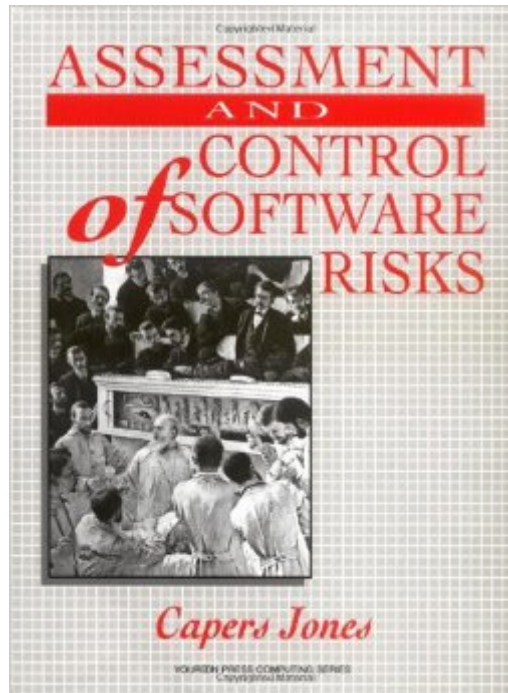


The book was found

# Assessment And Control Of Software Risks



## Synopsis

This handbook summarizes more than 50 of the major problems of building and maintaining software projects, and outlines the prevention/control âœœtherapiesâ • available. KEY TOPICS: Considers in depth the software-related risks in the domains of methodologies, tools, organization structures, skills and specialization, client relations, and sociological issues. For software managers and software professionals in software engineering, software quality assurance, and related software areas.

## Book Information

Hardcover: 464 pages

Publisher: Prentice Hall; 1 edition (December 17, 1993)

Language: English

ISBN-10: 0137414064

ISBN-13: 978-0137414062

Product Dimensions: 7 x 1.4 x 9 inches

Shipping Weight: 2.3 pounds (View shipping rates and policies)

Average Customer Review: 4.6 out of 5 starsÂ Â See all reviewsÂ (5 customer reviews)

Best Sellers Rank: #2,561,047 in Books (See Top 100 in Books) #58 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Quality Control #3043 inÂ Books > Textbooks > Computer Science > Software Design & Engineering #6635 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development

## Customer Reviews

The author proposes that we categorize software development risks like the Center for Disease Control categorizes disease. He builds on this theme by discussing risk from the perspective of symptoms, susceptibility, prevention, and therapies. The information is excellent., and is not available in this compact format from any other source (at least that I know of). This book is a 'must' for any manager intent on improving the system development life cycle. /// This book is primarily directed toward knowledgeable software professionals. It assumes you have a good understanding of development theory. Unfortunately, since the book was written in 1993, some information (related to vendor tools and products) is a little long in the tooth. That is the only reason I didn't give the book a rating of '10'. /// I would love to see a yearly update that included new symptoms, new 'diseases', additional therapies, and updated lists of vendor 'cures'. Overall content is first-rate and

very relevant, not theoretical. The hard part -- detailed therapy -- is properly left to the reader, but several options are presented

The book is a great resource for brainstorming potential risks to your projects and strategies to handle them. Almost all are timeless. However, the book is 10 years old and much has changed over the last 10 years. The Internet is a prime example. This new medium has most of the risks of legacy systems, but the Internet brings new possibilities and new expectations.

While an update to this book would be nice, the fact is that the major risks cited in this 1994 classic continue to be risks in 2002. What I like about this book is the way it's organized and structured. Jones starts with two chapters that highlight the most common and the most serious software risks. In that respect the essence of the book is provided up front. The rest of the book is a catalog of the top 43 risks, presented in a quasi-pattern format (that predates the GoF patterns movement), using a medical metaphor, based on the US Public Health Service publication titled "Control of Communicable Diseases in Man" as the pattern. The format is highly effective and intuitive, and also plants a subtle notion that the risk can be cured. The risks are arranged alphabetically for easy reference and range from Artificial Maturity Levels to Slow Technology Transfer, with the usual suspects that have plagued software engineering since its inception: corporate politics, excessive schedule pressure, low user satisfaction, malpractice (project management and technical staff), silver bullet syndrome ... sound familiar? In my opinion, until cures are found for the risks cited in the book we, as a profession, don't need an updated list of more. I strongly recommend this book to anyone who manages software engineering projects or processes. If you want a more condensed version of this book get Jones' "Software Systems Failure and Success", which was published in 1996. That book distills critical success (or failure, depending on your perspective) factors into a manageable group of twelve attributes. It doesn't fully replace this book, but does augment it well.

This book is quite complete for anyone needing to deal with software risks and its organisation makes it particularly handy. It is a marvelous reference book when tackling complex projects, re assessing risks or trying to find out what went wrong. As a bonus it brings broad references of all resources that can be leveraged on each specific risk.

If nothing else read the chapters on "Malpractice (Management)" and "Excessive Schedule Pressure". This book should be required reading for any project management.

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

Assessment and Control of Software Risks Control Self-Assessment: Reengineering Internal Control (Enterprise Governance, Control, Audit, Security, Risk Management and Business Continuity) Calculated Risks: The Toxicity and Human Health Risks of Chemicals in our Environment Global Health Risks: Mortality and Burden of Disease Attributable to Selected Major Risks The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities (2 Volume set) Control Your ERP Destiny: Reduce Project Costs, Mitigate Risks, and Design Better Business Solutions Sharing Environmental Risks: How to Control Governments' Losses in Natural Disasters (Westview Special Studies in Public Policy and Public Systems Management) Insuring the Bottom Line: How to Protect Your Company From Liabilities, Catastrophes and Other Business Risks First Edition (Taking Control) Measuring the Software Process: Statistical Process Control for Software Process Improvement Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for Software Protection Software Engineering Classics: Software Project Survival Guide/ Debugging the Development Process/ Dynamics of Software Development (Programming/General) Pocket Companion for Physical Examination and Health Assessment, 6e (Jarvis, Pocket Companion for Physical Examination and Health Assessment) Advanced Health Assessment of Women, Third Edition: Clinical Skills and Procedures (Advanced Health Assessment of Women: Clinical Skills and Pro) Rabbit Medicine and Surgery: Self-Assessment Color Review, Second Edition (Veterinary Self-Assessment Color Review Series) ParaPro Assessment Secrets Study Guide: ParaProfessional Test Review for the ParaPro Assessment Orthopedic Physical Assessment, 5e (Orthopedic Physical Assessment (Magee)) Software Assessment: Reliability, Safety, Testability (New Dimensions In Engineering Series) Guide for Reusable Software: Assessment Criteria for Aerospace Applications Yeast Control in Seven Days - How to Rebuild Health and Control Candida Infection Control and Management of Hazardous Materials for the Dental Team, 3e (INFECTION CONTROL & MGT/ HAZARDOUS MAT/ DENTAL TEAM ( MILLER))

[Dmca](#)