

Slack Tom Demarco

Why Does Software Cost So Much?

Subscribed

Summary: Slack

Agile IT Organization Design

Brutal Refactoring

The Incremental Commitment Spiral Model

Fierce Leadership

Controlling Software Projects

Complete Systems Analysis

The Meaning Revolution

Software Creativity 2.0

Sticking Points

Teaching The Elephant To Dance

Elastic Leadership

An Elegant Puzzle

The Art of Agile Development

Process for System Architecture and Requirements Engineering

The Soul of Enterprise

Building the IT Consulting Practice

Creating a Software Engineering Culture

Extreme Programming Applied

Slack

The Change Monster

The Agile Manager

Trench Warfare, 1914-1918

The Great Mental Models, Volume 1

Help the Helper

Managing the Unmanageable

Microserfs

Peopleware

Slack For Dummies

Dark Harbor House

What Did You Say?

The Naked Future

Happy to Work Here

Planning Extreme Programming

The Deadline

Waltzing with Bears

Debugging Teams

Iconoclast

Slack Tom Demarco

Downloaded from hl.uconnect.hi.u.edu by guest

SCARLET GLOVER

[Why Does Software Cost So Much?](#) Tyndale House Publishers, Inc.

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.

Subscribed "O'Reilly Media, Inc."

Advisor of Leadership at Google and former vice president of leadership at LinkedIn claims that the biggest driver of motivation is the chance to serve a larger purpose beyond our careers and ourselves, rather than salary, benefits, bonuses, or other material incentives; companies that are able to successfully focus their people, their teams, and their culture around meaning outperform their competition. Fred Kofman's approach to leadership has little to do with the standard practices taught in business school and traditional books. Bringing together economics and business theory, communications and conflict resolution, family counseling and mindfulness mediation, Kofman argues in *The Meaning Revolution* that our most deep-seated, unspoken, and universal anxiety stems from our fear that our life is being wasted--that the end of life will overtake us when our song is still unsung. Material incentives--salary and benefits--account for perhaps 15 percent of employees' motivation at work. The other 85 percent is driven by a need to belong, a feeling that what we do day in and day out makes a difference, that how we spend our time on earth serves a larger purpose beyond just ourselves. Kofman claims that transcendental leaders, wherever they

are in the hierarchy, are able to put aside their self-interests and help others to feel connected with others on a team or in an organization on a great mission and part of an ennobling purpose. He argues that every organization involved in work that is nonviolent and non-addictive has what he calls an "immortality project" at its core. And the challenge for leaders is to identify and expand on that core, to inspire all stakeholders to take part.

Summary: Slack Currency

From the author of the acclaimed book *Fierce Conversations* comes the antidote to some of the most wrongheaded practices of business today. · "Provide anonymous feedback." · "Hire smart people." · "Hold people accountable." These are all sound, business practices, right? Not so fast, says leadership visionary and bestselling author Susan Scott. In fact, these mantras — despite being long-accepted and adopted by business leaders everywhere — are completely wrongheaded. Worse, they are costing companies billions of dollars, driving away valuable employees and profitable customers, limiting performance, and stalling careers. Yet they are so deeply ingrained in organizational cultures that no one has questioned them. Until now. In *Fierce Leadership*, Scott

teaches us how to spot the worst “best” practices in our organizations using a technique she calls “squid eye”—the ability to see the “tells” or signs that we have fallen prey to disastrous behaviors by knowing what to look for. Only then, she says, can we apply the antidote. Informed by over a decade of conversations with Fortune 500 executives, this book is that antidote. With fierce new approaches to everything from employee feedback to corporate diversity to customer relations, Scott offers fresh and surprising alternatives to six of the so-called “best” practices permeating today’s businesses. This refreshingly candid book is a must-read for any manager or leader at any level who is ready to take a long hard look at what trouble might be lurking in their organization - and do something about it.

