Software Engineering for Real-time Systems
Cryptography in C and C++
ACM SIGPLAN Notices
Descriptive Summaries for Program Elements of the Research, Development, Test and Evaluation, Army Program FY (U).
Safer C
Embedded Microprocessor Systems
Embedded Software
The CERT® C Coding Standard, Second Edition
Trustworthy Systems Through Quantitative Software Engineering
Electronic Design
Developing Safety-Critical Software
Computer Safety, Reliability and Security
Hazard Analysis Techniques for System Safety
Formal Methods
Learning Swift
Federally Coordinated Program of Highway Research, Development and Technology
Modern Fortran in Practice
Software Safety and Reliability
Safer Bridge Railings. Volume 4: Appendix C., Part II. Final Report
CENELEC 50128 and IEC 62279
Standards
Software Engineering
Computer Security
Critical Information Infrastructures
Software Metrics
Reliable Software Technologies – Ada-Europe 2007
Software Design and Development: Concepts, Methodologies, Tools, and Applications
Safety and Reliability of Software Based Systems
Real-time Design Patterns
Automotive Electronics (Autotech '97)
Handbook of Bioequivalence Testing
Achievement and Assurance of Safety
Federally Coordinated Program of Highway Research and Development
Accuracy and Reliability in Scientific Computing
Secure Programming Cookbook for C and C++
Programming Rust
Working Effectively with Legacy Code
Beginning Programming All-In-One Desk Reference For Dummies
Second IEEE International Software Engineering Standards Symposium (ISESS'95)
Cryptographic Security Architecture
Embedded microprocessor systems are affecting our daily lives at a fast pace, mostly unrecognised by the general public. Most of us are aware of the part they are playing in increasing business efficiency through office applications such as personal computers, printers and copiers. Only a few people, however, fully appreciate the growing role of embedded systems in telecommunications and industrial environments, or even in everyday products like cars and home appliances. The challenge to engineers and managers is not only highlighted by the sheer size of the market, ' 1.5 billion microcontrollers and microprocessors are produced every year ' but also by the accelerating innovation in embedded systems towards higher complexity in hardware, software and tools as well as towards higher performance and lower consumption. To maintain competitiveness in this demanding environment, an optimum mix of innovation, time to market and system cost is required. Choosing the right options and strategies for products and companies is crucial and rarely obvious. In this book the editors have, therefore, skilfully brought together more than fifty contributions from some of the leading authorities in embedded systems. The papers are conveniently grouped in four sections.
CENELEC EN 50128 and IEC 62279 standards are applicable to the performance of software in the railway sector. The 2011 version of the 50128 standard firms up the techniques and methods to be implemented. This is a guide to its implementation, in order to understand the foundations of the standard and how it impacts on the activities to be undertaken, helping towards better a preparation for the independent evaluation phase, which is mandatory. The comprehensive coverage and real-world perspective makes the book accessible and appealing to both beginners and experienced designers. Covers both the fundamentals of software design and modern design
methodologies Provides comparisons of different development methods, tools and languages Blends theory and practical experience together Emphasises the use of diagrams and is highly illustrated“At Cisco, we have adopted the CERT C Coding Standard as the internal secure coding standard for all C developers. It is a core component of our secure development lifecycle. The coding standard described in this book breaks down complex software security topics into easy-to-follow rules with excellent real-world examples. It is an essential reference for any developer who wishes to write secure and resilient software in C and C++.” —Edward D. Paradise, vice president, engineering, threat response, intelligence, and development, Cisco Systems Secure programming in C can be more difficult than even many experienced programmers realize. To help programmers write more secure code, The CERT® C Coding Standard, Second Edition, fully documents the second official release of the CERT standard for secure coding in C. The rules laid forth in this new edition will help ensure that programmers’ code fully complies with the new C11 standard; it also addresses earlier versions, including C99. The new standard itemizes those coding errors that are the root causes of current software vulnerabilities in C, prioritizing them by severity, likelihood of exploitation, and remediation costs. Each of the text’s 98 guidelines includes examples of insecure code as