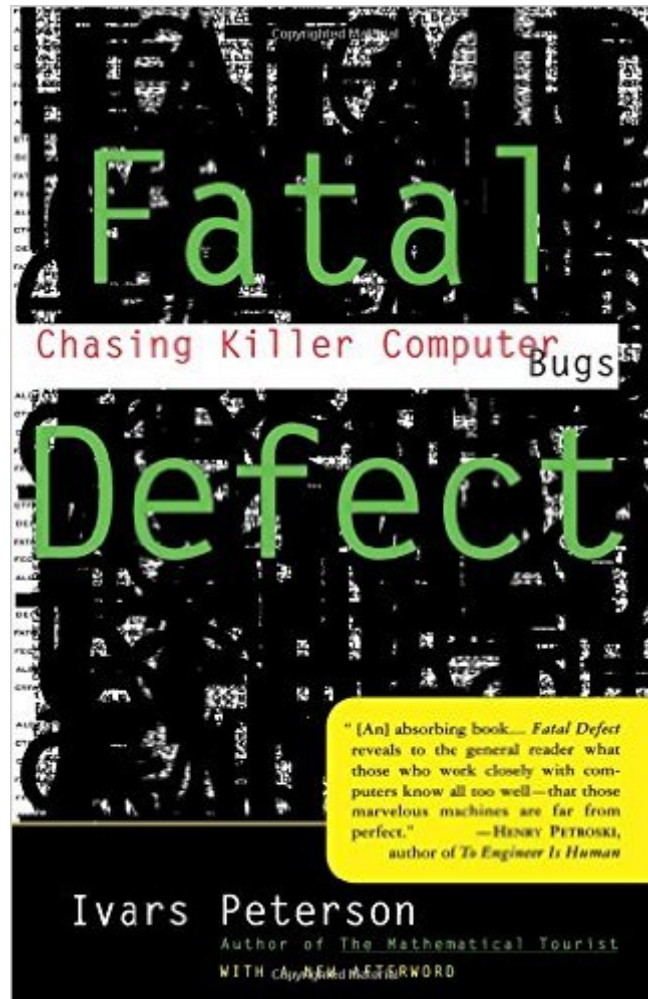


The book was found

Fatal Defect: Chasing Killer Computer Bugs



Synopsis

An airplane crashes, killing eighty-seven passengers. A cancer patient receives a fatal dose of radiation from a machine designed to be foolproof. The ATMs at a New York bank debit customers twice their actual withdrawals, resulting in a loss of millions of dollars. In every case, the culprit was a computer bug, a software error or design defect that may escape detection until it erupts into the real world with sometimes catastrophic results. This arresting and at times terrifying book tells us just how prevalent these defects are and how they are multiplying as computers become more sophisticated and more deeply embedded in our daily lives. It is also a riveting portrait of the men and women who find and "exterminate" those bugs, whether they occur in pocket calculators or nuclear reactors. *Fatal Defect* reveals what you should know about the computers in our lives. Read it before you buy a computer, use a cash machine, or book an airplane flight. Then pray that one of its real-life heroes was on the job.

Book Information

Paperback: 288 pages

Publisher: Vintage; Reprint edition (April 30, 1996)

Language: English

ISBN-10: 0679740279

ISBN-13: 978-0679740278

Product Dimensions: 5.5 x 0.6 x 8.5 inches

Shipping Weight: 13.8 ounces (View shipping rates and policies)

Average Customer Review: 4.8 out of 5 stars [See all reviews](#) (5 customer reviews)

Best Sellers Rank: #1,592,217 in Books (See Top 100 in Books) #37 in [Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Quality Control](#) #74 in [Books > Computers & Technology > Programming > Languages & Tools > Debugging](#) #28356 in [Books > Science & Math > Mathematics](#)

Customer Reviews

Henry Petroski has written several books that explore his dictum, "Form follows failure." His thesis is that improvements in engineering are made to overcome the failures of previous design. Petroski's books cover advances in civil and mechanical engineering. Ivars Petersen has written a similar book covering some of the notorious failures in software engineering and the efforts by a few leading engineers to define practices and design methods that can prevent such failures from recurring. *Fatal Defect* describes dozens of software failures, how they happened, and the efforts to

