Debugging The Development Process Practical Strategies For Staying Focused Hitting Ship Dates And Building Solid Teams

The Pragmatic Programmer

This book is a broad discussion covering the entire software development lifecycle. It uses a comprehensive case study to address each topic and features the following: A description of the development, by the fictional companyCompanyName, of the DigitalHome(DH) System, a system with "smart" devices for controlling home lighting, temperature, humidity, small appliance power, and security. A set of scenarios that provide a realistic framework for the use of the DH System material Just-in-time training: each chapter includes mini tutorials introducing various software engineering topics that are discussed in that chapter and used in the case study. A case study exercises for students in software development practice, either individually or in a team environment. Offering a new approach to learning about software engineering theory and practice, the text is specifically designed to teach software engineering, using a comprehensive case study covering the complete software development lifecycle. Offer opportunities for students to actively learn about and engage in software engineering practice. Provide a realistic environment to study a wide array of software engineering topics including agile development Software Engineering Practice: A Case Study Approach supports a student-centered, "active" learning style of teaching. The DH case study exercises provide a variety of opportunities for students to engage in realistic activities related to the theory and practice of software engineering. The text uses a fictitious team of software engineers to portray the nature of software engineering and to depict what actual engineers do when practicing software engineering. All the DH case study exercises can be used as team or group exercises in collaborative learning. Many of the exercises have specific goals related to team building and teaming skills. The text also can be used to support the professional development of certification of practicing software engineers. The case study exercises can be integrated with presentations in a workshop or short course for professionals.

Dynamics of Software Development

If you're passionate about programming and want to get better at it, you've come to the right source. Code Craft author Pete Goodlife presents a collection of useful techniques and approaches to the art and craft of programming that will help boost your career and your well-being. Goodlife presents sound advice that he's learned in 15 years of professional programming. The book's standalone chapters span the range of a software developer's life: dealing with code, learning in-deads, and innovating performance on language or industry bias. Whether you're a seasoned developer, a neophyte professional, or a hobbyist, you'll find valuable tips in five independent categories: Code-level techniques for crafting lines of code, testing, debugging, and coping with complexity. Practices, approaches, and attitudes: keep it simple, collaborate well, reuse, and create reliable code. Tactics for learning effectively, behaving ethically, finding challenges, and avoiding stagnation. Practical ways to complete things: use the right tools, know what "done" looks like, and seek help from colleagues. Habits for working well with others, and pursuing development as a social activity.

Software Development

The rules of battle for tracking down -- and eliminating -- hardware and software bugs. When the pressure is on to root out an elusive software or hardware glitch, what's needed is a cool head courtesy of a set of rules guaranteed to work on any system, in any circumstance. Written in a frank but engaging style, Debugging provides simple, foolproof principles guaranteed to help find any bug quickly. This book makes those shelves of application-specific debugging books (on C++, Perl, Java, etc.) obsolete. It changes the way readers think about debugging, making those pesky problems suddenly much easier to find and fix. Illustrating the rules with real-life bug-detection war stories, the book shows readers how to: * Understand the system: how perceiving the "roadmap" can hasten your journey * Quit thinking and look: when hands-on investigation can't be avoided * Isolate critical factors: why changing one element at a time can be an essential tool * Keep an audit trail: how keeping a record of the debugging process can win the day. The rules of battle for tracking down -- and eliminating -- hardware and software bugs. When the pressure is on to root out an elusive software or hardware glitch, what's needed is a cool head courtesy of a set of rules guaranteed to work on any system, in any circumstance. Written in a frank but engaging style, Debugging provides simple, foolproof principles guaranteed to help find any bug quickly. This book makes those shelves of application-specific debugging books (on C++, Perl, Java, etc.) obsolete. It changes the way readers think about debugging, making those pesky problems suddenly much easier to find and fix. Illustrating the rules with real-life bug-detection war stories, the book shows readers how to: * Understand the system: how perceiving the "roadmap" can hasten your journey * Quit thinking and look: when hands-on investigation can't be avoided * Isolate critical factors: why changing one element at a time can be an essential tool * Keep an audit trail: how keeping a record of the debugging process can win the day.