[Agile IT Organization Design](#) Dorset House Publishing Company, Incorporated

A human-centric guide to solving complex problems in engineering management, from sizing teams to handling technical debt. There’s a saying that people don’t leave companies, they leave managers. Management is a key part of any organization, yet the discipline is often self-taught and unstructured. Getting to the good solutions for complex management challenges can make the difference between fulfillment and frustration for teams—and, ultimately, between the success and failure of companies. Will Larson’s An Elegant Puzzle focuses on the particular challenges of engineering management—from sizing teams to handling technical debt to performing succession planning—and provides a path to the good solutions. Drawing from his experience at Digg, Uber, and Stripe, Larson has developed a thoughtful approach to engineering management for leaders of all levels at companies of all sizes. An Elegant Puzzle balances structured principles and human-centric thinking to help any leader create more effective and rewarding organizations for engineers to thrive in.

[Brutal Refactoring](#) Addison-Wesley

The world’s economy has been transformed from a twentieth-century materials-based economy to the Age of the Knowledge-Based Economy - and the currency of this realm is ideas, imagination, creativity, and knowledge. According to The World Bank, 80% of the developed world’s wealth now resides in human capital. Perhaps President Ronald Reagan said it best in his address to Moscow State University on May 31, 1988: "Like a chrysalis, we're emerging from the economy of the Industrial Revolution - an economy confined and limited by the Earth's physical resources - into, as one economist titled his book, "the economy in mind," in which there are no bounds on human imagination and the freedom to create is the most precious natural resource." Written by Ronald Baker and Ed Kless, hosts of The Soul of Enterprise: Business in the Knowledge Economy, the popular radio show on Voice America’s Business Channel, The Soul of Enterprise: Dialogues on Business in the Knowledge Economy sounds the clarion call that organizations can no longer ignore this seismic shift that has occurred in the economy since 1959. The Soul of Enterprise introduces the three components of Intellectual Capital - human capital, social capital, and structural capital - and how to leverage them to create wealth in today’s economy, by revealing: The physical fallacy - why wealth no longer consists of tangible things, but of ideas, imagination and knowledge from human minds The best learning tool ever invented: After Action Reviews Why Frederick Taylor and the Scientific Management movement was a fraud and the wrong focus for knowledge workers The fact that effectiveness always and everywhere trumps efficiency The First Law of Pricing: All value is subjective The Second Law of Pricing: All prices are contextual The Morality of Markets: Doing well and doing good Why your organization - and you - need to be driven by a higher purpose than profit The Soul of Enterprise will inspire and challenge readers to unlock the enormous financial and competitive power hidden in the intellectual capital of their organizations and knowledge workers."

[The Incremental Commitment Spiral Model](#) Addison-Wesley Professional

From prolific and influential consultant and author Tom DeMarco comes a project management novel that vividly illustrates the principles—and the outright absurdities—that affect the productivity of a software development team. With his trademark wit set free in the novel format, DeMarco centers the plot around the development of six software products. Mr. Tompkins, a manager downsized from a giant telecommunications company, divides the huge staff of developers at his disposal into eighteen teams—three for each of the software products. The teams are different sizes and use different methods, and they compete against each other and against an impossible deadline. With these teams—and with the help of numerous "fictionalized" consultants who come to his aid—Tompkins tests the project management principles he has gathered over a lifetime. Each chapter closes with journal entries that form the core of the eye-opening approaches to management illustrated in this entertaining novel.

[Fierce Leadership](#) Dorset House Publishing Company, Incorporated

Without careful ongoing planning, the software development process can fall apart. Extreme Programming (XP) is a new programming discipline, or methodology, that is geared toward the way that the vast majority of software development projects are handled -- in small teams. In this new book, noted software engineers Kent Beck and Martin Fowler show the reader how to properly plan a software development project with XP in mind. The authors lay out a proven strategy that forces the reader to plan as their software project unfolds, and therefore avoid many of the nasty problems that can potentially spring up along the way.

