languages and models, agents and the semantic Web, supporting software processes, software product lines, and case studies and visions.

The Common Component Modeling Example - Andreas Rausch 2008-08-15 Based on the 2007 Dagstuhl Research Seminar CoCoME, this book defines a common example for modeling approaches of component-based systems. The book makes it possible to compare different approaches and to validate existing models.

Agent-Oriented Software Engineering X - Marie-Pierre Gleizes 2011-03-14 This volume constitutes the thoroughly refereed post-conference proceedings of the 10th International Workshop on Agent-Oriented Software Engineering, AOSE 2009, held in Budapest, Hungary, in May 2009 as part of AAMAS 2009, the 8th International Conference on Autonomous Agents and Multiagent Systems.

The 10 revised full papers presented were carefully selected from numerous initial submissions during two rounds of reviewing and improvement. The papers have been organized into three sections on multi-agent organizations, concrete development techniques, and - one step higher - going beyond the concrete technique and proposing a development method for designing concrete types of systems. This state-of-the-art survey is rounded off by five additional lectures addressing key areas in development: agent-oriented modelling languages, implementation of MAS, testing of MAS, software processes, and formal methods for the development of MAS. They permit analysis of the current state in the generation of specifications of MAS, the way these specifications can be implemented, how they can be validated, and what steps are necessary to do so.

The Development of Component-based Information Systems - Sergio de Cesare 2015-05-15 This work provides a comprehensive overview of research and practical issues relating to component-based development information systems (CBIS). Spanning the organizational, developmental, and technical aspects of the subject, the original research included here provides fresh insights into successful CBIS technology and application. Part I covers component-based development methodologies and system architectures. Part II analyzes different aspects of managing component-based development. Part III investigates component-based development versus commercial off-the-shelf products (COTS), including the selection and trading of COTS products.

Models in Software Engineering - Michel R. V. Chaudron 2009-04-22 Following the tradition of previous editions of the MODELS conference, many satellite events were organized in co-location with the MODELS conference in Toulouse in 2008: 12 workshops, 3 symposia, 9 tutorials, a poster session, and a tools exhibition. The selection of the workshops was organized by a Workshop Selection Committee, which consisted of the following experts: – Michel R. V. Chaudron, Leiden University, The Netherlands (Chair) – Jochen Kuster ¨ , IBM Research Zurich, Switzerland – Henry Muccini, University of L'Aquila, Italy –

Holger Giese, Hasso-Plattner-Institute, Germany – Hans Vangheluwe, McGill University, Canada Some workshops have been running for several years as MODELS satellite events, but each year some workshops end. Furthermore, there are always new developments,and hence there is roomfor new workshops. Therefore,the Wo- shop Selection Committee very much welcomes new proposals. The workshops enabled groups of participants to exchange recent and/or preliminary results, to conduct intensive discussions, or to coordinate e?orts between representatives of a technical community. They served as forums for lively discussion of innovative ideas, recent progress, or practical experience on model-driven engineering for speci?c aspects, speci?c problems, or doma- speci?c needs. The three symposia this year were: the Doctoral Symposium, the Educators' Symposium, and the Research Projects Symposium. The Doctoral Symposium provided speci?c support for PhD students to discuss their work and receive guidance for the completion of their dissertation research.

Component-Based Software Quality - Alejandra Cechich 2003-06-26
Component-based software development, CBSD, is no longer just one more new paradigm in software engineering, but is effectively used in development and practice. So far, however, most of the efforts from the software engineering community have concentrated on the functional aspects of CBSD, leaving aside the treatment of the quality issues and extra-functional properties of software components and component-based systems. This book is the first one focusing on quality issues of components and component-based systems. The 16 revised chapters presented were carefully reviewed and selected for inclusion in the book; together with an introductory survey, they give a coherent and competent survey of the state of the art in the area. The book is organized in topical parts on COTS selection, testing and certification, software component quality models, formal models to quality assessment, and CBSD management.

New Trends in Software Methodologies, Tools, and Techniques - Hamido Fujita 2004
Software is the essential enabler for the new economy and science. It creates new markets and new directions for a more reliable, flexible, and robust society. It empowers the exploration of our world in ever more depth. However, software often falls short behind our expectations. Current software methodologies, tools, and techniques remain expensive and not yet reliable for a highly changeable and evolutionary market. Many approaches have been proven only as case-by-case oriented methods. This book presents a number of new trends and theories in the direction in which we believe software science and engineering may develop to transform the role of software and science in tomorrows information society. This book is an attempt to capture the essence of a new state of art in software science and its supporting technology. The book also aims at identifying the challenges such a technology has to master. This book covers subjects on Ontology and Software Requirement, Enterprise and Conceptual Software Models, Declarative Representation for Software Development, Requirement Representation and Formalization, Formal Specification and Language Interpretation, Legacy Systems and Language Conversions, Software Quality and Development Measurement, and Software Development Practices Models examples. Each of these chapters contains well-reviewed and selected papers, so the reader can enjoy the state-of-the-art on the need on such new emerged technology.