Why Programs Fail

Use Windows debuggers throughout the development cycle-and build better software. Rethink your use of Windows debugging and tracing tools-and learn how to make them a key part of test-driven software development. As lead developer of one of the Windows Fundamentals Team at Microsoft, you'll apply expert debugging and tracing techniques--and sharpen your C++ and C# code analysis skills--through practical examples and common scenarios. Learn why experienced developers use debuggers in every step of the development process, and not just when bugs appear. Discover how to: Go behind the scenes to examine how powerful Windows debuggers work Catch bugs early in the development cycle with static and runtime analysis tools Gain practical strategies to tackle the most common code defects Apply expert tricks to handle user-mode and kernel-level bugs Implement techniques such as set breakpoints, inspect thread stacks, and debug the operating system Animate software behavior with Xperf and the Event Tracing for Windows (ETW) framework.
Today, even the largest development organizations are turning to agile methodologies, seeking major productivity and quality improvements. However, large-scale agile development is difficult, and publicly available case studies have been scarce. Now, these agile pioneers at Hewlett-Packard present a candid, start-to-finish insider’s look at how they’ve succeeded with agile in one of the company’s most mission-critical software environments: firmware for HP LaserJet printers. This book tells the story of an extraordinary experiment and journey. Could agile principles be applied to re-engineer an enormous legacy code base? Could agile enable both timely delivery and ongoing innovation? Could it really be applied to 400+ developers distributed across four states, three continents, and four business units? Could it go beyond delivering incremental gains to meet the stretch goal of 10x developer productivity improvements? It could, and it did—but getting there was not easy. Writing for both managers and technologists, the authors candidly discuss both their successes and failures, presenting actionable lessons for other development organizations, as well as approaches that have proven themselves repeatedly in HP’s challenging environment. They not only illustrate the potential benefits of agile in large-scale development, they also systematically show how these benefits can actually be achieved. Coverage includes: • Clearly illustrating why and how to avoid those errors. Software Development and Professional Practice covers many of the topics described for the ACM Computing Curricula 2001 course C620 Software Development and Professional Practice. It is designed to be both a textbook and a manual for the working professional.

Software Development, Design and Coding

Software Development and Professional Practice reveals how to design and code great software. What factors do you take into account? What makes a good design? What methodologies and processes are out there for designing software? Is designing small programs different than designing large ones? How can you tell a good design from a bad one? You’ll learn the principles of good software design, and how to use those principles to create actual code. Software Development and Professional Practice is also about code construction—how to write great programs and make them work. What, you say? You’ve already written eight gazillion programs! Of course I know how to write code! Well, in this book you’ll re-examine what you already do, and you’ll investigate ways to improve. Using the Java language, you’ll look deeply into coding standards, debugging, unit testing, modularity, and other characteristics of good programming. You’ll also talk about reading code. How do you read code? What makes a program readable? Can good, readable code replace documentation? How much documentation do you really need? This book introduces you to software engineering—the application of engineering principles to the development of software. What are these engineering principles? First, all engineering efforts follow a defined process. So, you’ll be spending a bit of time talking about how you run a software development project and the different phases of a project. Secondly, all engineering work has a basis in the application of science and mathematics to real-world problems. And so does software development! You’ll therefore take the time to examine how to design and implement programs that solve specific problems. Finally, this book is also about human-computer interaction and user interface design issues. A poor user interface can ruin any desire to actually use a program; in this book, you’ll figure out why and how to avoid those errors. Software Development and Professional Practice covers many of the topics described for the ACM Computing Curricula 2001 course C620 Software Development and Professional Practice. It is designed to be both a textbook and a manual for the working professional.

Advanced Windows Debugging

Corporate and commercial software-development teams all want solutions for one important problem—how to get their high-pressure development schedules under control. In RAPID DEVELOPMENT, author Steve McConnell addresses that concern head-on with overall strategies, specific best practices, and valuable tips that help shrink and control development schedules and keep projects moving. Inside, you’ll find: A rapid-development strategy that can be applied to any project and the best practices to make that strategy work Candid discussions of great and not-so-great rapid-development practices—estimation, project planning, time management, risk management, and more A list of classic mistakes to avoid for rapid-development projects, including creeping requirements, cost overruns, and poor quality software The real-world guide to more efficient applications development.

