Get Free End User Software Engineering In The Spreadsheet Paradigm Free Download Pdf

Automotive Software Engineering **Automotive Software Engineering** Einführung in Software Engineering *Software Engineering und Prototyping* **Software Engineering - kompakt** Essentials of Software Engineering Software Engineering und Prototyping **Software Engineering in C Grundkurs Software-Engineering mit UML Fundamentals of Software Engineering Grundkurs Software-Engineering mit UML Software Engineering durch Modellierung wissensintensiver Entwicklungsprozesse** *Software Engineering* **Software Engineering for Multi-Agent Systems III Grundlagen des Software Engineering Software Tools: Automatisierung im Software Engineering Software-Engineering** Best Practice Software-Engineering *Software Engineering for Multi-Agent Systems II* **Concise Guide to Software Engineering** *Requirements Engineering for Software and Systems, Second Edition* **Software Engineering for Embedded Systems** *Model-Driven Software Engineering in Practice Bausteinbasierte Software Software Engineering* Software-Engineering **Software Engineering im Unterricht der Hochschulen SEUH '93 Best Practice Software-Engineering Software Engineering Design Building Great Software Engineering Teams** *Software Engineering* Encyclopedia of Software Engineering Three-Volume Set (Print) **Software Engineering Projektorganisation und Management im Software Engineering Software Engineering in IoT, Big Data, Cloud and Mobile Computing Rethinking Productivity in Software Engineering** Software-Engineering Essentials of Software Engineering *Iterative Software Engineering for Multiagent Systems* **Handbook of Software Engineering**

Software Engineering und Prototyping Jul 28 2022 Als erste Monographie im deutschsprachigen Raum vermittelt das vorliegende Buch eine Konstruktionslehre des Software Engineering über den gesamten Lebenszyklus eines Softwareproduktes. Während Software-Technologie üblicherweise in Hochschulen oder Softwarehäusern entsteht, wird hier eine Technologie dargestellt, die bei Anwendern entstanden ist. Nur Anwender sind in großem Stil mit dem gesamten Lebenszyklus von Software konfrontiert, da sie über 50% ihres DV-Personals für die Wartung der investierten Software einsetzen müssen. Es wird ein neues, objekt-orientiertes Vorgehensmodell für die Entwicklung kommerzieller Dialogsoftware vorgestellt. Die dabei besonders wichtige Funktion und die Erscheinungsformen des Prototyping im Software-Entwicklungsprozeß werden konstruktiv geklärt, wobei der Aspekt der Kommunikation der am Entwicklungsprozeß Beteiligten besonders herausgearbeitet wird. Das Buch vermittelt eine durchgehende, produktneutrale Methodik, hinter der 10 Jahre Industrieerfahrung, aber keine Verkaufsinteressen für bestimmte Hardware- oder Softwarewerkzeuge stehen. Trotzdem wird die Benutzung von Softwarewerkzeugen fundiert behandelt, da eine prozeßorientierte Software-Entwicklung nur mit Werkzeugen möglich ist. Neben den frühen Phasen, in denen Methoden zur Datenmodellierung und Prototyping als kommunikationsunterstützende Methode wesentlich sind, wird die "Phase" Wartung vertieft behandelt, in der die wichtigsten Entscheidungen bei der Evolution von Software fallen; hierzu wird auch ein praktisch eingesetztes Werkzeug skizziert. In einer Bibliographie sind sowohl grundlegende Quellen als auch aktuelle weiterführende Literatur zusammengestellt.

Projektorganisation und Management im Software Engineering Dec 29 2019 Die Entwicklung umfangreicher Softwaresysteme erfordert neben der Beherrschung von Softwaretechnik auch eine fundierte Projektplanung, -organisation und -durchführung. Nötig sind Kenntnisse der Aufwandsschätzung und des Angebots- und Vertragswesens. Das einzigartige Lehrbuch für Studenten bietet auch für Berufseinsteiger die Basis für Vorgehensweisen im Software Engineering. Die Kombination von praktischen Erfahrungen und methodischen Grundlagen zeigt die Anwendbarkeit der Inhalte auf. Zahlreiche Übungsaufgaben vertiefen das gewonnene Wissen.

Software Engineering for Embedded Systems Jan 10 2021 Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications, Second Edition provides the techniques and technologies in software engineering to optimally design and implement an embedded system. Written by experts with a solution focus, this encyclopedic reference gives an indispensable aid on how to tackle the day-to-day problems encountered when using software engineering methods to develop embedded systems. New sections cover peripheral programming, Internet of things, security and cryptography, networking and packet processing, and hands on labs. Users will learn about the principles of good architecture for an embedded system, design practices, details on principles, and much more. Provides a roadmap of key problems/issues and references to their solution in the text Reviews core methods and how to apply them Contains examples that demonstrate timeless implementation details Users case studies to show how key ideas can be implemented, the rationale for choices made, and design guidelines and trade-offs

Software Engineering Mar 31 2020 Das Buch vermittelt eine neue Sicht zur Software-Technik, in dem es vor allem den Engineering Aspekt stärker berücksichtigt. Das Buch ist Web-bezogen aufgebaut und ermöglicht damit, das Internet für den Wissenerwerb und ausgewählte Übungen zu nutzen.

Essentials of Software Engineering Aug 24 2019 Computer Architecture/Software Engineering

Requirements Engineering for Software and Systems, Second Edition Feb 08 2021 As requirements engineering continues to be recognized as the key to on-time and on-budget delivery of software and systems projects, many engineering programs have made requirements engineering mandatory in their curriculum. In addition, the wealth of new software tools that have recently emerged is empowering practicing engineers to improve their requirements engineering habits. However, these tools are not easy to use without appropriate training. Filling this need, Requirements Engineering for Software and Systems, Second Edition has been vastly updated and expanded to include about 30 percent new material. In addition to new exercises and updated references in every chapter, this edition updates all chapters with the latest applied research and industry practices. It also presents new material derived from the experiences of professors who have used the text in their classrooms. Improvements to this edition include: An expanded introductory chapter with extensive discussions on requirements analysis, agreement, and consolidation An expanded chapter on requirements engineering for Agile methodologies An expanded chapter on formal methods with new examples An expanded section on requirements traceability An updated and expanded section on requirements engineering tools New exercises including ones suitable for research projects Following in the footsteps of its bestselling predecessor, the text illustrates key ideas associated with requirements engineering using extensive case studies and three common example systems: an airline baggage handling system, a point-of-sale system for a large pet store chain, and a system for a smart home. This edition also includes an example of a wet well pumping system for a wastewater treatment station. With a focus on software-intensive systems, but highly applicable to non-software systems, this text provides a probing and comprehensive review of recent developments in requirements engineering in high integrity systems.