Software Product-Family Engineering - Frank van der Linden 2004-05-24
This book contains the proceedings of the 5th International Workshop on Product Family Engineering, PFE-5. This workshop was held in Siena, Italy, November 4–6, 2003. This workshop was the fifth in the series, with the same subject, software product family engineering. These workshops have been held initially irregularly about every 18 months since 1996. Since 1999 the workshop has been held every second year in the fall. The proceedings of the second, third and fourth workshops were published as Springer LNCS volumes 1429, 1951 and 2290. The workshops were organized within co-operation projects of European industry. The first two were organized by ARES (Esprit IV 20.477) 1995–1999; this project had 3 industrial and 3 academic partners, and studied software

architectures for product families. Some of the partners continued in the ITEA project if99005 ESAPS (1999–2001). ITEA is the software development programme (?! 2023) within the European Eureka initiative. ITEA projects last for 2 years, and ESAPS was succeeded by CAFÉ (ITEA if00004) for 2001–2003 and FAMILIES (ITEA if02009). This fifth workshop was initially prepared within CAFÉ and the preparation continued in FAMILIES. As usual Henk Obbink was the workshop chair, and Linda Northrop and Sergio Bandinelli were the co-chairs.

Software Product Lines - Gary J. Chastek 2002-07-31
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.
A Practical Guide to Enterprise Architecture - James McGovern 2004

bull; Written by expert practitioners who have hands-on experience solving real-world problems for large corporations bull; Helps enterprise architects make sense of data, systems, software, services, product lines, methodologies, and much more bull; Provides explanation of theory and implementation with real-world business examples to support key points
Designing Software Product Lines with UML - Hassan Gomaa 2005
"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.

Software Evolution and Maintenance - Priyadarshi Tripathy 2014-10-07 Provides students and engineers with the fundamentaldevelopments and common practices of software evolution andmaintenance Software Evolution and Maintenance: A Practitioner'sApproach introduces readers to a set of well-roundededucational materials, covering the fundamental developments insoftware evolution and common maintenance practices in theindustry. Each chapter gives a clear understanding of a particulartopic in software evolution, and discusses the main ideas withdetailed examples. The authors first explain the basic concepts andthen drill deeper into the important aspects of software evolution.While designed as a text in an undergraduate course in softwareevolution and maintenance, the book is also a great resourceforsoftware engineers, information technology professionals, andgraduate students in software engineering. Based on the IEEE SWEBOK (Software

Engineering Body ofKnowledge) Explains two maintenance standards: IEEE/EIA 1219 andISO/IEC14764 Discusses several commercial reverse and domain engineeringtoolkits Slides for instructors are available online Software Evolution and Maintenance: APractitioner's Approach equips readers with a solidunderstanding of the laws of software engineering, evolution andmaintenance models, reengineering techniques, legacy informationsystems, impact analysis, refactoring, program comprehension, andreuse.

Proceedings of the Estonian Academy of Sciences, Engineering - 2005-12

Development of Component-based Information Systems - Sergio De Cesare 2006 Annotation This work provides a comprehensive overview of research and practical issues relating to component-based information systems (CBIS). Spanning the organizational, developmental, and technical aspects of the subject, the original research included here provides fresh insights into successful CBIS technology and application, including the selection and trading of commercial off-the shelf products (COTS).

Model-Driven Software Development - Markus Völter 2013-06-26 Model-Driven Software Development (MDSD) is currently a highlyregarded development paradigm among developers and researchers.With the advent of OMG's MDA and Microsoft's Software Factories,the MDSD approach has moved to the centre of the programmer'sattention, becoming the focus of conferences such as OOPSLA, JAOOand OOP. MDSD is about using domain-specific languages to create models thatexpress application structure or behaviour in an efficient anddomain-specific way. These models are subsequently transformed intoexecutable code by a sequence of model transformations. This practical guide for software architects and developers ispeppered with practical examples and extensive case studies.International experts deliver: * A comprehensive overview of MDSD and how it relates to industrystandards such as MDA and Software Factories. * Technical details on meta modeling, DSL construction,model-to-model and model-to-code transformations, and softwarearchitecture. * Invaluable insight into the software development process, plusengineering issues such as versioning, testing and product lineengineering. * Essential management knowledge covering economic andorganizational topics, from a global perspective. Get started and benefit from some practical support along the way!