well as secure, C11-conforming, alternative implementations. If uniformly applied, these guidelines will eliminate critical coding errors that lead to buffer overflows, format-string vulnerabilities, integer overflow, and other common vulnerabilities. This book reflects numerous experts’ contributions to the open development and review of the rules and recommendations that comprise this standard. Coverage includes Preprocessor Declarations and Initialization Expressions Integers Floating Point Arrays Characters and Strings Memory Management Input/Output Environment Signals Error Handling Concurrency Miscellaneous IssuesThe world moves on Critical Information Infrastructures, and their resilience and protection is of vital importance. Starting with some basic definitions and assumptions on the topic, this book goes on to explore various aspects of Critical Infrastructures throughout the world – including the technological, political, economic, strategic and defensive. This book will be of interest to the CEO and Academic alike as they grapple with how to prepare Critical Information Infrastructures for new challenges.As the generic pharmaceutical industry continues to grow and thrive, so does the need to conduct adequate, efficient bioequivalence studies. In recent years, there have been significant changes to the statistical models for evaluating bioequivalence. In addition, advances in the analytical technology used to detect drug and metabolite levels have mThe amount of software used in safety-critical systems is increasing at a rapid rate. At the same time, software technology is changing, projects are pressed to develop software faster and more cheaply, and the software is being used in more critical ways. Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance equips you with the information you need to effectively and efficiently develop safety-critical, life-critical, and mission-critical software for aviation. The principles also apply to software for automotive, medical, nuclear, and other safety-critical domains. An international authority on safety-critical software, the author helped write DO-178C and the U.S. Federal Aviation Administration’s policy and guidance on safety-critical software. In this book, she draws on more than 20 years of experience as a certification authority, an avionics manufacturer, an aircraft integrator, and a software developer to present best practices, real-world examples, and concrete recommendations. The book includes: An overview of how software fits into the systems and safety
processes. Detailed examination of DO-178C and how to effectively apply the guidance Insight into the DO-178C-related documents on tool qualification (DO-330), model-based development (DO-331), object-oriented technology (DO-332), and formal methods (DO-333) Practical tips for the successful development of safety-critical software and certification Insightful coverage of some of the more challenging topics in safety-critical software development and verification, including real-time operating systems, partitioning, configuration data, software reuse, previously developed software, reverse engineering, and outsourcing and offshoring An invaluable reference for systems and software managers, developers, and quality assurance personnel, this book provides a wealth of information to help you develop, manage, and approve safety-critical software more confidently.

Each year there are improvements in safety-critical system technology. These arise both from developments in the contributing technologies, such as safety engineering, software engineering, human factors and risk assessment, and from the adoption or adaptation of appropriate techniques from other domains, such as security. For these improvements to be of real benefit, they need to be applied during the appropriate stage in the life cycle of the system, whether it be development, assessment, or operation. For this to occur, they must be communicated and explained. Each year the Safety-critical Systems Symposium offers a distinguished forum for the presentation of papers on such developments, and also for papers from industry on the lessons learned from the use of technologies and methods. The results of many collaborative research projects, with components from both industry and academia, are reported in a universally understandable form. In 1995 the Symposium was held in Brighton, a venue calculated to stimulate not just the presenters of papers, but all the delegates. Yet, this book of Proceedings is intended not only for the delegates but also for readers not able to attend the event itself. We welcome both categories of reader. Delegates have the benefit of attending the presentations and the opportunity to participate in the discussions; those who take up this book after the event can peruse it at their leisure and, perhaps, on account of it will resolve to attend subsequent symposia.