Einführung in Software Engineering Aug 29 2022

Software Engineering in C Mar 24 2022 1 Introduction To Programming.- 1.1 High-Level Programming Languages.- 1.2 History of C.- 1.3 ANSI Standard.- 1.4 Nature of C.- 2 C Essentials.- 2.1 Program Development.- 2.2 Functions.- 2.3 Anatomy of a C Function.- 2.4 Formatting Source Files.- 2.5 The main() Function.- 2.6 The printf() Function.- 2.7 The scanf() Function.- 2.8 The Preprocessor.- Exercises.- 3 Scalar Data Types.- 3.1 Declarations.- 3.2 Different Types of Integers.- 3.3 Different Kinds of Integer Constants.- 3.4 Floating-Point Types.- 3.5 Initialization.- 3.6 Finding the Address of an Object.- 3.7 Introduction to Pointers.- 3.8 Typedefs.- 3.9 Mixing Types.- 3.10 Explicit Conversions - Casts.- 3.11 Enumeration Types.- 3.12 The void Data Type.- Exercises.- 4 Control Flow.- 4.1 Conditional Branching.-

4.2 The switch Statement.- 4.3 Looping.- 4.4 Nested Loops.- 4.5 A Simple Calculator Program.- 4.6 The break and continue Statements.- 4.7 The goto Statement.- 4.8 Infinite Loops.- Exercises.- 5 Operators and Expressions.- 5.1 Precedence and Associativity.- 5.2 Unary Minus Operator.- 5.3 Binary Arithmetic Operators.- 5.4 Arithmetic Assignment Operators.- 5.5 Increment and Decrement Operators.- 5.6 Comma Operator.- 5.7 Relational Operators.- 5.8 Logical Operators.- 5.9 Bit-Manipulation Operators.- 5.10 Bitwise Assignment Operators.- 5.11 Cast Operator.- 5.12 sizeof operator.- 5.13 Conditional Operator (?:).- 5.14 Memory Operators.- Exercises.- 6 Arrays and Pointers.- 6.1 Declaring an Array.- 6.2 How Arrays Are Stored in Memory.- 6.3 Initializing Arrays.- 6.4 Array Example: Encryption and Decryption.- 6.5 Pointer Arithmetic.- 6.6 Passing Pointers as Function Arguments.- 6.7 Accessing Array Elements Through Pointers.- 6.8 Passing Arrays as Function Arguments.- 6.9 Sorting Algorithms.- 6.10 Strings.- 6.11 Multidimensional Arrays.- 6.12 Arrays of Pointers.- 6.13 Pointers to Pointers.- Exercises.- 7 Storage Classes.- 7.1 Fixed vs. Automatic Duration.- 7.2 Scope.- 7.3 Global Variables.- 7.4 The register Specifier.- 7.5 Summary of Storage Classes.- 7.6 Dynamic Memory Allocation.- Exercises.- 8 Structures and Unions.- 8.1 Structures.- 8.2 Linked Lists.- 8.3 Unions.- 8.4 enum Declarations.- Exercises.- 9 Functions.- 9.1 Passing Arguments.- 9.2 Declarations and Calls.- 9.3 Pointers to Functions.- 9.4 Recursion.- 9.5 The main() Function.- 9.6 Complex Declarations.- Exercises.- 10 The C Preprocessor.- 10.1 Macro Substitution.- 10.2 Conditional Compilation.- 10.3 Include Facility.- 10.4 Line Control.- Exercises.- 11 Input and Output.- 11.1 Streams.- 11.2 Buffering.- 11.3 The Header File.- 11.4 Error Handling.- 11.5 Opening and Closing a File.- 11.6 Reading and Writing Data.- 11.7 Selecting an I/O Method.- 11.8 Unbuffered I/O.- 11.9 Random Access.- Exercises.- 12 Software Engineering.- 12.1 Product Specification.- 12.2 Software Design.- 12.3 Project Management and Cost Estimation.- 12.4 Software Tools for Software Production.- 12.5 Debugging.- 12.6 Testing.- 12.7 Performance Analysis.- 12.8 Documentation.- Exercises.- Appendix A The ANSI Runtime Library.- A.1 Function Names.- A.2 Header Files.- A.3 Synopses.- A.4 Functions vs. Macros.- A.5 Error Handling.- A.6 Diagnostics.- A.7 Character Handling.- A.8 Setting Locale Parameters.- A.9 Mathematics.- A.10 Non-Local Jumps.- A.11 Signal Handling.- A.12 Variable Argument Lists.- A.13 I/O Functions.- A.14 General Utilities.- A.15 String-Handling Functions.- A.16 Date and Time Functions.- Appendix B Syntax of ANSI C.- Appendix C Implementation Limits.- C.1 Translation Limits.- C.2 Numerical Limits.- Appendix D Differences Between the ANSI and K&R Standards.- D.1 Source Translation Differences.- D.2 Data Type Differences.- D.3 Statement Differences.- D.4 Expression Differences.- D.5 Storage Class and Initialization Differences.- D.6 Preprocessor Differences.- Appendix E Reserved Names.- Appendix F C Interpreter Listing.- Appendix G ASCII Codes.