correct them. The defects occur in banking systems, stock exchange mechanisms, aircraft and spacecraft guidance computers, medical equipment, telecommunications, and scientific computation. Some of these failures are famous; others are little known. Regardless, the descriptions always provide the kind of technical detail that you need to really appreciate the situation. Petersen is a journalist for science news and is clearly a professional when it comes to describing technical issues for the intelligent layman. Moreover, he tells the stories of people who found the errors, lead the efforts to correct them, or who tried to raise the standards of the industry. Nancy Leveson investigated the Therac-25 defect that led to several deaths in 1986. This influenced her efforts to design software safety standards. Learning from failures requires knowing about them. But the details of many failures are often kept quiet, being marked proprietary or secret to avoid embarrassment or litigation. Peter Neumann tried to open up the discussion of computer failures with RISK digest. He started it in 1985 but even today it remains one of the best places to learn about the technical details behind dangerous system failures. David Parnas took the lead in criticizing the Star Wars strategic defense initiative. He noted that there would inevitably be defects in the software and that there was no way to conduct a comprehensive system test, short of a nuclear war. He then moved on to overseeing the engineering processes at the Darlington nuclear plant, ensuring that the software was correct, even though this delayed the project completion by three years. Vic Basili was one of the first to conduct controlled experiments with programming teams to determine which development methods actually produced the most reliable software. For example, in 1982, he established that code reviews were far more effective than functional testing, a result that is only beginning to be regularly applied to engineering practice today. These results led him and Harlan Mills to develop the Cleanroom process which Mills taught at NASA and IBM. Petersen tells the stories of these and other engineers, describing their background and how their careers led them on the paths that they ended up. I'd been familiar with the ideas of many of these people, but I found it very interesting to learn of the experiences that had led them to formulate and articulate these ideas. The issue of what constitutes effective means for developing reliable software is becoming more than just a practical matter with recent events. Earlier this year the Texas board of professional engineers started licensing software engineers and the Institute for Electrical and Electronics Engineers plans to start certifying software engineers in 2000. Licensing means more than just professionalism and status. It also means acknowledging accepted practice and deviating from it at the risk of malpractice. If the licensing process is done well, it will base itself on the fine, but tentative work done by the people described in this book. If it is done poorly, it may merely enshrine the latest fad in law.

From the subtitle, "Chasing Killer Computer Bugs," you would think that this would be a book about software testing. It isn't. Rather, it's a book about things going wrong with software. The author tells readable stories about some infamous software bugs, at least one of which I've never been able to forget. This is the story of the Therac-25 radiation therapy machine, which caused deaths and injuries due to a software problem. Some of these stories have morals which software professionals would do well to keep in mind--and not just developers, but the people who give them their marching orders. For instance, the Therac-25 story makes clear in a very sobering way how an apparently VERY minor change to a program can have VERY unanticipated consequences. The A320 story makes a convincing case for thinking hard about user interface design. Other parts of the book talk about why building good software can be so hard, and about some of the people and organizations that work towards developing approaches to issues in software quality and construction. You wouldn't think that these would be particularly interesting subjects, but for the most part the author makes them come alive. This is not a technical book--don't expect to come away from it with any new debugging techniques. Rather, expect it to give you lots of food for thought.

Had to read this for a college course. Even given the fast pace of change in technologies, this book remains relevant.

Ivars Peterson has written a well thought-out and interesting book that provides just the right amount of depth to this fascinating subject. Both the layman and the software engineer will find Fatal Defect interesting reading. This book should be required reading for everybody in the IT industry!

Good layman's level overview of the technical problems with critical control via software and a discussion of how industry and individual experts are trying to deal with the problem

[Download to continue reading...](#)

Fatal Defect: Chasing Killer Computer Bugs Hacking: Computer Hacking: The Essential Hacking Guide for Beginners, Everything You need to know about Hacking, Computer Hacking, and Security ... Bugs, Security Breach, how to hack) How Many Bugs in a Box?: A Pop-up Counting Book (David Carter's Bugs) Chasing Lincoln's Killer Toward Zero Defect Programming The Practical Guide to Defect Prevention (Developer Best Practices) The Practical Guide to Defect Prevention (Best Practices) Local Flaps in Facial Reconstruction: A Defect Based Approach Construction Defect

Claims: Handbook for Insurance, Risk Management, Construction/Design Professionals Killer Pies: Delicious Recipes from North America's Favorite Restaurants (Killer (Chronicle Books)) Confession of a Serial Killer: The Untold Story of Dennis Rader, the BTK Killer The Great Zodiac Killer Hoax of 1969 (The Great Zodiac Killer Hoax series Book 2) Python: Python Programming For Beginners - The Comprehensive Guide To Python Programming: Computer Programming, Computer Language, Computer Science Python: Python Programming For Beginners - The Comprehensive Guide To Python Programming: Computer Programming, Computer Language, Computer Science (Machine Language) The Computer Insectary: A Field Guide to Viruses, Bugs, Worms, Trojan Horses, and Other Stuff That Will Eat Your Programs and Rot Your Brain Fatal Descent: Andreas Lubitz and the Crash of Germanwings Flight 9525 Blood Brothers: The Fatal Friendship Between Muhammad Ali and Malcolm X Fatal Invention: How Science, Politics, and Big Business Re-create Race in the Twenty-first Century Fatal Judgment (Guardians of Justice, Book 1) (Volume 1) The Grizzly Maze: Timothy Treadwell's Fatal Obsession with Alaskan Bears

[Dmca](#)