Numerical software is used to test scientific theories, design airplanes and bridges, operate manufacturing lines, control power plants and refineries, analyze financial derivatives, identify genomes, and provide the understanding necessary to derive and analyze cancer treatments. Because of the high stakes involved, it is essential that results computed using software be accurate, reliable, and robust. Unfortunately, developing accurate and reliable scientific software is notoriously difficult. This book investigates some of the difficulties related to scientific computing and provides insight into how to overcome them and obtain dependable results. The tools to assess existing scientific applications are described, and a variety of techniques that can improve the accuracy and reliability of newly developed applications is discussed. Accuracy and Reliability in Scientific Computing can be considered a handbook for improving the quality of scientific computing. It will help computer scientists address the problems that affect software in general as well as the particular challenges of numerical computation: approximations occurring at all levels, continuous functions replaced by discretized versions, infinite processes replaced by finite ones, and real numbers replaced by finite precision numbers. Divided into three parts, it starts by illustrating some of the difficulties in producing robust and reliable scientific software. Well-known cases of failure are reviewed and the what and why of numerical computations are considered. The second section describes diagnostic tools that can be used to assess the accuracy and
reliability of existing scientific applications. In the last section, the authors describe a variety of techniques that can be employed to improve the accuracy and reliability of newly developed scientific applications. The authors of the individual chapters are international experts, many of them members of the IFIP Working Group on Numerical Software. Reliable Software Technologies is an annual series of international conferences devoted to the promotion and advancement of all aspects of reliable software technologies. The objective of this series of conferences, initiated and sponsored by Ada-Europe, the European federation of national Ada societies, is to provide a forum to promote the development of reliable softwares both as an industrial technique and an academic discipline. Previous editions of the Reliable Software Technologies conference were held in: Porto (Portugal) in 2006, York (UK) in 2005, Palma de Mallorca (Spain) in 2004, Toulouse (France) in 2003, Vienna (Austria) in 2002, Leuven (Belgium) in 2001, Potsdam (Germany) in 2000, Santander (Spain) in 1999, Uppsala (Sweden) in 1998, London (UK) in 1997 and Montreux (Switzerland) in 1996. The 12th International Conference on Reliable Software Technologies took place in Geneva, Switzerland, June 25–29, 2007, under the continued sponsoring of Ada-Europe, in cooperation with ACM SIGAda. It was organized by members of the University of Applied Sciences, Western Switzerland (Engineering School of Geneva), in collaboration with colleagues from various places in Europe. The 13th conference, in 2008, will take place in Venice, Italy. Presents a novel design that allows for a great deal of customization, which many current methods fail to include; Details a flexible, comprehensive design that can be easily extended when necessary; Proven results: the versatility of the design has been effectively tested in implementations ranging from microcontrollers to supercomputers; A Framework for Managing, Measuring, and Predicting Attributes of Software Development Products and Processes Reflecting the immense progress in the development and use of software metrics in the past decades, Software Metrics: A Rigorous and Practical Approach, Third Edition provides an up-to-date, accessible, and comprehensive introduction to software metrics. Like its popular predecessors, this third edition discusses important issues, explains essential concepts, and offers new approaches for tackling long-standing problems. New to the Third Edition This edition contains new material relevant to object-oriented design, design patterns, model-driven development, and agile development processes. It includes a new chapter on causal models and Bayesian networks and their application to software engineering. This edition also incorporates recent references to the latest software metrics activities, including research results, industrial case studies, and standards. Suitable for a Range of Readers With numerous examples and exercises, this book continues to serve a wide audience. It can be used as a textbook for a software metrics and quality assurance course or as a useful supplement in any software engineering course. Practitioners will appreciate the important results that have previously only appeared in research-oriented publications. Researchers will welcome the material on new results as well as the extensive bibliography of measurement-related information. The book also gives software managers and developers practical guidelines for selecting metrics and planning their use in a measurement program. Safety and Reliability of Software Based Systems contains papers presented at the twelfth annual workshop organised by the Centre for Software Reliability. Contributions come from different industries in many countries, and provide discussion and cross-fertilisation of ideas relevant to systems whose safety and/or reliability are of paramount concern. This book discusses safety cases and their varying roles in different industries;
Where To Download Safer C Developing Software For High Integrity And Safety Critical Systems Mcgraw Hill International Series In Software Engineering