Software-Engineering Sep 25 2019 Die Kosten zur Erstellung von Software steigen im Vergleich zu den Hardwarekosten ständig. Um die wachsenden Anwenderansprüche zu befriedigen und mit dem rapiden technischen Fortschritt im Hardwarebereich Schritt halten zu können, wurde eine eigene Ingenieurdisziplin "Software-Engineering" notwendig. Das ingenieurmäßige Vorgehen soll sicherstellen, daß die Software-Herstellung termingerecht, kostengünstig, rationell und qualitätsbewußt geschieht. Software-Engineering ist ein ganz junges Gebiet der Informationswissenschaft. Deshalb sind die meisten Werke über diesen Bereich von der wissenschaftlichen Diskussion geprägt und im Anspruchsniveau relativ hoch. Für viele in der Programmierpraxis stehende Per sonen fehlt ein Lehrbuch einfacheren Charakters, das die Methoden und Hilfsmittel für die besonders wichtigen und kostenintensiven Phasen Entwurf und Test beschreibt. ln diese Lücke möchte dieses Buch stoßen. Aus der Fülle der in der Literatur vorgeschlagenen Methoden wurden diejenigen ausge wählt, die einerseits auf Grund ihrer systematischen Vorgehensweise besonders effizient sind und die andererseits wegen ihrer leichten Lernbarkeit und Einsetzbarkeit in der Praxis am häufigsten und erfolgreichsten Anwendung finden. Aus Gründen der Klarheit wurde auf eine eingehende Problematisierung der vorgestellten Methoden verzichtet. Der an einer kritischen Hinterfragung interessierte Leser sei auf das ausführliche Literaturver zeichnis verwiesen.

Best Practice Software-Engineering May 14 2021 Software-Komponenten tragen durch einen hohen Grad an Wiederverwendbarkeit, bessere Testbarkeit und Wartbarkeit zur effizienten Herstellung komplexer Software-Anwendungen bei. Diese Vorteile bedingen jedoch oft eine aufwendigere Einarbeitung beim Einstieg in diese Materie durch die Vielzahl an komplexen Komponenten-Frameworks, Werkzeugen und Entwurfsansätzen. Das vorliegende Buch „Best-Practice Software Engineering" bietet Neu- und Wiedereinsteigern in die komponentenorientierte Software-Entwicklung eine Einführung in die Materie durch eine abgestimmte Zusammenstellung von praxiserprobten Konzepten, Techniken und Werkzeugen für alle Aspekte eines erfolgreichen Projekts. Für moderne Software-Entwicklung sind eine Vielzahl von unterschiedlichen Fähigkeiten erforderlich, die nur im richtiger Kombination zu einem erfolgreichen Ergebnis führen. Daher wird in diesem Buch besonderer Wert darauf gelegt, nicht einzelne Techniken des Software Engineerings isoliert zu betrachten, sondern das effiziente Zusammenspiel verschiedener Aspekte darzustellen. Schwerpunkte liegen auf Vorgehensstrategien im Software-Lebenszyklus, Projektmanagement, Qualitätssicherung, UML-Modellierung, Entwurfsmustern und Architekturen, komponentenorientierter Software-Entwicklung sowie ausgewählten Techniken und Werkzeugen. Zu den Beispielen im Buch finden Sie den vollständigen Source Code sowie umfangreiche Fallbeispiele zu Artefakten aus dem Projektverlauf auf der Webseite zum Buch.

Software Engineering und Prototyping Apr 24 2022 Als erste Monographie im deutschsprachigen Raum vermittelt das vorliegende Buch eine Konstruktionslehre des Software Engineering über den gesamten Lebenszyklus eines Softwareproduktes. Während Software-Technologie üblicherweise in Hochschulen oder Softwarehäusern entsteht, wird hier eine Technologie dargestellt, die bei Anwendern entstanden ist. Nur Anwender sind in großem Stil mit dem gesamten Lebenszyklus von Software konfrontiert, da sie über 50% ihres DV-Personals für die Wartung der investierten Software einsetzen müssen. Es wird ein neues, objekt-orientiertes Vorgehensmodell für die Entwicklung kommerzieller Dialogsoftware vorgestellt. Die dabei besonders wichtige Funktion und die Erscheinungsformen des Prototyping im Software-Entwicklungsprozeß werden konstruktiv geklärt, wobei der Aspekt der Kommunikation der am Entwicklungsprozeß Beteiligten besonders herausgearbeitet wird. Das Buch vermittelt eine durchgehende, produktneutrale Methodik, hinter der 10 Jahre Industrieerfahrung, aber keine Verkaufsinteressen für bestimmte Hardware- oder Softwarewerkzeuge stehen. Trotzdem wird die Benutzung von Softwarewerkzeugen fundiert behandelt, da eine prozeßorientierte Software-Entwicklung nur mit Werkzeugen möglich ist. Neben den frühen Phasen, in denen Methoden zur Datenmodellierung und Prototyping als kommunikationsunterstützende Methode wesentlich sind, wird die "Phase" Wartung vertieft behandelt, in der die wichtigsten Entscheidungen bei der Evolution von Software fallen; hierzu wird auch ein praktisch eingesetztes Werkzeug skizziert. In einer Bibliographie sind sowohl grundlegende Quellen als auch aktuelle weiterführende Literatur zusammengestellt.

Software Tools: Automatisierung im Software Engineering Jul 16 2021 Erstmalig werden in einem Buch Automatisierungspotentiale und Werkzeuge der Software-Entwicklung gemeinsam dargestellt. Begonnen wird mit einer Analyse der automatisierbaren Tätigkeiten im Software Life Cycle. Darauf aufbauend erfolgt die Präsentation einer an den Funktionen und Einsatzbereichen der Software Tools orientierten Systematik; sie unterstützt sowohl die theoretische Einordnung als auch die praxisorientierte Auswahl der Programmierwerkzeuge. Parallel dazu werden die wesentlichen Eigenschaften der Werkzeuge auf der Basis einer vereinheitlichten Terminologie erläutert. Dadurch erhält der Leser einen umfassenden Überblick über Funktionen und Einsatzbereiche von Software Tools (incl. Fourth Generation Languages - 4GLs), der auch weniger gebräuchliche Tools mit teilweise ungewöhnlichen Funktionen berücksichtigt.

