

The Incremental Commitment Spiral Model Principles And Practices For Successful Systems And Software

Right here, we have countless book The Incremental Commitment Spiral Model Principles And Practices For Successful Systems And Software and collections to check out. We additionally find the money for variant types and also type of the books to browse. The okay book, fiction, history, novel, scientific research, as with ease as various supplementary sorts of books are readily comprehensible here.

As this The Incremental Commitment Spiral Model Principles And Practices For Successful Systems And Software, it ends up mammal one of the favored books The Incremental Commitment Spiral Model Principles And Practices For Successful Systems And Software collections that we have. This is why you remain in the best website to see the incredible book to have.

Disciplinary Convergence in Systems Engineering Research Azad M. Madni 2017-11-24 The theme of this volume on systems engineering research is disciplinary convergence: bringing together concepts, thinking, approaches, and technologies from diverse disciplines to solve complex problems. Papers presented at the Conference on Systems Engineering Research (CSER), March 23-25, 2017 at Redondo Beach, CA, are included in this volume. This collection provides researchers in academia, industry, and government forward-looking research from across the globe, written by renowned academic, industry and government researchers.

Implementation Patterns - Studentenausgabe Kent Beck 2010

Refactoring Martin Fowler 2020-03-20 • Umfassend überarbeitete und aktualisierte Neuauflage des Standardwerks in vollständig neuer Übersetzung • Verbesserungsmöglichkeiten von bestehender Software anhand von Code-Smells erkennen und Code effizient überarbeiten • Umfassender Katalog von Refactoring-Methoden mit Code-Beispielen in JavaScript Seit mehr als zwanzig Jahren greifen erfahrene Programmierer rund um den Globus auf dieses Buch zurück, um bestehenden Code zu verbessern und leichter lesbar zu machen sowie Software besser warten und erweitern zu können. In diesem umfassenden Standardwerk zeigt Ihnen Martin Fowler, was die Vorteile von Refactoring sind, wie Sie verbesserungsbereitigen Code erkennen und wie Sie ein Refactoring – unabhängig von der verwendeten Programmiersprache – erfolgreich durchführen. In einem umfangreichen Katalog gibt Fowler Ihnen verschiedene Refactoring-Methoden mit ausführlicher Erläuterung, Motivation, Vorgehensweise und einfachen Beispielen in JavaScript an die Hand. Darüber hinaus behandelt er insbesondere folgende Schwerpunkte: • Allgemeine Prinzipien und Durchführung des Refactorings • Refactoring anwenden, um die Lesbarkeit, Wartbarkeit und Erweiterbarkeit von Programmen zu verbessern • Code-Smells erkennen, die auf Verbesserungsmöglichkeiten durch Refactoring hinweisen • Entwicklung zuverlässiger Tests für das Refactoring • Erkennen von Fallstricken und notwendigen Kompromissen bei der Durchführung eines Refactorings Diese vollständig neu übersetzte Ausgabe wurde von Grund auf überarbeitet, um den maßgeblichen Veränderungen der modernen Programmierung Rechnung zu tragen. Sie enthält einen aktualisierten Katalog von Refactoring-Methoden sowie neue Beispiele für einen funktionalen Programmieransatz.

INCOSE Systems Engineering Handbook INCOSE 2015-06-12 A detailed and thorough reference on the discipline and practice of systems engineering The objective of the International Council on Systems Engineering (INCOSE) Systems Engineering Handbook is to describe key process activities performed by systems engineers and other engineering professionals throughout the life cycle of a system. The book covers a wide range of fundamental system concepts that broaden the thinking of the systems engineering practitioner, such as system thinking, system science, life cycle management, specialty engineering, system of systems, and agile and iterative methods. This book also defines the discipline and practice of systems engineering for students and practicing professionals alike, providing an authoritative reference that is acknowledged worldwide. The latest edition of the INCOSE Systems Engineering Handbook: Is consistent with ISO/IEC/IEEE 15288:2015 Systems and software engineering—System life cycle processes and the Guide to the Systems Engineering Body of Knowledge (SEBoK) Has been updated to include the latest concepts of the INCOSE working groups Is the body of knowledge for the INCOSE Certification Process This book is ideal for any engineering professional who has an interest in or needs to apply systems engineering practices. This includes the experienced systems engineer who needs a convenient reference, a product engineer or engineer in another discipline who needs to perform systems engineering, a new systems engineer, or anyone interested in learning more about systems engineering.