using measurement to improve reliability and safety of software-based systems; latest developments in managing, developing and assessing software intensive systems where reliability and/or safety are important considerations; and practical experiences of others in industry. The role of software has changed from simply generating financial or mechanical data to monitoring and controlling equipment that directly affects human life and safety. As a result, a more thorough understanding and familiarity with the specialized techniques used to achieve and assess the safety and reliability of software is needed in academia, industry, and government. This original text introduces the concepts, techniques, and approaches used to achieve and assess software safety and reliability. Debra Herrmann presents a cross-section of current safety and reliability standards that cross multiple industrial sectors while focusing on the additional required activities to achieve software safety and reliability. In organizing this text, the she has three objectives. The first is to raise the reader's awareness on the importance of software safety and reliability and on its role in mission critical systems by presenting many illustrative, ever day examples. The second objective is to provide practical information about the current methods used to achieve and assess software safety and reliability. The final objective is to improve the understanding and practice of software safety and reliability by consolidating the latest research so that it can be compared and analyzed for the future. The book is written for engineers, scientists, managers, regulators, and policy makers involved in the design, development, acquisition, and certification of safety-critical systems. Innovative tools and techniques for the development and design of software systems are essential to the problem solving and planning of software solutions. Software Design and Development: Concepts, Methodologies, Tools, and Applications brings together the best practices of theory and implementation in the development of software systems. This reference source is essential for researchers, engineers, practitioners, and scholars seeking the latest knowledge on the techniques, applications, and methodologies for the design and development of software systems. So you want to be a programmer? Or maybe you just want to be able to make your computer do what YOU want for a change? Maybe you enjoy the challenge of identifying a problem and solving it. If programming intrigues you for whatever reason, Beginning Programming All-In-One Desk Reference For Dummies is like having a starter programming library all in one handy, if beefy, book. In this practical guide, you'll find out about compiling, algorithms, best practices, debugging your programs, and much more. The concepts are illustrated in several different programming languages, so you'll get a feel for the variety of languages and the needs they fill. Seven minibooks cover: Getting started Programming basics Data structures Algorithms Web programming Language syntax Applications Beginning Programming All-In-One Desk Reference For Dummies shows you how to decide what you want your program to do, turn your instructions into “machine language” that the computer understands, use programming best practices, explore the “how” and “why” of data structuring, and more. You’ll even get a look into various applications like database management, bioinformatics, computer security, and artificial intelligence. Soon you’ll realize that — wow! You’re a programmer! Note: CD-ROM/DVD and other supplementary materials are not included as part of eBook file. Password sniffing, spoofing, buffer overflows, and denial of service: these are only a few of the attacks on today's computer systems and networks. At the root of this epidemic is poorly written, poorly tested, and insecure code that puts everyone at risk. Clearly, today's developers need help figuring out how to write code that
attackers won't be able to exploit. But writing such code is surprisingly difficult. Secure Programming Cookbook for C and C++ is an important new resource for developers serious about writing secure code. It contains a wealth of solutions to problems faced by those who care about the security of their applications. It covers a wide range of topics, including safe initialization, access control, input validation, symmetric and public key cryptography, cryptographic hashes and MACs, authentication and key exchange, PKI, random numbers, and anti-tampering. The rich set of code samples provided in the book's more than 200 recipes will help programmers secure the C and C++ programs they write for both Unix® (including Linux®) and Windows® environments. Readers will learn: How to avoid common programming errors, such as buffer overflows, race conditions, and format string problems How to properly SSL-enable applications How to create secure channels for client-server communication without SSL How to integrate Public Key Infrastructure (PKI) into applications Best practices for using cryptography properly Techniques and strategies for properly validating input to programs How to launch programs securely How to use file access mechanisms properly Techniques for protecting applications from reverse engineering The book's web site supplements the book by providing a place to post new recipes, including those written in additional languages like Perl, Java, and Python. Monthly prizes will reward the best recipes submitted by readers. Secure Programming Cookbook for C and C++ is destined to become an essential part of any developer's library, a code companion developers will turn to again and again as they seek to protect their systems from attackers and reduce the risks they face in today's dangerous world.