Software Engineering - kompakt Jun 26 2022 Im Software-Engineering geht es um die Modellierung und Entwicklung komplexer, qualitativ hochwertiger Software und die für einen erfolgreich durchgeführten Realisierungsprozess geeigneten Methoden, Werkzeuge und Standards. In diesem kompakten Lehrbuch werden die wichtigsten Themen rund um Software-Engineering erklärt, zusammengefasst und mit kleinen Praxisbeispielen vertieft. Von zentraler Bedeutung für das Software-Engineering ist der Software-Lebenszyklus. Gemeint ist damit der gesamte Prozess, der zur Erstellung und Erhaltung eines Softwaresystems führt. Sowohl in traditionellen als auch in agilen Softwareerstellungsprozessen läuft dieser Lebenszyklus ab. Bewährt hat sich in der Praxis die Einteilung in sogenannte Phasen, denen die Gliederung folgt. Nach einer kurzen Einführung werden in Kapitel 2 vorab phasenübergreifende Verfahren wie divergierende Vorgehensmodelle und Projektmanagement

besprochen. Kapitel 3 behandelt die Planungsphase; Kapitel 4 ist dem Requirements-Engineering gewidmet, bei dem die Software-Anforderungen kreativ konstruiert, analysiert und – traditionell oder agil – dokumentiert werden. In Kapitel 5 folgt die Besprechung der Verfahren für die Designphase der Software. Hier wird hinterfragt, wie gute Software-Architekturen Erfolg versprechend erdacht, mit der UML-Notation geeignet modelliert und in späteren Projekten wiederverwendet werden können. Kapitel 6 widmet sich der Test- und Abnahmephase und damit den wichtigen Qualitätssicherungsfragen. Abschließend wird in Kapitel 7 die Wartung – zur wirksamen Erhaltung von Softwaresystemen – erklärt. Anfänger erhalten eine schnelle Orientierung und kompaktes, fundiertes Grundwissen. Fortgeschrittene Leser finden hier ein aktuelles, gut strukturiertes Nachschlagewerk. Unter https://www.hanser-fachbuch.de/buch/Software+Engineering+kompakt/9783446459496 finden interessierte Leser weitere Übungsaufgaben zum Thema Software-Engineering.

Model-Driven Software Engineering in Practice Dec 09 2020 This book discusses how model-based approaches can improve the daily practice of software professionals. This is known as Model-Driven Software Engineering (MDSE) or, simply, Model-Driven Engineering (MDE). MDSE practices have proved to increase efficiency and effectiveness in software development, as demonstrated by various quantitative and qualitative studies. MDSE adoption in the software industry is foreseen to grow exponentially in the near future, e.g., due to the convergence of software development and business analysis. The aim of this book is to provide you with an agile and flexible tool to introduce you to the MDSE world, thus allowing you to quickly understand its basic principles and techniques and to choose the right set of MDSE instruments for your needs so that you can start to benefit from MDSE right away. The book is organized into two main parts. The first part discusses the foundations of MDSE in terms of basic concepts (i.e., models and transformations), driving principles, application scenarios and current standards, like the well-known MDA initiative proposed by OMG (Object Management Group) as well as the practices on how to integrate MDSE in existing development processes. The second part deals with the technical aspects of MDSE, spanning from the basics on when and how to build a domain-specific modeling language, to the description of Model-to-Text and Model-to-Model transformations, and the tools that support the management of MDSE projects. The book is targeted to a diverse set of readers, spanning: professionals, CTOs, CIOs, and team managers that need to have a bird's eye vision on the matter, so as to take the appropriate decisions when it comes to choosing the best development techniques for their company or team; software analysts, developers, or designers that expect to use MDSE for improving everyday work productivity, either by applying the basic modeling techniques and notations or by defining new domain-specific modeling languages and applying end-to-end MDSE practices in the software factory; and academic teachers and students to address undergrad and postgrad courses on MDSE. In addition to the contents of the book, more resources are provided on the book's website, including the examples presented in the book. Table of Contents: Introduction / MDSE Principles / MDSE Use Cases / Model-Driven Architecture (MDA) / Integration of MDSE in your Development Process / Modeling Languages at a Glance / Developing your Own Modeling Language / Model-to-Model Transformations / Model-to-Text Transformations / Managing Models / Summary

Software-Engineering Jun 14 2021 Software-Engineering befaßt sich mit der Entwicklung von Softwaresystemen, insbesondere den dafür nötigen und zweckmäßigen Methoden und Werkzeugen. Dabei geht es nicht nur um die technische Gestaltung von Systemen, also deren Architektur, sondern auch um die geordnete Abwicklung von Projekten, also um Managementfragen. Dieses Buch ist der Extrakt aus eineinhalb Jahrzehnten Arbeit an einer Reihe großer, unter industriellen Bedingungen durchgeführter Projekte. Es behandelt hauptsächlich Methoden - nur in geringem Umfang Werkzeuge - des Software-Engineering, genauer gesagt, das von sd&m praktizierte Methodensystem, das theoretisch fundiert und praktisch erprobt ist. Die objektorientierte Methodik spielt darin eine zentrale Rolle. Der Erfahrungshintergrund des Autors ist stark, wenn auch keineswegs ausschließlich, durch betriebliche Informationssysteme geprägt. Die dargestellten Methoden sind aber so allgemeingültig, das sie auch in anderen Anwendungsbereichen nutzbringend anwendbar sind.