[Controlling Software Projects](#) O'Reilly Media, Inc."

This is the digital version of the printed book (Copyright © 2000). Derek Hatley and Imtiaz

Pirbhai—authors of Strategies for Real-Time System Specification—join with influential consultant Peter Hruschka to present a much anticipated update to their widely implemented Hatley/Pirbhai methods. Process for System Architecture and Requirements Engineering introduces a new approach that is particularly useful for multidisciplinary system development: It applies equally well to all technologies and thereby provides a common language for developers in widely differing disciplines. The Hatley-Pirbhai-Hruschka approach (H/H/P) has another important feature: the coexistence of the requirements and architecture methods and of the corresponding models they produce. These two models are kept separate, but the approach fully records their ongoing and changing interrelationships. This feature is missing from virtually all other system and software development methods and from CASE tools that only automate the requirements model. System managers, system architects, system engineers, and managers and engineers in all of the diverse engineering technologies will benefit from this comprehensive, pragmatic text. In addition to its models of requirements and architecture and of the development process itself, the book uses in-depth case studies of a hospital monitoring system and of a multidisciplinary groundwater analysis system to illustrate the principles. Compatibility Between the H/H/P Methods and the UML: The Hatley/Pirbhai architecture and requirements methods—described in Strategies for Real-Time System Specification—have been widely used for almost two decades in system and software development. Now known as the Hatley/Hruschka/Pirbhai (H/H/P) methods, they have always been compatible with object-oriented software techniques, such as the UML, by defining architectural elements as classes, objects, messages, inheritance relationships, and so on. In Process for System Architecture and Requirements Engineering, that compatibility is made more specific through the addition of message diagrams, inheritance diagrams, and new notations that go with them. In addition, state charts, while never excluded, are now specifically included as a representation of sequential machines. These additions make definition of the system/software boundary even more straightforward, while retaining the clear separation of requirements and design at the system levels that is a hallmark of the H/H/P methods—not shared by most OO techniques. Once the transition to software is made, the developer is free to continue using the H/H/P methods, or to use the UML or any other software-specific technique.

[Complete Systems Analysis](#) Addison-Wesley

The shock and slaughter of the battlefields of the Somme, Verdun and Passchendale is well documented. However, during the smaller battles soldiers could, and often did, make personal decisions. From these evolved a culture of live and let live, which constrained that of kill and be killed.

[The Meaning Revolution](#) Harper Collins

This is the first time in American history that we have had four different generations working side-by-side in the workplace: the Traditionalists (born before 1945), the Baby Boomers (born 1945-1964), Gen X (born 1965-1980), and the Millennials (born 1981-2001). Haydn Shaw, popular business speaker and generational expert, has identified 12 places where the 4 generations typically come apart in the workplace (and in life as well). These sticking points revolve around differing attitudes toward managing one’s own time, texting, social media, organizational structure, and of course, clothing preferences. If we don’t learn to work together and stick together around these 12 sticking points, then we’ll be wasting a lot of time fighting each other instead of enjoying a friendly and productive team. Sticking Points is a must-read book that will help you understand the generational differences you encounter while teaching how we can learn to speak one another’s language and get better results together.

[Software Creativity 2.0](#) Addison-Wesley Professional

The must-read summary of Tom DeMarco's book: "Slack: Getting Past Burnout, Busywork and the Myth of Total Efficiency". This complete summary of the ideas from Tom DeMarco's book "Slack"

shows that companies who constantly insist on efficiency are actually creating an environment where they limit flexibility, reduce the capacity for creativity and increase burnout in employees. In his book, the author explains that effective managers should be able to identify and introduce exactly the right amount of slack within working practices in order to avoid these outcomes. This summary introduces four key components for implementing this philosophy and the two different types of slack that will create the best working environment with high levels of efficiency. Added-value of this summary: • Save time • Understand key concepts • Expand your knowledge To learn more, read "Slack" and discover the key to creating a flexible working environment that will get the best out of your employees.