The People's Scrum Tobias Mayer 2017-11-14 Dieses außergewöhnliche Buch vermittelt Scrum in einzigartiger Weise als eine Weltanschauung und nicht ausschließlich als Werkzeug oder Methode. Dabei wird Scrum aus der Perspektive des erfahrenen Anwenders betrachtet und pragmatische Agilität in den Mittelpunkt gestellt. Der Leser findet in 39 Essays viele Denkanstöße, Ideen und praktische Hinweise zu Agilität und Scrum mit den drei zentralen Themen: "Menschen", "Prozesse" und "Kultur". Hierbei geht es gleichermaßen um das Individuum, um Selbstorganisation, technische Schulden und Aufwandschätzung wie um Firmenpolitik und die Agilität großer Organisationen. Die Essays spiegeln ein breites Spektrum von Denkansätzen und Vorgehensweisen wider: Sie wurzeln im Pragmatismus oder Idealismus, gründen auf Erfahrung oder entspringen abstraktem Denken. Teils beziehen sie sich aufeinander, teils widersprechen oder ergänzen sie sich. Dabei nehmen die Autoren in Form von persönlichen Erfahrungsberichten und Anekdoten immer wieder eine kritische Sichtweise ein. Dadurch wird auch der Leser ermutigt, seine Handlungsweisen und Entwicklungspraktiken im Projektmanagement, in Arbeitsprozessen sowie in der Firmenpolitik kritisch zu hinterfragen und neue Wege zu gehen. Die 2. Auflage wurde vollständig überarbeitet und in zahlreichen Aspekten ergänzt.

Self-Aware Computing Systems Samuel Kounev 2017-01-23 This book provides formal and informal definitions and taxonomies for self-aware computing systems, and explains how self-aware computing relates to many existing subfields of computer science, especially software engineering. It describes architectures and algorithms for self-aware systems as well as the benefits and pitfalls of self-awareness, and reviews much of the latest relevant research across a wide array of disciplines, including open research challenges. The chapters of this book are organized into five parts: Introduction, System Architectures, Methods and Algorithms, Applications and Case Studies, and Outlook. Part I offers an introduction that defines self-aware computing systems from multiple perspectives, and establishes a formal definition, a taxonomy and a set of reference scenarios that help to unify the remaining chapters. Next, Part II explores architectures for self-aware computing systems, such as generic concepts and notations that allow a wide range of self-aware system architectures to be described and compared with both isolated and interacting systems. It also reviews the current state of reference architectures, architectural frameworks, and languages for self-aware systems. Part III focuses on methods and algorithms for self-aware computing systems by addressing issues pertaining to system design, like modeling, synthesis and verification. It also examines topics such as adaptation, benchmarks and metrics. Part IV then presents applications and case studies in various domains including cloud computing, data centers, cyber-physical systems, and the degree to which self-aware computing approaches have been adopted within those domains. Lastly, Part V surveys open challenges and future research directions for self-aware computing systems. It can be used as a handbook for professionals and researchers working in areas related to self-aware computing, and can also serve as an advanced textbook for lecturers and postgraduate students studying subjects like advanced software engineering, autonomic computing, self-adaptive systems, and data-center resource management. Each chapter is largely self-contained, and offers plenty of references for anyone wishing to pursue the topic more deeply.

Agile and Lean Service-Oriented Development: Foundations, Theory, and Practice Wang, Xiaofeng 2012-11-30 Challenges in unpredictable markets, changing customer requirements, and advancing information technologies have led to progression towards service oriented engineering and agile and lean software development. These prevailing approaches to software systems provide solutions to challenges in demanding business environments. Agile and Lean Service-Oriented Development: Foundations, Theory and Practice explores the groundwork of service-oriented and agile and lean development and the conceptual basis and experimental evidences for the combination of the two approaches. Highlighting the best tools and guidelines for these developments in practice, this book is essential for researchers and practitioners in the software development and service computing fields.

Healthcare IT Transformation John C. Dodd 2016-10-04 This book gives examples from healthcare institutions that are using IT automation and innovation to drive change and provides guidance on the strategic direction of HIT over the next five years. Improving the delivery of healthcare through HIT is vital for both the economic success of healthcare organizations and the care of the patient, but most EMR systems do not have an integrated and architected approach. This book provides a detailed approach on how to leverage IT for transformation. It also shows how to build upon the experiences of other industries and helps foster innovation by providing a vision of where technology can be an enabler.