Best Practice Software-Engineering Jul 04 2020 Software-Komponenten tragen durch einen hohen Grad an Wiederverwendbarkeit, bessere Testbarkeit und Wartbarkeit zur effizienten Herstellung komplexer Software-Anwendungen bei. Diese Vorteile bedingen jedoch oft eine aufwendigere Einarbeitung beim Einstieg in diese Materie durch die Vielzahl an komplexen Komponenten-Frameworks, Werkzeugen und Entwurfsansätzen. Das vorliegende Buch „Best Practice Software-Engineering" bietet Neu- und Wiedereinsteigern in die komponentenorientierte Software-Entwicklung eine Einführung in die Materie durch eine abgestimmte Zusammenstellung von praxiserprobten Konzepten, Techniken und Werkzeugen für alle Aspekte eines erfolgreichen Projekts. Für moderne Software-Entwicklung sind eine Vielzahl von unterschiedlichen Fähigkeiten erforderlich, die nur in richtiger Kombination zu einem erfolgreichen Ergebnis führen. Daher wird in diesem Buch besonderer Wert darauf gelegt, nicht einzelne Techniken des Software-Engineerings isoliert zu betrachten, sondern das effiziente Zusammenspiel verschiedener Aspekte darzustellen. Schwerpunkte liegen auf Vorgehensstrategien im Software-Lebenszyklus, Projektmanagement, Qualitätssicherung, UML-Modellierung, Entwurfsmustern und Architekturen, komponentenorientierter Software-Entwicklung sowie ausgewählten Techniken und Werkzeugen. Zu den Beispielen im Buch finden Sie den vollständigen Sourcecode sowie umfangreiche Fallbeispiele zu Artefakten aus dem Projektverlauf auf der Webseite zum Buch (http://bpse.ifs.tuwien.ac.at). .

Software Engineering durch Modellierung wissensintensiver Entwicklungsprozesse Nov 19 2021

Iterative Software Engineering for Multiagent Systems Jul 24 2019 The agent metaphor and the agent-based approach to systems design constitute a promising new paradigm for building complex distributed systems. However, until now, the majority of the agent-based applications available have been built by researchers who specialize in agent-based computing and distributed artificial intelligence. If agent-based computing is to become anything more than a niche technology practiced by the few, then the base of people who can successfully apply the approach needs to be broadened dramatically. A major step in this broadening endeavor is the development of methodologies for agent-oriented software engineering accessible to and attractive for professional software engineers in their daily work. Against this background, this book presents one of the first coherent attempts to develop such a methodology for a broad class of agent-based systems. The author provides a clear introduction to the key issues in the field of agent-oriented software engineering.

Essentials of Software Engineering May 26 2022 Written for the undergraduate, one-term course, Essentials of Software Engineering, Fourth Edition provides students with a systematic engineering approach to software engineering principles and methodologies. Comprehensive, yet concise, the Fourth Edition includes new information on areas of high interest to computer scientists, including Big Data and developing in the cloud.

Grundlagen des Software Engineering Aug 17 2021 Software -- Software Engineering.

Rethinking Productivity in Software Engineering Oct 26 2019 Get the most out of this foundational reference and improve the productivity of your software teams. This open access book collects the wisdom of the 2017 "Dagstuhl" seminar on productivity in software engineering, a meeting of community leaders, who came together with the goal of rethinking traditional definitions and measures of productivity. The results of their work, Rethinking Productivity in Software Engineering, includes chapters covering definitions and core concepts related to productivity, guidelines for measuring productivity in specific contexts, best practices and pitfalls, and theories and open questions on productivity. You'll benefit from the many short chapters, each offering a focused discussion on one aspect of productivity in software engineering. Readers in many fields and industries will benefit from their collected work. Developers wanting to improve their personal productivity, will learn effective strategies for overcoming common issues that interfere with progress. Organizations thinking about building internal programs for measuring productivity of programmers and teams will learn best practices from industry and researchers in measuring productivity. And researchers can leverage the conceptual frameworks and rich body of literature in the book to effectively pursue new research directions. What You'll Learn Review the definitions and dimensions of software productivity See how time management is having the opposite of the intended effect Develop valuable dashboards Understand the impact of sensors on productivity Avoid software development waste Work with human-centered methods to measure productivity Look at the intersection of neuroscience and productivity Manage interruptions and context-

switching Who Book Is For Industry developers and those responsible for seminar-style courses that include a segment on software developer productivity. Chapters are written for a generalist audience, without excessive use of technical terminology.

Encyclopedia of Software Engineering Three-Volume Set (Print) Feb 29 2020 Software engineering requires specialized knowledge of a broad spectrum of topics, including the construction of software and the platforms, applications, and environments in which the software operates as well as an understanding of the people who build and use the software. Offering an authoritative perspective, the two volumes of the Encyclopedia of Software Engineering cover the entire multidisciplinary scope of this important field. More than 200 expert contributors and reviewers from industry and academia across 21 countries provide easy-to-read entries that cover software requirements, design, construction, testing, maintenance, configuration management, quality control, and software engineering management tools and methods. Editor Phillip A. Laplante uses the most universally recognized definition of the areas of relevance to software engineering, the Software Engineering Body of Knowledge (SWEBOK®), as a template for organizing the material. Also available in an electronic format, this encyclopedia supplies software engineering students, IT professionals, researchers, managers, and scholars with unrivaled coverage of the topics that encompass this ever-changing field. Also Available Online This Taylor & Francis encyclopedia is also available through online subscription, offering a variety of extra benefits for researchers, students, and librarians, including: Citation tracking and alerts Active reference linking Saved searches and marked lists HTML and PDF format options Contact Taylor and Francis for more information or to inquire about subscription options and print/online combination packages. US: (Tel) 1.888.318.2367; (E-mail) e-reference@taylorandfrancis.com International: (Tel) +44 (0) 20 7017 6062; (E-mail) online.sales@tandf.co.uk