Rust is a new systems programming language that combines the performance and low-level control of C and C++ with memory safety and thread safety. Rust’s modern, flexible types ensure your program is free of null pointer dereferences, double frees, dangling pointers, and similar bugs, all at compile time, without runtime overhead. In multi-threaded code, Rust catches data races at compile time, making concurrency much easier to use. Written by two experienced systems programmers, this book explains how Rust manages to bridge the gap between performance and safety, and how you can take advantage of it. Topics include: How Rust represents values in memory (with diagrams) Complete explanations of ownership, moves, borrows, and lifetimes Cargo, rustdoc, unit tests, and how to publish your code on crates.io, Rust’s public package repository High-level features like generic code, closures, collections, and iterators that make Rust productive and flexible Concurrency in Rust: threads, mutexes, channels, and atomics, all much safer to use than in C or C++ Unsafe code, and how to preserve the integrity of ordinary code that uses it Extended examples illustrating how pieces of the language fit together This revised and enlarged edition of a classic in Old Testament scholarship reflects the most up-to-date research on the prophetic books and offers substantially expanded discussions of important new insight on Isaiah and the other prophets. A tutorial guide that shows programmers how to apply features of Fortran 2008 in a modular, concise, object-oriented and resource-efficient manner, using multiple processors. Featuring an associated Web page, and consistently combining theory with real-world practical applications, this text includes thought-provoking questions about legal and ethical issues in software engineering. Computers and their interactions are becoming the characteristic features of our time: Many people believe that the industrial age is going over into the information age. In the same way as life of the beginning of this century was dominated by machines, factories, streets and railways, the starting century will be characterised by computers and their networks. This change naturally affects also the
Institutions and the installations our lives depend upon: power plants, including nuclear ones, chemical plants, mechanically working factories, cars, railways and medical equipment; they all depend on computers and their connections. In some cases it is not human life that may be endangered by computer failure, but large investments; e. g. if a whole plant interrupts its production for a long time. In addition to loss of life and property one must not neglect public opinion, which is very critical in many countries against major technical defects. The related computer technology, its hardware, software and production process differ between standard applications and safety related ones: In the safety case it is normally not only the manufacturers and the customers that are involved, but a third party, usually an assessor, who is taking care of the public interest on behalf of a state authority. Usually safety engineers are in a better position than their colleagues from the conventional side, as they may spend more time and money on a particular task and use better equipment.

Get more out of your legacy systems: more performance, functionality, reliability, and manageability

Is your code easy to change? Can you get nearly instantaneous feedback when you do change it? Do you understand it? If the answer to any of these questions is no, you have legacy code, and it is draining time and money away from your development efforts. In this book, Michael Feathers offers start-to-finish strategies for working more effectively with large, untested legacy code bases. This book draws on material Michael created for his renowned Object Mentor seminars: techniques Michael has used in mentoring to help hundreds of developers, technical managers, and testers bring their legacy systems under control. The topics covered include: Understanding the mechanics of software change: adding features, fixing bugs, improving design, optimizing performance Getting legacy code into a test harness Writing tests that protect you against introducing new problems Techniques that can be used with any language or platform—with examples in Java, C++, C, and C# Accurately identifying where code changes need to be made Coping with legacy systems that aren't object-oriented Handling applications that don't seem to have any structure This book also includes a catalog of twenty-four dependency-breaking techniques that help you work with program elements in isolation and make safer changes.