[Sticking Points](#) Addison-Wesley

“The title makes a huge promise: a way to divide commitment into increments that are both meetable (good news for developers) and meaningful (good news for managers and stakeholders). And the book makes good on that promise.” -Tom DeMarco, Principal, The Atlantic Systems Guild, author of Peopleware, Deadline, and Slack “I am seriously impressed with this ICSM book. Besides being conceptually sound, I was amazed by the sheer number of clear and concise characterizations of issues, relationships, and solutions. I wanted to take a yellow highlighter to it until I realized I’d be highlighting most of the book.” -Curt Hibbs, Chief Agile Evangelist, Boeing Use the ICSM to Generate and Evolve Your Life-Cycle Process Assets to Best Fit Your Organization’s Diverse and Changing Needs Many systems development practitioners find traditional “one-size-fits-all” processes inadequate for the growing complexity, diversity, dynamism, and assurance needs of their products and services. The Incremental Commitment Spiral Model (ICSM) responds with a principle- and risk-based framework for defining and evolving your project and corporate process assets, avoiding pitfalls and disruption, and leveraging opportunities to increase value. This book explains ICSM’s framework of decision criteria and principles, and shows how to apply them through relevant examples. It demonstrates ICSM’s potential for reducing rework and technical debt, improving maintainability, handling emergent requirements, and raising assurance levels. Its coverage includes What makes a system development successful ICSM’s goals, principles, and usage as a process-generation framework Creating and evolving processes to match your risks and opportunities Integrating your current practices and adopting ICSM concepts incrementally, focusing on your greatest needs and opportunities About the Website: Download the evolving ICSM guidelines, subprocesses, templates, tools, white papers, and academic support resources at csse.usc.edu/ICSM.

[Teaching The Elephant To Dance](#) Penguin UK

The world is in a permanent state of change. We must work in new ways. To change the work we must change how we manage; how we think about management. What got you here won't get you there. There are new ways of managing which are changing business, government, and not-for-profit organisations, big and small. This isn't about leaders, it's about managers. It's not about mystical mind methods, it's about principles to work by. It's about agility, the agile Manager. This book starts your journey. The book has four sections: 1. A set of principles which you as an agile manager must get your head around in order to function in the new world. 2. A set of management practices which follow from those principles. 3. A set of agile work practices that you need to understand and support. 4. Guidance on the journey to new ways of working. . . . Agile is a way of thinking about work. Agile is a thing now, it has become a noun as well as an adjective. Agile thinking is impacting Information Technology, enterprises, government, and society. It may have started in IT but now it is transforming work everywhere, and even how we think. Its impact is far-reaching enough to talk of it as a renaissance in thinking, a refresh or step change that comes only once or twice a century. This is not an exaggeration. . . . It is not just Agile, there is a suite of new ideas transforming work. They all aim for "better value sooner, safer, happier". We simply call them the New Ways of Working, NWoW. This book is about the impact of these new ways on management in the modern enterprise. . . . The new thinking empowers people to be knowledge workers, to design the work and make the decisions. It treats them like they are over 18 and on the same side. Conventional management too often treats people like clerical workers, like plug-compatible wetware, like Human Resources, who can't be trusted, who are evaluated numerically, who are an overhead to be minimised, who need to be told what to do and how to do it. Which is not conducive to satisfaction and mental health. . . . The Agile way is iterative, incremental, experimenting, exploring complex systems. These are displacing the ideas of conventional enterprises: big-bang projects; zero risk; certainty and accuracy; plan once execute perfectly; failure is not an option. Along the way Agile is resurfacing (and standing on the shoulders of) the

