Increasingly, system functionality is implemented in software. Where we used to split hardware from software, the business case entirely determines such boundaries now—what we best package at which level in which component, be it software or silicon. For example, a TV set in the 1970s had no software, whereas today its competitive advantages and the majority of engineering efforts are software-driven.

The software business, however, has manifold challenges, ranging from the creation process and its inherent risks to direct balance sheet impacts. For example, the Standish Group found in its survey (2000 edition of the Chaos Report) that only 26 percent of the projects finished on time and within budget and a staggering 28 percent were canceled before de-
livery. Moreover, the remaining projects, which all finished late, over budget or both, delivered only a fraction of the planned functionality (www.standishgroup.com).

Introducing a product to market late loses market share; canceling a product before it ever reaches the market sinks scarce R&D funds. Not only is software increasing in size, complexity, and percentage of functionality, it is increasing in contribution to the balance sheet and profit-and-loss statements. To make matters worse, requirements are easily and frequently changed. A recent study by the U S National Institute of Standards and Technology reports that insufficient software testing costs the US as much as US$9 billion a year and that up to US$22 billion of that could be saved if licensed software had just 50 percent fewer defects (NIST, *The Economic Impacts of Inadequate Infrastructure for Software Testing*, Washington, D.C., 2002; see also the News report on p. 97).

This special issue is devoted to the business of software engineering. We explore some of the critical factors associated with succeeding in today’s high-tech software businesses and discuss skills, knowledge, and abilities that software practitioners need to improve their business decision-making capabilities. We illustrate business cases to supplement technical arguments for process and technology improvement. We also address how software engineering can help small and large businesses as well as start-ups.

Every business needs good communication to be successful and reduce friction, whether it is from engineer to manager, manager to engineer, or engineer to engineer. It is easy for a company to relegate software to a low priority when it is focusing on other technologies in its products. Software engineers must speak out clearly and be heard and understood by management. Both sides must learn how to address each other’s real needs. Management doesn’t care for technical jargon, and engineers are easily confused with capitalization and depreciation questions about their software. The brief article on translating “software developer speak” to “management speak” and vice versa can help here.

We must consistently set targets on each level in the company and track them continuously against actual performance. We must manage changes in the business climate, or they will ripple through uncontrolled. And the committed targets must be followed through! We therefore present the Balanced Scorecard approach. Each company and software team can introduce this approach to focus on the right things and balance short-term (survival) needs with medium- to long-term investments and improvements.

As the saying goes, time is money; therefore, wasted development time is lost revenue. Thus, processes and methods that improve our ability to deliver reliable, quality software are important.

Security is a quickly growing software business. Recent examples for business models include the sale of hacker insurance, which works to keep corporate Web sites from defacement or denial-of-service attacks by hackers and protects databases, such as those maintaining credit card information. The more we share and network, the more we are exposed to attacks of all kinds. The exploding need for secure software and protection schemes for our business processes, end-to-end, indicate this impact. Our Point/Counterpoint discussion takes up one example from the security domain and illustrates two ways to approach software security and how that decision ripples into business decisions.

We have not explicitly addressed the “technical career track” versus the “management ladder.” We believe that such discussions are individual choices. However, we do hope that all software practitioners value their stake in their respective software business decisions and that these pages offer ideas for increasing your return on investment in that business. The “Suggested Reading” sidebar offers more food for thought.

Whether your customer is internal to your company or a traditional external client or user of your product, whether the product is shrink-wrapped and shipped or a service or embedded system, customer satisfaction is part of good business and of good software. Here’s to you and your customer. ☺
Books


This practical handbook shows you how to build an effective business case when you need to justify—and persuade management to accept—software change or improvement. Based on real-world scenarios, the book covers the most common situations that require business case analyses and explains specific techniques that have proved successful. The book provides examples of successful business cases; along the way, tables, tools, facts, figures, and metrics guide you through the entire analytic process. An excellent book to learn how to prepare and implement a business case and thus make software a successful business.


This book decodes the software product management process with an emphasis on coordinating the needs of stakeholders ranging from engineering, sales, and product support to technical writing and marketing. Based on real-world experience in managing the development of enterprise software, this book details how a team can work together smoothly to achieve their goal of releasing a superior software product on time. Although it’s not primarily about setting up a business, the book explains hands-on what is necessary in daily operational fights to succeed.


The book describes results from a McKinsey study about what’s driving the prosperity of the world’s best software companies and what’s responsible for the failure of others. It’s loaded with sharp insights and colorful anecdotes from leaders of companies such as Microsoft Germany, Keane Inc., BroadVision, Andersen Consulting, Oracle, Sun Microsystems, and Navision in Denmark. The authors conclude that business opportunities in the software arena remain strong. It thus serves as a huge collection of excellent lessons learned for those who are about to launch.


An up-close, personal look at the high-tech industry’s most powerful venture capitalists, technologists, and entrepreneurs. This fascinating book goes beyond Silicon Valley’s glitter and glamour to tap into the energy and vision that turned it into the epicenter of global business. The book nicely links the various dimensions—from finance to technology to operations—that make software a successful business.


This book is unfortunately titled. It is primarily about bringing real data and rigor to bear on many of the conventional “stories” about the economics of the new economy, rather than about the Microsoft antitrust situation. With some well-documented and original research, the authors conclude that Microsoft is as successful as it is for a simple reason: good products win. This book is also the best we have seen in its treatment of the overall economics of information technology standards.


Information Rules is a blueprint for success and survival in today’s highly dynamic and competitive Internet economy. The authors posit that although technology changes, the laws of economics do not. They stress that we can learn much from success stories as well as past failures. This book offers models, concepts, and analysis that give readers a deeper understanding of the fundamental principles in today’s high-tech industries and enable them to craft winning strategies for tomorrow’s network economy.

The Internet

- www.dacs.dtic.mil: State-of-the-art software-related information and technical support
- www.processimprovement.com/resources/spm.htm: Project management resources; a good entry portal for further links
- www.pmi.org: Project Management Institute; includes excerpts from the Project Management Body of Knowledge
- www.construx.com: Software engineering tools, consulting, training, and more
- www.spr.com: Software Productivity Research; estimation, project management, metrics, training, and more
- www.nnm.com: earned value Web site
- http://smallbusiness.yahoo.com: Yahoo’s popular portal for small businesses
- www.standishgroup.com: Entry point to the Standish Group’s reports and project summaries
- www.cio-dpi.gc.ca: Homepage of the Chief Information Officer Branch of the Treasury Board of Canada; good introductions into portfolio management and business cases for IT project

Suggested Reading

About the Authors

Ann Miller is the Cynthia Tang Missouri Distinguished Professor of Computer Engineering at the University of Missouri-Rolla and chair of the NATO Information Systems Technology Panel. She has held several senior management and technical positions inindustry and in government service. She is the IEEE Software associate editor in chief for management and is on the Administrative Committee of the IEEE Reliability Society. Contact her at the Univ. of Missouri-Rolla, 125 Emerson Electric Company Hall, Rolla, MO 65409-0440; millera@ece.umr.edu.

Christof Ebert is director of the Chief Technology Office in charge of software coordination and process improvement at Alcatel in Paris, where he drives R&D process change and innovation programs. Previously, he led the biggest Alcatel business unit to CMM Level 3, achieving substantial quality improvements and cycle time reduction. He is the IEEE Software associate editor in chief for requirements. Contact him at Alcatel, 54, rue la Boetie, F-75008 Paris; christof.ebert@alcatel.com.