Broad themes of this text on the subject of automotive electronics include: EMC and component design; software and controllers; electronic systems; testing and maintenance; and power devices. A completely up-to-date resource on computer security Assuming no previous experience in the field of computer security, this must-have book walks you through the many essential aspects of this vast topic, from the newest advances in software and technology to the most recent information on Web applications security. This new edition includes sections on Windows NT, CORBA, and Java and discusses cross-site scripting and JavaScript hacking as well as SQL injection. Serving as a helpful introduction, this self-study guide is a wonderful starting point for examining the variety of competing security systems and what makes them different from one another. Unravels the complex topic of computer security and breaks it down in such a way as to serve as an ideal introduction for beginners in the field of computer security. Examines the foundations of computer security and its basic principles. Addresses username and password, password protection, single sign-on, and more. Discusses operating system integrity, hardware security features, and memory. Covers Unix security, Windows security, database security, network security, web security, and software security. Packed with in-depth coverage, this resource spares no details when it comes to the critical topic of computer security. Although formal analysis programming techniques may be
quite old, the introduction of formal methods only dates from the 1980s. These techniques enable us to analyze the behavior of a software application, described in a programming language. It took until the end of the 1990s before formal methods or the B method could be implemented in industrial applications or be usable in an industrial setting. Current literature only gives students and researchers very general overviews of formal methods. The purpose of this book is to present feedback from experience on the use of "formal methods" (such as proof and model-checking) in industrial examples within the transportation domain. This book is based on the experience of people who are currently involved in the creation and evaluation of safety critical system software. The involvement of people from within the industry allows us to avoid the usual problems of confidentiality which could arise and thus enables us to supply new useful information (photos, architecture plans, real examples, etc.). Topics covered by the chapters of this book include SAET-METEOR, the B method and B tools, model-based design using Simulink, the Simulink design verifier proof tool, the implementation and applications of SCADE (Safety Critical Application Development Environment), GATEL: A V&V Platform for SCADE models and ControlBuild. Contents 1. From Classic Languages to Formal Methods, Jean-Louis Boulanger. 2. Formal Method in the Railway Sector & the First Complex Application: SAET-METEOR, Jean-Louis Boulanger. 3. The B Method and B Tools, Jean-Louis Boulanger. 4. Model-Based Design Using Simulink - Modeling, CodeGeneration, Verification, and Validation, Mirko Conrad and Pieter J. Mosterman. 5. Proving Global Properties with the Aid of the SIMULINK DESIGNVERIFIER Proof Tool, Véronique Delebarre and Jean-Frédéric Etienne. 6. SCADE: Implementation and Applications, Jean-Louis Camus. 7. GATEL: A V&V Platform for SCADE Models, Bruno Marre, Benjamin Bianc, Patricia Mouy and Christophe Junke. 8. ControlBuild, a Development Framework & for Control Engineering, Franck Corbier. 9. Conclusion, Jean-Louis Boulanger. This important and timely book contains vital information for all developers working with C, whether in high-integrity areas or not, who need to produce reliable and effective software. As the embedded world expands, developers must have a strong grasp of many complex topics in order to make faster, more efficient and more powerful microprocessors to meet the public's growing demand. Embedded Software: The Works covers all the key subjects embedded engineers need to understand in order to succeed, including Design and Development, Programming, Languages including C/C++, and UML, Real Time Operating Systems Considerations, Networking, and much more. New material on Linux, Android, and multi-core gives engineers the up-to-date practical know-how they need in order to succeed. Colin Walls draws upon his experience and insights from working in the industry, and covers the complete cycle of embedded software development: its design, development, management, debugging procedures, licensing, and reuse. For those new to the field, or for experienced engineers looking to expand their skills, Walls provides the reader with detailed tips and techniques, and rigorous explanations of technologies. Key features include: New chapters on Linux, Android, and multi-core - the cutting edge of embedded software development! Introductory roadmap guides readers through the book, providing a route through the separate chapters and showing how they are linked. About the Author Colin Walls has over twenty-five years experience in the electronics industry, largely dedicated to embedded software. A frequent presenter at conferences and seminars and author of numerous technical articles and two books on embedded software, he is a member of the marketing team of the Mentor Graphics Embedded Software Division. He writes a regular blog on the Mentor website (blogs.mentor.com/colinwalls). New chapters on Linux, Android, and multi-