Creating Games

Sponsored by the Association for Educational Communications and Technology (AECT), this book presents a definition of the field of study and practice known as educational technology or instructional technology. It reflects the collaborative efforts of all members of the AECT Definition and Terminology Committee. The volume begins with the statement of the definition itself (chapter 1), followed by commentary chapters on each of the key terms and concepts contained in the definition (chapters 2-9). Chapter 10 provides historical context for the current definition by reviewing salient elements of prior AECT definitions. Practical Considerations and chapter 12 concludes by discussing ramifications of the current definition for academic programs in educational technology. This book is appropriate for anyone working in the field of educational technology: students, instructors, researchers and in-service providers.


The first In-Depth, Real-World, Insider’s Guide to Powerful Windows Debugging For Windows developers, few tasks are more challenging than debugging—or more crucial. Reliable and realistic information about Windows debugging has always been scarce. Windows debuggers and products are usually too complicated to learn. In RAPID DEVELOPMENT, authors candidly discuss both their successes and failures, presenting actionable lessons for other development organizations, as well as approaches that have proven themselves repeatedly in HP’s challenging environment. They not only illustrate the potential benefits of agile in large-scale development, they also systematically show how these benefits can actually be achieved. Coverage includes: • Clearly illustrating why and how to avoid those errors. Software Development and Professional Practice covers many of the topics described for the ACM Computing Curricula 2001 course C620 Software Development and Professional Practice. It is designed to be both a textbook and a manual for the working professional.

The Practice of Programming

This fully updated second edition includes 100+ pages of new material, including new chapters on Verifying Code, Predicting Errors, and Preventing Errors. Cutting-edge tools such as FindBugs and AGITAR are explained, techniques from integrated environments like Jazz.net are highlighted, and all-new demos with Eclipse and SpeeD. Eclipse and Mozilla are included. This complete and pragmatic overview of debugging is authored by Andrew Zeller, the talented researcher who developed the GNU Data Display Debugger (DDD), a tool that over 250,000 professionals use to visualize the data structures of programs while they are running. Unlike other books on debugging, Zeller’s text is product agnostic, appropriate for all programming languages and skill levels. Why Programs Fail explains best practices ranging from systematically tracking error reports, to observing symptoms, reproducing errors, and correcting defects. It covers a wide range of tools and techniques from hands-on observation to fully automated diagnostics, and also explains cutting-edge techniques for integrating minimal input to reproduce an error and for tracking causes and effect through a program. It even includes instructions on how to create automated debugging tools. The new edition of this award-winning productivity-booster is for any developer who has ever been frustrated by elusive bugs. Brand new chapters demonstrate cutting-edge debugging techniques and tools, enabling readers to put the latest time-saving developments to work for them. Learn by doing. New exercises and detailed examples focus on emerging tools, languages and environments, including AGITAR, FindBugs, Python and Eclipse. The text includes exercises and extensive references for further study, and a companion website with source code for all examples and additional debugging resources.
Debugging the Development Process

An irreverent look at how Visual FoxPro really works. Tells you the inside scoop on every command, function, property, event and method of Visual FoxPro 7.0. The eagerly awaited revision to the Hacker's Guide for Visual FoxPro 6.0, this completely updated book is the one you'll keep by your side for as long as you develop in Visual FoxPro.

Product Design and Factory Development

Deep Learning and Practice with MindSpore

Looks at a successful software project and provides details for software development for clients using object-oriented design and programming.

Software Project Survival Guide


Software Engineering Practice

The ability to solve difficult problems is what makes a good engineer great. This book teaches techniques and tools for developers to tackle even the most persistent bugs. You'll find that tough issues can be made simple with the right knowledge, tools, and practices. Practical Debugging for .NET Developers will transform you into the guy or gal who everyone turns to for help. Issues covered include .NET Core, C#, Memory Leaks, Performance Problems, ASP.NET, Performance Counters, ETW Events, Production Debugging, Memory Pressure, Visual Studio, Hangs, Profiling, Deadlocks, Crashes, Memory Dumps, and Azure. * Discover the best tools in the industry to diagnose and fix problems * Learn advanced debugging techniques with Visual Studio * Fix memory leaks and memory pressure issues * Detect, profile, and fix performance problems * Find the root cause of crashes and hangs * Debug production code and third-party code * Analyze ASP.NET applications for slow performance, failed requests, and hangs * Use dump files, Performance Counters, and ETW events to investigate what happens under the hood * Troubleshoot cloud environments, including Azure VMs and App Services * Code samples in C# * Covering .NET Core, .NET Framework, Windows, and Linux

Debugging

The author explains how he organized and supervised effective software development teams at the Microsoft company to come up with timely and high-quality commercial applications, offering a candid look at the group dynamics of software development. Original. (Advanced).