ideas of Lean, which ironically go back pre-second-world-war; and Agile is drawing on the principles of complex systems and the modern understanding of human behaviour and social constructs. . . . At least as important, though, is New Ways of Managing. Too often, management views the transformation to New Ways as something done to improve the practitioner workforce, not management. This can't be. For an organisation to change, the management must change. This is one of the biggest issues facing organisations moving to agile ways of working. Managers must understand and focus on empowerment, collaboration, agility, and flow. Why focus on management in agile transformation? Because we see it often neglected, and because it is the key. This book is about how to manage in an agile way, not what agile work looks like. Plenty has been written about that. . . . It is our personal offering. We hope you like it and derive value from it. Join our community and tell us how you felt about it.

[Elastic Leadership](#) Simon and Schuster

"A thorough yet thoroughly digestible book on the ubiquity of data gathering and the unraveling of personal privacy." —Daniel Pink, author of Drive Thanks to recent advances in technology, prediction models for individual behavior grow more sophisticated by the day. Whether you'll marry, commit a crime or fall victim to one, or contract a disease are becoming easily accessible facts. The naked future is upon us, and the implications are staggering. Patrick Tucker draws on fascinating stories from health care to urban planning to online dating. He shows how scientists can predict your behavior based on your friends' Twitter updates, anticipate the weather a year from now, figure out the time of day you're most likely to slip back into a bad habit, and guess how well you'll do on a test before you take it. Tucker knows that the rise of Big Data is not always a good thing. But he also shows how we've gained tremendous benefits that we have yet to fully realize.

[An Elegant Puzzle](#) developer.* Books

From the era-defining author of Generation X comes a novel of overworked coders who escape the serfdom of Bill Gates to forge their own path. They are Microserfs—six code-crunching computer whizzes who spend upward of sixteen hours a day "coding" and eating "flat" foods (food which, like Kraft singles, can be passed underneath closed doors) as they fearfully scan company e-mail to learn whether the great Bill is going to "flame" one of them. But now there's a chance to become innovators instead of cogs in the gargantuan Microsoft machine. The intrepid Microserfs are striking out on their own—living together in a shared digital flophouse as they desperately try to cultivate well-rounded lives and find love amid the dislocated, subhuman whirl and buzz of their computer-driven world.

[The Art of Agile Development](#) Penguin

"The real lessons of teamwork don't happen on camera. They happen behind the closed doors of locker rooms and team meetings and practice facilities. Kevin and John open those closed doors. All you need to do is get reading!" —Larry Bird "Help the helper" is a basketball motto preached by

some of the sport's legendary coaches, including Dean Smith and Phil Jackson. All good players know they should support a teammate who's under pressure. But the true greats know how to take it one step further. They fill the gaps left behind when one teammate goes to help another—gaps that are often far from the basket and out of the spotlight. The true greats step up in quiet ways to make sure no subtle holes develop on defense and no opportunities are missed on offense. Help the Helper will show you how to put this level of teamwork to work in your business, to build a culture that recognizes and rewards those who help the helper—even when they don't have sexy statistics. In the process, it will teach you how to de-emphasize the CEO/quarterback/superstar and effectively redefine leadership. You'll learn, for instance, how to: Create a dynasty of unselfishness. Manage energy, not people. Eat obstacles for breakfast. Act like an "unleader." Consider how it works in the hospitality industry. In a great restaurant you don't have to wait for your server to check on you; your needs are taken care of instantaneously, sometimes before you notice them. Everyone from the busboy to the maitre d' has one goal: the success of the team. Such coordination seems complicated for a small eatery, nearly impossible for a large organization. But it's easier than you think. For a combined forty years, Pritchard and Eliot have focused on building high-performing groups. They've crushed Malcolm Gladwell's 10,000-Hour Rule, logging upward of 50,000 hours studying the factors that create champions and dynasties, from the NBA and Major League Baseball to the Fortune 500. Exhaustive testing, scouting, and evaluating have taught them that truly special teams in all fields have one common denominator: a willingness to do whatever it takes to help the helper. Drawing on true and inspirational stories from sports to medicine to business, Help the Helper shows what's behind the curtain that fuels great team performance.