core - the cutting edge of embedded software development! Introductory roadmap guides readers through the book, providing a route through the separate chapters and showing how they are linked. This book covers everything you need to know to write professional-level cryptographic code. This expanded, improved second edition includes about 100 pages of additional material as well as numerous improvements to the original text. The chapter about random number generation has been completely rewritten, and the latest cryptographic techniques are covered in detail. Furthermore, this book covers the recent improvements in primality testing. Explains in detail how to perform the most commonly used hazard analysis techniques with numerous examples of practical applications. Includes new chapters on Concepts of Hazard Recognition, Environmental Hazard Analysis, Process Hazard Analysis, Test Hazard Analysis, and Job Hazard Analysis. Updated text covers introduction, theory, and detailed description of many different hazard analysis techniques and explains in detail how to perform them as well as when and why to use each technique. Describes the components of a hazard and how to recognize them during an analysis. Contains detailed examples that apply the methodology to everyday problems. Get valuable hands-on experience with Swift 3, the latest version of Apple’s programming language. With this practical guide, skilled programmers with little or no knowledge of Apple development will learn how to code with Swift 3 by developing three complete, tightly linked versions of the Notes application for the OS X, iOS, and watchOS platforms. In the process, you’ll learn Swift’s fundamentals, including its syntax and features, along with the basics of the Cocoa, CocoaTouch, and WatchKit frameworks. This book teaches you how to use common design patterns for Swift, how to structure an application for Apple’s platforms, and how to submit working apps to the App Store. Divided into four distinct parts, this book includes: Swift 2 basics: Learn Swift’s basic building blocks and features for object-oriented development. OS X app development: Set up the document model, build out features, and sync data with iCloud. iOS app development: Use multimedia, contacts, location, notifications, and iCloud files to build a fully featured iOS Notes app. Advanced app extensions: Build an Apple Watch app, and learn how to debug, monitor, and test all three of your Swift apps. A benchmark text on software development and quantitative software engineering. "We all trust software. All too frequently, this trust is misplaced. Larry Bernstein has created and applied quantitative techniques to develop trustworthy software systems. He and C. M. Yuhas have organized this quantitative experience into a book of great value to make software trustworthy for all of us." -Barry Boehm.
students, engineers, professional developers, managers, and professionals involved in software engineering projects. An Instructor's Manual presenting detailed solutions to all the problems in the book is available from the Wiley editorial department. An Instructor Support FTP site is also available.

It's no secret that we are living in the Digital Age. Technology companies make up seven of the world's ten largest firms by market capitalization. And the key to their success is the key to all modern organizations. Jonathan Smart, business agility practitioner, thought leader, and coach, reveals the patterns and antipatterns that will help organizations from every industry deliver better value sooner, safer, and happier through high levels of engagement, inclusion, and empowerment.

Through his decades of experience in the technology world, Smart provides business leaders with a blueprint for creating a world-class organization of the future. Through Agile and Lean ways of working, business leaders can empower teams to improve production, grow together, and create better services for their customers. These better ways of working have overflowed from the IT department to every corner of successful organizations, taking root in every industry from aerospace to accounting, insurance to shipping.

This book is not about software development. It is not a book about the computer industry. This book is about applying agility across the entire organization. It's a book that will put you at the front of change and ahead of the competition.

Copyright code: 91e6a749ec096c7a149db23db7c2acb2