Software Engineering Design Jun 02 2020 Taking a learn-by-doing approach, Software Engineering Design: Theory and Practice uses examples, review questions, chapter exercises, and case study assignments to provide students and practitioners with the understanding required to design complex software systems. Explaining the concepts that are immediately relevant to software designers, it begins with a review of software design fundamentals. The text presents a formal top-down design process that consists of several design activities with varied levels of detail, including the macro-, micro-, and construction-design levels. As part of the top-down approach, it provides in-depth coverage of applied architectural, creational, structural, and behavioral design patterns. For each design issue covered, it includes a step-by-step breakdown of the execution of the design solution, along with an evaluation, discussion, and justification for using that particular solution. The book outlines industry-proven software design practices for leading large-scale software design efforts, developing reusable and high-quality software systems, and producing technical and customer-driven design documentation. It also: Offers one-stop guidance for mastering the Software Design & Construction sections of the official Software Engineering Body of Knowledge (SWEBOK®) Details a collection of standards and guidelines for structuring high-quality code Describes techniques for analyzing and evaluating the quality of software designs Collectively, the text supplies comprehensive coverage of the software design concepts students will need to succeed as professional design leaders. The section on engineering leadership for software designers covers the necessary ethical and leadership skills required of software developers in the public domain. The section on creating software design documents (SDD) familiarizes students with the software design notations, structural descriptions, and behavioral models required for SDDs. Course notes, exercises with answers, online resources, and an instructor’s manual are available upon qualified course adoption. Instructors can contact the author about these resources via the author's website: http://softwareengineeringdesign.com/

Software Engineering im Unterricht der Hochschulen SEUH ’93 Aug 05 2020

Concise Guide to Software Engineering Mar 12 2021 This textbook presents a concise introduction to the fundamental principles of software engineering, together with practical guidance on how to apply the theory in a real-world, industrial environment. The wide-ranging coverage encompasses all areas of software design, management, and quality. Topics and features: presents a broad overview of software engineering, including software lifecycles and phases in software development, and project management for software engineering; examines the areas of requirements engineering, software configuration management, software inspections, software testing, software quality assurance, and process quality; covers topics on software metrics and problem solving, software reliability and dependability, and software design and development, including Agile approaches; explains formal methods, a set of mathematical techniques to specify and derive a program from its specification, introducing the Z specification language; discusses software process improvement, describing the CMMI model, and introduces UML, a visual modelling language for software systems; reviews a range of tools to support various activities in software engineering, and offers advice on the selection and management of a software supplier; describes such innovations in the field of software as distributed systems, service-oriented architecture, software as a service, cloud computing, and embedded systems; includes key learning topics, summaries and review questions in each chapter, together with a useful glossary. This practical and easy-to-follow textbook/reference is ideal for computer science students seeking to learn how to build high quality and reliable software on time and on budget. The text also serves as a self-study primer for software engineers, quality professionals, and software managers.

Bausteinbasierte Software Nov 07 2020 Das Buch zeigt, wie mit Hilfe des Konzeptes "Baustein" moderne Formen des Software-Engineering realisiert und dadurch u.a. Componentware und Frameworks verstanden und entwickelt werden können. Der Leser erfährt, wie Baustein als generalisiertes Konzept eine durchgängige Systematisierung unterschiedlicher Stufen und verschiedener Realisierungsformen von Softwarekomponenten ermöglicht. Damit wird eine in sich geschlossene Behandlung so verschiedenartiger Komponenten wie Frameworks, Entwurfsmuster oder Klassen, Entwurfskonstrukt, Quellcode oder Binärcode erreicht. Grundzüge der Architekturen verbunden mit direkt nachvollziehbaren Beispielen bringen Theorie und Praxis näher zusammen. Systematische Konstruktion mittels Bausteinen unter objektorientierten Bedingungen ist dabei die zentrale Idee. Ein kompaktes Lehrbuch für Studenten der Informatik und ein nützliches Nachschlagewerk für die Praxis moderner Softwareentwicklung mit Anwendungen auch aus den Bereichen Multimedia, Intra- und Internet.

Automotive Software Engineering Sep 29 2022 Nahezu alle Funktionen des Fahrzeugs werden inzwischen elektronisch gesteuert, geregelt oder überwacht. Die Realisierung von Funktionen durch Software bietet einzigartige Freiheitsgrade beim Entwurf. In der Fahrzeugentwicklung müssen jedoch Randbedingungen wie hohe Zuverlässigkeits- und Sicherheitsanforderungen, vergleichsweise lange Produktlebenszyklen, begrenzte Kosten, verkürzte Entwicklungszeiten und zunehmende Variantenvielfalt berücksichtigt werden. Dieses Buch enthält Grundlagen und praktische Beispiele zu Prozessen, Methoden und Werkzeugen, die zur sicheren Beherrschbarkeit von elektronischen Systemen und Software im Fahrzeug beitragen. Dabei stehen die elektronischen Systeme des Antriebsstrangs, des Fahrwerks und der Karosserie im Vordergrund. Die überarbeitete 3. Auflage enthält verbesserte Bilddarstellungen sowie ein deutsch-englisches Sachwortverzeichnis.

Software Engineering for Multi-Agent Systems III Sep 17 2021 This book presents a coherent and well-balanced survey of recent advances in software engineering approaches to the design and analysis of realistic large-scale multi-agent systems (MAS). The chapters included are devoted to various techniques and methods used to cope with the complexity of real-world MAS. The power of agent-based software engineering is illustrated using examples that are representative of successful applications. The 16 thoroughly reviewed and revised full papers are organized in topical sections on agent methodologies and processes, requirements engineering and software architectures, modeling languages, and dependability and coordination. Most of the papers were initially presented at the 3rd International Workshop on Software Engineering for Large-Scale Multi-agent Systems, SELMAS 2004, held in Edinburgh, UK in May 2004 in association with ICSE 2004. Other papers were invited to complete coverage of all relevant aspects.