Das UML-Benutzerhandbuch Grady Booch 2006 In dieser - lang erwarteten - Überarbeitung zur Version 2.0 der umfassenden Einführung in UML bieten die Entwickler der Sprache - Grady Booch, James Rumbaugh, Ivar Jacobsen - eine Einführung, die sich mit den Kernpunkten befasst. Ausgehend von einer Übersicht über UML wird die Sprache anhand der Vorstellung bestimmter Konzepte und Schreibweisen in jedem Kapitel Schritt für Schritt erläutert. Das Buch sorgt einerseits für einen umfassenden Überblick über alle Diagrammtypen sowie Elemente von UML in der zweiten Version und stellt andererseits den nötigen Praxisbezug her, um UML 2.0 effektiv für eigene Projekte einzusetzen. Die tief greifenden Erläuterungen und die an Beispielen orientierte Herangehensweise der Autoren, sorgen für ein schnelles Verständnis des komplexen Themas.

Das Buch der Fragen Gregory Stock 2012

Agiles Coaching Rachel Davies 2010

Sieben Wochen, sieben Sprachen (Prags) Bruce A. Tate 2011-06-30 Mit diesen sieben Sprachen erkunden Sie die wichtigsten Programmiermodelle unserer Zeit. Lernen Sie die dynamische Typisierung kennen, die Ruby, Python und Perl so flexibel und verlockend macht. Lernen Sie das Prototyp-System verstehen, das das Herzstück von JavaScript bildet. Erfahren Sie, wie das Pattern Matching in Prolog die Entwicklung von Scala und Erlang beeinflusst hat. Entdecken Sie, wie sich die rein funktionale Programmierung in Haskell von der Lisp-Sprachfamilie, inklusive Clojure, unterscheidet. Erkunden Sie die parallelen Techniken, die das Rückgrat der nächsten Generation von Internet-Anwendungen bilden werden. Finden Sie heraus, wie man Erlangs "Lass es abstürzen"-Philosophie zum Aufbau fehlertoleranter Systeme nutzt. Lernen Sie das Actor-Modell kennen, das das parallele Design bei Io und Scala bestimmt. Entdecken Sie, wie Clojure die Versionierung nutzt, um einige der schwierigsten Probleme der Nebenläufigkeit zu lösen. Hier finden Sie alles in einem Buch. Nutzen Sie die Konzepte einer Sprache, um kreative Lösungen in einer anderen Programmiersprache zu finden - oder entdecken Sie einfach eine Sprache, die Sie bisher nicht kannten. Man kann nie wissen - vielleicht wird sie sogar eines ihrer neuen Lieblingswerkzeuge.

Software Project Management in a Changing World Günther Ruhe 2014-09-04 By bringing together various current directions, Software Project Management in a Changing World focuses on how people and organizations can make their processes more change-adaptive. The selected chapters closely correspond to the project management knowledge areas introduced by the Project Management Body of Knowledge, including its extension for managing software projects. The contributions are grouped into four parts, preceded by a general introduction. Part I "Fundamentals" provides in-depth insights into fundamental topics including resource allocation, cost estimation and risk management. Part II "Supporting Areas" presents recent experiences and results related to the management of quality systems, knowledge, product portfolios and global and virtual software teams. Part III "New Paradigms" details new and evolving software-development practices including agile, distributed and open and inner-source development. Finally, Part IV "Emerging Techniques" introduces search-based techniques, social media, software process simulation and the efficient use of empirical data and their effects on software-management practices. This book will attract readers from both academia and practice with its excellent balance between new findings and experience of their usage in new contexts. Whenever appropriate, the presentation is based on evidence from empirical evaluation of the proposed approaches. For researchers and graduate students, it presents some of the latest methods and techniques to accommodate new challenges facing the discipline. For professionals, it serves as a source of inspiration for refining their project-management skills in new areas.

Acquisition Perspectives on the Incremental Commitment Model 2008 CONCLUSIONS: (1) STRENGTHS -- ICM is a promising, new, development life cycle model * Recent ICM publications do contribute to the better understanding of spiral development principles * In ICM the original spiral graphical metaphor has been replaced with the uncoiled spiral, making the model's use easier for project managers * ICM emphasizes the importance of gaining stakeholder commitment before progressing to the next life cycle phase. (2) WEAKNESSES -- The renaming of APs to DRs deemphasized the earlier, important notion in the Spiral Model that all activities in a spiral increment are focusing on the satisfaction of the objectives of the upcoming Anchor Point * The mapping of ICM Anchor Points into the DoD 5000.2 milestones is artificial and not supportive of either the DoD 5000.2 instruction in general, to its preferred, evolutionary acquisition strategy in particular.