On the Move to Meaningful Internet Systems: OTM 2011 Workshops - Robert Meersman 2011-11-09 This volume constitutes the refereed proceedings of nine international workshops, EI2N+NSF ICE, ICSP, INBAST, ISDE, MONET, ORM, SeDeS, SWWS, and VADER 2011, held as part of OTM 2011 in Hersonissos on the island of Crete, Greece, in October 2011. The 64 revised full papers presented were carefully reviewed and selected from a total of 104 submissions. The volume also includes three papers from the On the Move Academy (OTMA) 2011 and five ODBASE 2011 poster papers. Topics of the workshop papers are enterprise integration and semantics, information centric engineering, interoperability, industrial and business applications of semantic Web applications, information systems in distributed environments, process management in distributed information system development, distributed information systems: implementation issues and applications, industrial applications of fact-oriented modeling, data warehouse modeling, extensions to fact-oriented modeling, model validation procedures, schema transformations and mapping, semantic Web and Web semantics, ontology development, deployment and interoperability, data access and efficient computation, efficient information processing, exchange and knowledge synthesis algorithms, mobile and networking technologies for social applications, semantic and decision support, variability in software architecture, and dynamic and adaptive architectures.

Testing Object-oriented Systems - Robert Binder 2000 More than ever, mission-critical and business-critical applications depend on object-oriented (OO) software. Testing techniques tailored to the unique challenges of OO technology are necessary to achieve high reliability and quality. "Testing Object-Oriented Systems: Models,

Patterns, and Tools" is an authoritative guide to designing and automating test suites for OO applications. This comprehensive book explains why testing must be model-based and provides in-depth coverage of techniques to develop testable models from state machines, combinational logic, and the Unified Modeling Language (UML). It introduces the test design pattern and presents 37 patterns that explain how to design responsibility-based test suites, how to tailor integration and regression testing for OO code, how to test reusable components and frameworks, and how to develop highly effective test suites from use cases. Effective testing must be automated and must leverage object technology. The author describes how to design and code specification-based assertions to offset testability losses due to inheritance and polymorphism. Fifteen micro-patterns present oracle strategies--practical solutions for one of the hardest problems in test design. Seventeen design patterns explain how to automate your test suites with a coherent OO test harness framework. The author provides thorough coverage of testing issues such as: The bug hazards of OO programming and differences from testing procedural code How to design responsibility-based tests for classes, clusters, and subsystems using class invariants, interface data flow models, hierarchic state machines, class associations, and scenario analysis How to support reuse by effective testing of abstract classes, generic classes, components, and frameworks How to choose an integration strategy that supports iterative and incremental development How to achieve comprehensive system testing with testable use cases How to choose a regression test approach How to develop expected test results and evaluate the post-test state of an object How to automate testing with assertions, OO test drivers, stubs, and test frameworks Real-world experience, world-class best practices, and the latest research in object-oriented testing are included. Practical examples illustrate test design and test automation for Ada 95, C++, Eiffel, Java, Objective-C, and Smalltalk. The UML is used throughout, but the test design patterns apply to systems developed with any OO language or methodology. 0201809389B04062001

Objects, Components, Architectures,

Services, and Applications for a Networked World - Mehmet Aksit 2003-02-25
This book constitutes the thoroughly refereed post-proceedings of the international conference NetObjectDays 2002, held in Erfurt, Germany, in October 2002. The 26 revised full papers presented were carefully selected during two rounds of reviewing and revision. The papers are organized in topical sections on embedded and distributed systems; components and MDA; Java technology; Web services; aspect-oriented software design; agents and mobility; software product lines; synchronization; testing, refactoring, and CASE tools.

Model-Driven and Software Product Line Engineering - Jean-Claude Royer 2013-03-06
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.

Software Product Lines - Timo Käkölä 2007-02-07
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.

Software Product Lines - Patrick Donohoe 2012-12-06
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.

Component-Based Software Engineering - Grace A. Lewis 2009-06-09
The 2009 Symposium on Component-Based Software Engineering (CBSE 2009) was the 12thin a series ofsuccessful eventsthat havegrowninto the main forum for industrial and academic experts to discuss component technology. Component-based software engineering (CBSE) has emerged as the under-ing technology for the assembly of ?exible software systems. In essence, CBSE is about composing computational building blocks to construct larger building blocks that ful?ll client needs. Most software engineers are involved in some form of component-based development. Nonetheless, the implications of CBSE adoption are wide-reaching and its challenges grow in tandem with its uptake, continuing to inspire our scienti?c speculation. Component-based development necessarily involves elements of software - chitecture, modular software design, software veri?cation, testing, con?guration and deployment. This year's submissions represent a cross-section of CBSE - search that touches upon all these aspects. The theoretical foundations of c- ponent speci?cation, composition, analysis, and veri?cation continue to pose research challenges. What exactly constitutes an

adequate semantics for c- munication and composition so that bigger things can be built from smaller things? How can formal approaches facilitate predictable assembly through b- ter analysis? We have grouped the proceedings into two sub-themes that deal with these issues: component models and communication and composition. At the same time, the world is changing.

Requirements Engineering: Foundation for Software Quality - Pete Sawyer 2007-06-28
This book constitutes the refereed proceedings of the 13th International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ 2007, held in Trondheim, Norway. It covers goal-driven requirements engineering (RE), products and product-lines, value-based RE and the value of RE, requirements elicitation, requirements specification, industrial experience of RE, and requirements quality and quality requirements.

Software Product Lines - Patrick Donohoe 2000-08-31
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.