Code Craft

Opening moves; The organization; The competition; The customer; The design; Development; The middle game; Ship mode; The launch; Appendix; Index.

Inside Visual C++

Addresses the creation of bug-free Perl code while explaining how to trace code, test programs, operate the debugger, locate and correct syntax errors, and deal with run-time exceptions and semantical errors.

Debugging the Development Process

This book offers a comprehensive practical guide to SAP ABAP for ABAP professionals. Part I of this two-part series lays the groundwork with ABAP basics. Readers will learn fundamental methods and procedures for everyday ABAP use— for example, how to download files from SAP directories to workstations. Dive into the SAP Data Dictionary and how it works. Get detailed information on effective debugging techniques and how to use the SAP Debugger. Clarify when it is best to use standard SAP tables vs. X-tables. Get expert developer tips and tricks including how to navigate AUI grid lists. Understand the documentation programs available to you and how to use them. Obtain useful reference lists of SAP transactions and SAP database tables. By using practical examples, tips, and screenshots, the author brings readers up to speed on the fundamentals of SAP ABAP. - How to get the most out of SAP ABAP - Guide for understanding and using the SAP Data Dictionary - Beginner and advanced debugging techniques - Expert ABAP development techniques

Hacker's Guide to Visual FoxPro 7.0

Provides information on the techniques of debugging software and code.

Practical Debugging for .NET Developers

In Debugging the Development Process, Maguire describes the sometimes controversial but always effective practices that enabled his software teams at Microsoft to develop high-quality software - on schedule. With the refreshing candor reviewers admired in Writing Solid Code, Maguire talks about what did and what didn't work at Microsoft and tells you how to energize software teams to work effectively - and to enjoy their work; why you might want to kick your star programmer off your team; how to avoid corporate snarls and overblown corporate processes; which tiny changes produce major results; how to deliver on schedule and without overtime; how to pull twice the value out of everything you do; how to get your team going on a creative roll; and how to raise the average programmer level at your company.

Rapid Development

Widely considered one of the best practical guides to programming, Steve McConnell's original CODE COMPLETE has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices—and hundreds of new code samples—illustrating the art and science of software construction. Capturing the body of knowledge available from research,
Effective Debugging

Creating Games offers a comprehensive overview of the technology, content, and mechanics of game design. It emphasizes the broad view of a games team and teaches you enough about your teammates’ areas so that you can work effectively with them. The authors have included many worksheets and exercises to help you build a small indie team off the ground. Special features: Exercises at the end of each chapter combine comprehension tests with problems that help the reader interact with the material. Worksheet exercises provide creative activities to help project teams generate new ideas and then structure them in a modified version of the format of a game industry design document. Pointers to the best resources for digging deeper into each specialized area of game development. Website with guidelines, figures from the book, and teaching materials that include study guides, lecture presentations, syllabi, supplemental exercises, and assessment materials.

Debug It!

To understand the principles and practice of software development, there is no better motivator than participating in a software project with real-world value and a life beyond the academic arena. Software Development: An Open Source Approach immerses students directly into an agile free and open source software (FOSS) development process. It focuses on teaching the development process used in multicore software development. It is written with a focus on solving day to day problems using practical tips and tricks and industry case studies to reinforce the key concepts in multicore software development. Coverage includes: The multicore landscape. Principles of parallel computing. Multicore SoC architectures. Multicore programming models. The multicore development process. Multicore programming with threads. Concurrency abstraction layers. Debugging Multicore Systems. Practical techniques for getting started in multicore development. Case studies in multicore development. Sample code to reinforce many of the concepts discussed. Presents the ‘nuts and bolts’ of programming a multicore system. Provides a short-format book on the practical processes and techniques used in multicore software development. Covers practical tips, tricks, and industry case studies to enhance the learning process.

A Practical Approach to Large-scale Agile Development

This book systematically introduces readers to the theory of deep learning and explores its practical applications based on the MindPore AI computing framework. Divided into 14 chapters, the book covers deep learning, deep neural networks (DNNs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), unsupervised learning, deep reinforcement learning, automated machine learning, device-cloud collaboration, deep learning visualization, and data preparation for deep learning. To help clarify the complex topics discussed, this book includes numerous examples and links to online resources.