A Holistic View of Software and Hardware Reuse Fevzi Belli 2021-06-11 This book focuses on software reuse and the chances, dependability tests and recommendations for best reuse practice. A short introduction of the Ecodesign of hardware is given combined with the latest update of relevant EU legislation and standardization. It also describes the combination of different states of software in a E&E system in order to guarantee dependability of the product to be resold.

Versionskontrolle mit Git Jon Loeliger 2010 Git wurde von keinem Geringeren als Linus Torvalds ins Leben gerufen. Sein Ziel: die Zusammenarbeit der in aller Welt verteilten Entwickler des Linux-Kerns zu optimieren. Mittlerweile hat das enorm schnelle und flexible System eine große Fangemeinde gewonnen. Viele Entwickler ziehen es zentralisierten Systemen vor, und zahlreiche bekannte Entwicklungsprojekte sind schon auf Git umgestiegen. Verständliche Einführung: Wer Git einsetzen und dabei großmögliche Nutzen aus seinen vielseitigen Funktionen ziehen möchte, findet in diesem Buch einen idealen Begleiter. Versionskontrolle mit Git führt grundlich und gut verständlich in die leistungsstarke Open Source-Software ein und demonstriert ihre vielfältigen Einsatzmöglichkeiten. Auf dieser Basis kann der Leser Git schon nach kurzer Zeit produktiv nutzen und optimal auf die Besonderheiten seines Projekts abstimmen. Insider-Tipps aus erster Hand: Jon Loeliger, der selbst zum Git-Entwicklerteam gehört, lässt den Leser tief ins Innere des Systems blicken, so dass er ein umfassendes Verständnis seiner internen Datenstrukturen und Aktionen erlangt. Neben alltäglichen Szenarios behandelt Loeliger auch fortgeschrittene Themen wie die Verwendung von Hooks zum Automatisieren von Schritten, das Kombinieren von mehreren Projekten und Repositories zu einem Superprojekt sowie die Arbeit mit Subversion-Repositories in Git-Projekten.

Der rational unified process Philippe Kruchten 1999

Informatik-Handbuch Peter Rechenberg 2006 Eine Gesamtdarstellung der Informatik auf Universitätsniveau. Inzwischen ein etabliertes Standardwerk, dessen Inhalt für diese Auflage (3. Auflage BA 7/02) erweitert, aktualisiert und neu akzentuiert wurde. Unter den neu aufgenommenen Kapiteln finden sich die Themen Modellierung und Theorie verteilter Systeme, Dokument-Austauschformate und Auszeichnungssprachen, virtuelle und erweiterte Realität, Web-Programmierung und Rechtsfragen der Informatik. Überarbeitet wurden die Kapitel Softwaretechnik, grafische Datenverarbeitung, das Internet und Normen und Spezifikationen. Auch die übrigen Kapitel befinden sich jetzt auf dem neuesten Stand. Demgegenüber verzichten die Autoren auf die Behandlung der visuellen Programmierung und ebenfalls

entfernt wurde der komplette Abschnitt zur Wirtschaftsinformatik. Diese eigenständige Disziplin hat sich in den letzten Jahren deutlich weiterentwickelt und eine Berücksichtigung hätte den Rahmen des Buches gesprengt. Aufbau und Gliederung sind unverändert, die Literaturangaben enthalten, je nach Thema, teilweise neue Titel. Voraufgabe ersetzen. (2 A).

Handbook of Software Engineering Sungdeok Cha 2019-02-11 This handbook provides a unique and in-depth survey of the current state-of-the-art in software engineering, covering its major topics, the conceptual genealogy of each subfield, and discussing future research directions. Subjects include foundational areas of software engineering (e.g. software processes, requirements engineering, software architecture, software testing, formal methods, software maintenance) as well as emerging areas (e.g., self-adaptive systems, software engineering in the cloud, coordination technology). Each chapter includes an introduction to central concepts and principles, a guided tour of seminal papers and key contributions, and promising future research directions. The authors of the individual chapters are all acknowledged experts in their field and include many who have pioneered the techniques and technologies discussed. Readers will find an authoritative and concise review of each subject, and will also learn how software engineering technologies have evolved and are likely to develop in the years to come. This book will be especially useful for researchers who are new to software engineering, and for practitioners seeking to enhance their skills and knowledge.