Software-Engineering Sep 05 2020

Grundkurs Software-Engineering mit UML Feb 20 2022 Mit der Entwicklung neuer Technologien werden auch die einzelnen Software-Projekte stetig komplexer. Zu analysieren, warum manche Projekte scheitern und andere erfolgreich sind, wird daher immer wichtiger. Dieses Buch ist ein praktischer Leitfaden für die Entwicklung neuer

Software. Systematisch beschreibt der Autor die Chancen und Risiken, die einem bei der Entwicklung einer Software begegnen können. Vom gemeinsamen Kundengespräch, das Anforderungen und Ziele der Software festlegt, über die erste Modellierung bis hin zur systematischen Erfassung der Anforderungen zeigt er, wie die unterschiedlichen Prozesse mit Hilfe der UML (Unified Modeling Language) koordiniert werden können. Diese Modellierungssprache hilft, die Ideen des Entwicklers nachzuvollziehen und die Erfahrungen aus erfolgreichen Projekten auf andere Projekte zu übertragen.Neben Maßnahmen zur Qualitätssicherung beschreibt das Buch weitere Ansätze zur Projektplanung und Projektdurchführung und zeigt, wie die Softwareentwicklung in den Gesamtprozess eines Unternehmens eingebettet ist. Zum Verständnis des Buches werden Grundkenntnisse in einer objektorientierten Programmiersprache wie Java, C# oder C++ vorausgesetzt. Durch zahlreiche Wiederholungsfragen und Übungsaufgaben am Ende der Kapitel wird dieses Buch zum idealen Begleiter für Studenten der Informatik und verschiedener Ingenieurswissenschaften. Aber auch erfahrene Entwickler können von den vielen Kommentaren zur Verwendung in der Praxis zur kontinuierlichen Weiterentwicklung des Software-Engineerings profitieren. Die vorliegende vierte Auflage des bewährten Buches enthält erneut wichtige Erweiterungen und Ergänzungen.
Software Engineering for Multi-Agent Systems II Apr 12 2021 This book presents a coherent and well-balanced survey of recent advances in software engineering approaches to the development of realistic multi-agent systems (MAS). In it, the concept of agent-based software engineering is demonstrated through examples that are relevant to and representative of real-world applications. The 15 thoroughly reviewed and revised full papers are organized in topical sections on requirements engineering, software architecture and design, modeling, dependability, and MAS frameworks. Most of the papers were initially presented at the Second International Workshop on Software Engineering for Large-Scale Multi-Agent Systems, SELMAS 2003, held in Portland, Oregon, USA, in May 2003; three papers were added in order to complete the coverage of the relevant topics.
Building Great Software Engineering Teams May 02 2020 Building Great Software Engineering Teams provides engineering leaders, startup founders, and CTOs concrete, industry-proven guidance and techniques for recruiting, hiring, and managing software engineers in a fast-paced, competitive environment. With so much at stake, the challenge of scaling up a team can be intimidating. Engineering leaders in growing companies of all sizes need to know how to find great candidates, create effective interviewing and hiring processes, bring out the best in people and their work, provide meaningful career development, learn to spot warning signs in their team, and manage their people for long-term success. Author Josh Tyler has spent nearly a decade building teams in high-growth startups, experimenting with every aspect of the task to see what works best. He draws on this experience to outline specific, detailed solutions augmented by instructive stories from his own experience. In this book you'll learn how to build your team, starting with your first hire and continuing through the stages of development as you manage your team for growth and success. Organized to cover each step of the process in the order you'll likely face them, and highlighted by stories of success and failure, it provides an easy-to-understand recipe for creating your high-powered engineering team.What you'll learn Effective techniques for finding engineering candidates for your company, including how to make your company more attractive to prospective employees and tips for navigating the employment visa process How to leverage commonly overlooked resources for finding employees, such as hiring from other geographic regions and how to approach college recruiting How to successfully hire the best candidates, from first contact through making an offer and getting it accepted How to manage engineers for optimum morale and performance, foster confidence throughout your organization, and promote career development for your team members What to expect as you build an engineering team: common challenges, growing pains, and solutions How to use team-building skills to propel your career as individual contributor Who this book is for The primary audience is engineering leaders, startup founders, CTOs, and others tasked with building an engineering team to fulfill their company's mission. The secondary audience is any engineering manager or senior engineer interested in learning more about how to hire or manage engineers effectively. Table of Contents The Challenge of Building an Engineering Team from the Ground Up An Enlightened Approach to Recruiting Six Destructive Myths about Recruiting Software Engineers Eight Steps to Recruiting Success Hiring Is Hard The Myth of the Ninja Rockstar Developer The Hiring Decision Checklist Making Interviews Fun for Your Team Why We Don't Allow Java in Job Interviews Do I Want to be a Manager? A Manager's Most Important Deliverable Technical vs Management Tracks: Helping Your People Grow Tricks of the Trade for Engineering Managers Advice to Give Engineers on Finding a Great Job and Growing in Their Careers
Handbook of Software Engineering Jun 22 2019 This handbook provides a unique and in-depth survey of the current state-of-the-art in software engineering, covering its major topics, the conceptual genealogy of each subfield, and discussing future research directions. Subjects include foundational areas of software engineering (e.g. software processes, requirements engineering, software architecture, software testing, formal methods, software maintenance) as well as emerging areas (e.g., self-adaptive systems, software engineering in the cloud, coordination technology). Each chapter includes an introduction to central concepts and principles, a guided tour of seminal papers and key contributions, and promising future research directions. The authors of the individual chapters are all acknowledged experts in their field and include many who have pioneered the techniques and technologies discussed. Readers will find an authoritative and concise review of each subject, and will also learn how software engineering technologies have evolved and are likely to develop in the years to come. This book will be especially useful for researchers who are new to software engineering, and for practitioners seeking to enhance their skills and knowledge.
Software Engineering in IoT, Big Data, Cloud and Mobile Computing Nov 27 2019 This edited book presents scientific results of the International Semi-Virtual Workshop on Software Engineering in IoT, Big data, Cloud and Mobile Computing (SE-ICBM 2020) which was held on October 15, 2020, at Soongsil University, Seoul, Korea. The aim of this workshop was to bring together researchers and scientists, businessmen and entrepreneurs, teachers, engineers, computer users, and students to discuss the numerous fields of computer science and to share their experiences and exchange new ideas and information in a meaningful way. Research results about all aspects (theory, applications and tools) of computer and information science, and to discuss the practical challenges encountered along the way and the solutions adopted to solve them. The workshop organizers selected the best papers from those papers accepted for presentation at the workshop. The papers were chosen based on review scores submitted by members of the program committee and underwent further rigorous rounds of review. From this second round of review, 17 of the conference's most promising papers are then published in this Springer (SCI) book and not the conference proceedings. We impatiently await the important contributions that we know these authors will bring to the field of computer and information science.
Software Engineering Jan 28 2020 "The ninth edition of Software Engineering presents a broad perspective of software engineering, focusing on the processes and techniques fundamental to the creation of reliable, software systems. Increased coverage of agile methods and software reuse, along with coverage of 'traditional' plan-driven software engineering, gives readers the most up-to-date view of the field currently available. Practical case studies, a full set of easy-to-access supplements, and extensive web resources make teaching the course easier than ever."--Publisher's website.
Software Engineering Oct 19 2021 Each and every chapter covers the contents up to a reasonable depth necessary for the intended readers in the field. The book consists in all about 1200 exercises based on the topics and sub-topics covered. Keeping in view the emerging trends in newly emerging scenario with new dimension of software engineering, the book specially includes the following chapters, but not limited to these only. This book explains all the notions related to software engineering in a very systematic way, which is of utmost importance to the novice readers in the field of software Engineering.
Software Engineering Oct 07 2020
Grundkurs Software-Engineering mit UML Dec 21 2021 Software-Projekte scheitern aus den unterschiedlichsten Gründen. Dieses Buch zeigt anhand der systematischen Analyse von Chancen und Risiken, wie die Wege zu erfolgreichen Software-Projekten aussehen. Ausgehend von der Basis, dass das Zusammenspiel aller an einem Projekt Beteiligten in Prozessen koordiniert werden soll, wird mit Hilfe der UML (Unified Modeling Language) der Weg von den Anforderungen über die Modellierung bis zur Implementierung beschrieben. Es werden situationsabhängige Alternativen diskutiert und der gesamte Prozess mit qualitätssichernden Maßnahmen begleitet. Zur Abrundung des Themengebiets werden wichtige Ansätze