Extreme Programming Installed

Learn the principles of good software design, and how to turn those principles into great code. This book introduces you to software engineering — from the application of engineering principles to the development of software. You’ll see how to run a software development project, examine the different phases of a project, and learn how to design and implement programs that solve specific problems. It’s also about how to build great programs and make them work. Whether you’re new to programming or have written hundreds of applications, in this book you’ll re-examine what you already do, and you’ll investigate ways to improve. Using the Java language, you’ll look deeply into coding standards, debugging, unit testing, modularity, and other characteristics of good programs. With Software Development, Design, and Coding, author and professor John Dooley distills his years of teaching and development experience to demonstrate practical techniques for great coding. You’ll learn about agile methodologies including Scrum and Lean programming, leverage the capabilities of modern computer systems with parallel programming, and develop project-based patterns to exploit application development best practices. Use modern tools for development, collaboration, and source code control. Who This Book Is For Early career software developers, or upper-level students in software engineering courses.
Code Complete

This book offers a comprehensive practical guide to SAP ABAP for ABAP professionals. Part I of this two-part series lays the groundwork with ABAP basics. Readers will learn fundamental methods and procedures for everyday ABAP use—example, how to download files from SAP directories to workstations. Dive into the SAP Data Dictionary and how it works. Get detailed information on effective debugging techniques and how to use the SAP Debugger. Clarify when it is best to use standard SAP tables vs. R-tables. Get expert developer tips and tricks including how to navigate ALV grid lists. Understand the document management processes available to you and how to use them. Obtain useful program listings of SAP transactions and SAP database tables. By using practical examples, tips, and screen shots, the author brings readers up to speed on the fundamentals of SAP ABAP.

- How to get the most out of SAP ABAP — Guide for understanding and using the SAP Data Dictionary — Beginner and advanced debugging techniques — Expert ABAP development techniques

Educational Technology

Provides techniques and methods to design, deploy and manage an Exchange Server operation. Describes exchange " best practices" developed by author and his colleagues at Compaq.

Software Development and Professional Practice

Every software developer and IT professional understands the crucial importance of effective debugging. Often, debugging consumes most of a developer’s workday, and mastering the required techniques and skills can be a daunting task. In Effective Debugging, Diamidis Spinellis helps experienced programmers accelerate their journey to mastery, by systematically categorizing, explaining, and illustrating the most useful debugging methods, strategies, techniques, and tools. Drawing on more than thirty-five years of experience, Spinellis expands your arsenal of debugging techniques, helping you choose the best approaches for each challenge. He presents vendor-neutral, example-rich advice on general principles, high-level strategies, concrete techniques, high-efficiency tools, creative tricks, and the behavioral traits associated with effective debugging. Spinellis’s 66 expert techniques address every facet of debugging and are illustrated with step-by-step instructions and actual code. He addresses the full spectrum of problems that can arise in modern software systems, especially problems caused by complex interactions among components and services running on hosts scattered around the planet. Whether you’re debugging isolated runtime errors or catastrophic enterprise system failures, this guide will help you get the job done more quickly, and with less pain. Key features include High-level strategies and methods for addressing diverse software failures Specific techniques to apply when programming, compiling, and running code Better ways to make the most of your debugger General-purpose techniques and tools for debugging multithreaded, asynchronous, and embedded systems Bug avoidance thanks to improved software design, construction, and management