[Process for System Architecture and Requirements Engineering](#) Penguin

This is the digital version of the printed book (Copyright © 2003). If There's No Risk On Your Next Project, Don't Do It. Greater risk brings greater reward, especially in software development. A company that runs away from risk will soon find itself lagging behind its more adventurous competition. By ignoring the threat of negative outcomes—in the name of positive thinking or a can-do attitude—software managers drive their organizations into the ground. In *Waltzing with Bears*, Tom DeMarco and Timothy Lister—the best-selling authors of *Peopleware*—show readers how to identify and embrace worthwhile risks. Developers are then set free to push the limits. The authors present the benefits of risk management, including that it makes aggressive risk-taking possible, protects management from getting blindsided, provides minimum-cost downside protection, reveals invisible transfers of responsibility, isolates the failure of a subproject. Readers are armed with strategies for confronting the most common risks that software projects face: schedule flaws, requirements inflation, turnover, specification breakdown, and under-performance. *Waltzing with Bears* will help you mitigate the risks—before they turn into project-killing problems. Risks are out there—and they should be there—but there is a way to manage them.

[The Soul of Enterprise](#) Pfeiffer

Articulating the principles behind Extreme Programming (XP) and offering practical advice

concerning its application, this guide outlines the first steps toward XP discipline and offers examples of its application to a variety of organizations. It provides guidelines for implementing XP, highlighting key points with anecdotes drawn from the experiences of those who developed the methodology. Auer and Miller are software developers. c. Book News Inc.

[Building the IT Consulting Practice](#) Pan Macmillan

This is the digital version of the printed book (Copyright © 1996). Written in a remarkably clear style, *Creating a Software Engineering Culture* presents a comprehensive approach to improving the quality and effectiveness of the software development process. In twenty chapters spread over six parts, Wiegers promotes the tactical changes required to support process improvement and high-quality software development. Throughout the text, Wiegers identifies scores of culture builders and culture killers, and he offers a wealth of references to resources for the software engineer, including seminars, conferences, publications, videos, and on-line information. With case studies on process improvement and software metrics programs and an entire part on action planning (called "What to Do on Monday"), this practical book guides the reader in applying the concepts to real life. Topics include software culture concepts, team behaviors, the five dimensions of a software project, recognizing achievements, optimizing customer involvement, the project champion model, tools for sharing the vision, requirements traceability matrices, the capability maturity model, action planning, testing, inspections, metrics-based project estimation, the cost of quality, and much more! Principles from Part 1 Never let your boss or your customer talk you into doing a bad job. People need to feel the work they do is appreciated. Ongoing education is every team member's responsibility. Customer involvement is the most critical factor in software quality. Your greatest challenge is sharing the vision of the final product with the customer. Continual improvement of your software development process is both possible and essential. Written software development procedures can help build a shared culture of best practices. Quality is the top priority; long-term productivity is a natural consequence of high quality. Strive to have a peer, rather than a customer, find a defect. A key to software quality is to iterate many times on all development steps except coding: Do this once. Managing bug reports and change requests is essential to controlling quality and maintenance. If you measure what you do, you can learn to do it better. You can't change everything at once. Identify those changes that will yield the greatest benefits, and begin to implement them next Monday. Do what makes sense; don't resort to dogma.

[Creating a Software Engineering Culture](#) Currency

Through vivid accounts of successful innovators ranging from glass artist Dale Chihuly to physicist Richard Feynman to the country/rock trio the Dixie Chicks, Berns reveals the inner workings of the iconoclast's mind with remarkable clarity. Each engaging chapter goes on to describe practical actions we can each take to understand and unleash our own potential to think differently—such as seeking out new environments, novel experiences, and first-time acquaintances.