zur Projektplanung und zur Projektdurchführung beschrieben, die die Einbettung der Software-Entwicklung in die Gesamtprozesse eines Unternehmens aufzeigen. Alle Kapitel schließen mit Wiederholungsfragen und Übungsaufgaben. Lösungsskizzen sind über das Internet erhältlich.

Automotive Software Engineering Oct 31 2022 Dieses Fachbuch enthält die Grundlagen sowie zahlreiche Anregungen und praktische Beispiele zu Prozessen, Methoden und Werkzeugen, die zur sicheren Beherrschbarkeit von elektronischen Systemen und Software im Fahrzeug beitragen. Dabei werden der AUTOSAR-Standard und die Norm ISO 26 262 durchgehend behandelt. Die aktuelle Auflage berücksichtigt Elektro- und Hybridantriebskonzepte sowie Fahrerassistenzsysteme und enthält die Grundlagen zu Produktlinien- und Variantenmanagement. Seit Anfang der 1970er Jahre ist die Entwicklung von Kraftfahrzeugen geprägt von einem rasanten Anstieg des Einsatzes von Elektronik und Software. Dieser Trend hält bis heute an und wird getrieben von steigenden Kunden- und Umweltanforderungen. Nahezu alle Funktionen des Fahrzeugs werden inzwischen elektronisch gesteuert, geregelt oder überwacht. Die Realisierung von Funktionen durch Software bietet einzigartige Freiheitsgrade beim Entwurf. In der Fahrzeugentwicklung müssen jedoch Randbedingungen wie hohe Zuverlässigkeits- und Sicherheitsanforderungen, lange Produktlebenszyklen, begrenzte Kostenrahmen, kurze Entwicklungszeit und zunehmende Variantenvielfalt berücksichtigt werden. In diesem Spannungsfeld steht Automotive Software Engineering.

Fundamentals of Software Engineering Jan 22 2022 Practical Handbook to understand the hidden language of computer hardware and software DESCRIPTION This book teaches the essentials of software engineering to anyone who wants to become an active and independent software engineer expert. It covers all the software engineering fundamentals without forgetting a few vital advanced topics such as software engineering with artificial intelligence, ontology, and data mining in software engineering. The primary goal of the book is to introduce a limited number of concepts and practices which will achieve the following two objectives: Teach students the skills needed to execute a smallish commercial project. Provide students with the necessary conceptual background for undertaking advanced studies in software engineering through courses or on their own. KEY FEATURES - This book contains real-time executed examples along with case studies. - Covers advanced technologies that are intersectional with software engineering. - Easy and simple language, crystal clear approach, and straight forward comprehensible presentation. - Understand what architecture design involves, and where it fits in the full software development life cycle. - Learning and optimizing the critical relationships between analysis and design. - Utilizing proven and reusable design primitives and adapting them to specific problems and contexts. WHAT WILL YOU LEARN This book includes only those concepts that we believe are foundational. As executing a software project requires skills in two dimensions—engineering and project management—this book focuses on crucial tasks in these two dimensions and discuss the concepts and techniques that can be applied to execute these tasks effectively. WHO THIS BOOK IS FOR The book is primarily intended to work as a beginner’s guide for Software Engineering in any undergraduate or postgraduate program. It is directed towards students who know the program but have not had formal exposure to software engineering. The book can also be used by teachers and trainers who are in a similar state—they know some programming but want to be introduced to the systematic approach of software engineering. TABLE OF CONTENTS 1. Introductory Concepts of Software Engineering 2. Modelling Software Development Life Cycle 3. Software Requirement Analysis and Specification 4. Software Project Management Framework 5. Software Project Analysis and Design 6. Object-Oriented Analysis and Design 7. Designing Interfaces & Dialogues and Database Design 8. Coding and Debugging 9. Software Testing 10. System Implementation and Maintenance 11.Reliability 12. Software Quality 13. CASE and Reuse 14. Recent Trends and Development in Software Engineering 15. Model Questions with Answers