Practical Guide to SAP ABAP

What others in the trenches say about The Pragmatic Programmer "The cool thing about this book is that it’s great for keeping the programming process fresh. The book helps you to continue to grow and clearly comes from people who have been there." —Kent Beck, author of Extreme Programming Explained: Embrace Change "I found this book to be a great mix of solid advice and wonderful analogies!" —Martin Fowler, author of Refactoring: improving understandable code "I would buy a copy, read it twice, then tell all my colleagues to run out and grab a copy. This is a book I would never loan because I would worry about it being lost." —Kevin Runland, Management Science, MSc Logistics "The wisdom and practical experience of the authors is obvious. The topics presented are relevant and useful. By far its greatest strength for me has been evidence of analogical reasoning built on analogical thinking, and the fabulous helicopter-like illustration of the need for orthogonality, especially in a crisis situation. I have no doubt that this book will eventually become an excellent source of useful information for journalism professionals and expert mentors alike." —John Lackey, author of Large-Scale C++ Software Design "This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients." —Eric Vought, Software Engineer "Most modern books on software development fail to cover the basics of what makes a great software developer, instead spending their time on syntax or technology where in reality the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book." —Jeta McLean, Independent Consultant "Since reading this book, I have implemented many of the practical suggestions and tips it contains. Across the board, they have saved my company time and money while helping me get my job done quicker! This should be a desktop reference for everyone who works with code for a living." —Jared Richardson, Senior Software Developer, iRenaissance, Inc. "I would like to thank you every single day for the help you’ve given to my company." —Joe Kurata, President of Simplus, Inc. "I was impressed by the quality of the documentation you create." —Jared Richardson, President of Simplus, Inc. "I read this book a few weeks ago and in general I was impressed by the quality of the documentation you create." —Joe Kurata, President of Simplus, Inc.

Software Engineering Classics

Doing Objects in Visual Basic 2005 is the authoritative guide to object-oriented design, architecture, and development with Visual Basic 2005. Author Deborah Kurata is the original pioneer in building object-oriented applications with Visual Basic. In this book she continues to offer clarity and deliver best practices for using object-oriented techniques in Visual Basic 2005. She has been honored with Microsoft’s prestigious MVP designation for her expertise and contributions to the community. Kurata begins with a concise introduction to core object-oriented concepts and the Visual Basic 2005 features that support them. Next she introduces a pragmatic and agile approach to designing effective applications along with an application framework. From there she walks you through the process of building the user interface, business logic, and data access layers of an application, highlighting key VB 2005 techniques and best practices. Kurata’s step-by-step “building along” activities provide you with deep hands-on mastery; your finished application can serve as the starting point for virtually any custom project. This book Show how to leverage Visual Studio 2005 combined with a solid object-oriented approach can help minimize the complexity of software development and improve productivity. Kurata’s comprehensive object-oriented software development methodology to help analyze and design applications for the real world Covers building the user interface layer using a base form class, programmatic interfaces, and object binding Details building the business logic layer using a base business object class and validation rules Demonstrates how to build the data access layer using ADO.NET Provides best practices and tips for experienced .NET developers, those new to .NET, and for those developers moving from VB6 to .NET

Doing Objects in Visual Basic 2005

Offers application debugging techniques for Microsoft .NET Framework and Windows, covering topics such as exception monitoring, crash handlers, and multithreaded deadlocks.
Recent Advances and Issues in Computers
A guide to writing computer code covers such topics as variable naming, presentation style, error handling, and security.

Perl Debugged
Extreme Programming Installed explains the core principles of Extreme Programming and details each step in the XP development cycle. This book conveys the essence of the XP approach—techniques for implementation, obstacles likely to be encountered, and experience-based advice for successful execution.

Debugging Teams
In the course of their 20+-year engineering careers, authors Brian Fitzpatrick and Ben Collins-Sussman have picked up a treasure trove of wisdom and anecdotes about how successful teams work together. Their conclusion? Even among people who have spent decades learning the technical side of their jobs, most haven’t really focused on the human component. Learning to collaborate is just as important to success. If you invest in the “soft skills” of your job, you can have a much greater impact for the same amount of effort. The authors share their insights on how to lead a team effectively, navigate an organization, and build a healthy relationship with the users of your software. This is valuable information from two respected software engineers whose popular series of talks—including “Working with Poisonous People”—has attracted hundreds of thousands of followers.

Multicore Software Development Techniques
Debugging by Thinking: A Multi-Disciplinary Approach is the first book to apply the wisdom of six disciplines—logic, mathematics, psychology, safety analysis, computer science, and engineering—to the problem of debugging. It uses the methods of literary detectives such as Sherlock Holmes, the techniques of mathematical problem solving, the results of research into the cognitive psychology of human error, the root cause analyses of safety experts, the compiler analyses of computer science, and the processes of modern engineering to define a systematic approach to identifying and correcting software errors. * Language Independent Methods: Examples are given in Java and C++ * Complete source code shows actual bugs, rather than contrived examples * Examples are accessible with no more knowledge than a course in Data Structures and Algorithms requires * A “thought process diary” shows how the author actually resolved the problems as they occurred

Microsoft Exchange Server for Windows 2000
Describes the latest developments and innovations in computer science research and technology.