Boosting Collaborative Networks 4.0 Luis M. Camarinha-Matos 2020-11-16 This book constitutes the refereed proceedings of the 21st IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2020, held in Valencia, Spain, in November 2020. The conference was held virtually. The 53 full papers were carefully reviewed and selected from 135 submissions. They provide a comprehensive overview of major challenges and recent advances in various domains related to the digital transformation and collaborative networks and their applications with a strong focus on the following areas related to the main theme of the conference: collaborative business ecosystems; collaborative business models; collaboration platform; data and knowledge services; blockchain and knowledge graphs; maintenance, compliance and liability; digital transformation; skills for organizations of the future; collaboration in open innovation; collaboration in supply chain; simulation and analysis in collaborative systems; product and service systems; collaboration impacts; boosting sustainability through collaboration in Agri-food 4.0; digital innovation hubs for digitalizing European industry; and collaborative networks for health and wellness data management.

Software Design and Development: Concepts, Methodologies, Tools, and Applications Management Association, Information Resources 2013-07-31 Innovative tools and techniques for the development and design of software systems are essential to the problem solving and planning of software solutions. Software Design and Development: Concepts, Methodologies, Tools, and Applications brings together the best practices of theory and implementation in the development of software systems. This reference source is essential for researchers, engineers, practitioners, and scholars seeking the latest knowledge on the techniques, applications, and methodologies for the design and development of software systems.

Extreme Programming Kent Beck 2003

The Incremental Commitment Spiral Model 2014

Smart Service Engineering Oliver Thomas 2016-11-21 Dieser Sammelband stellt innovative Methoden, Konzepte und Werkzeuge zur Transformation klassischer Dienstleistungen zu sogenannten „Smart Services“ vor. Durch Digitalisierung, Individualisierung und Kundenzentrierung entstehen flexible und zukunftsweisende Modelle, die branchenübergreifend als Schlüssel zu erfolgreichen Dienstleistungsinnovationen zu nutzen sind. Zahlreiche Beispiele verdeutlichen die Leistungsfähigkeit und die praktische Umsetzbarkeit der Smart Services als wesentliche Komponente neuer Geschäftsmodelle.

Software testen und analysieren Mauro Pezzè 2009-01-01 Software Testen und Analysieren: Prozesse, Prinzipien und Techniken ist das erste Buch, das eine grosse Bandbreite sich ergänzender Software-Test und -Analysetechniken in einer ganzheitlichen, schlussigen Form erklärt. Es behandelt das ganze Themenspektrum, angefangen bei den Grundlagen und Basistheorien bis hin zu Organisations- und Prozessfragen von Anwendersoftware. Der Schwerpunkt des Buches liegt darin, anwendungsorientierte Techniken auszuwählen, um beim Testen und Analysieren von Software eine akzeptable Qualität zu akzeptablen Preisen zu bekommen.

Die Organisation des Wissens Ikujiro Nonaka 2012-03 In jedem Mitarbeiter schlummert wertvolles Wissen. Wer seine Firma richtig organisiert, kann diese Schätze heben. Japanische Manager haben dies früh begriffen und ihren Unternehmen langfristige Wettbewerbsvorteile verschafft. Anhand ausführlicher Fallbeispiele erläutern die Autoren in diesem Klassiker der Managementliteratur, wie Führungskräfte die Ressource Wissen nutzbar machen und sie in erfolgreiche Produkte und Technologien umwandeln können. "Dies ist das bisher kreativste Buch über Management aus Japan. Absolute Pflichtlektüre für Manager in einer Welt ohne Grenzen." Kenichi Ohmae "Ein wichtiges Buch für die Unternehmenspraxis." Michael E. Porter "Mit Nonakas Thesen zeichnet sich ein Paradigmenwechsel ab." Wirtschaftswoche "Spannend geschrieben, mit vielen Aha-Erlebnissen beim Lesen." ManagerSeminar

Wirtschaftsinformatik als Schlüssel zum Unternehmenserfolg Kerstin Fink 2007-10-30 Renommierte Autoren aus Wissenschaft, Praxis und Hochschulpolitik diskutieren die Bedeutung der Wirtschaftsinformatik als Schlüssel zum Unternehmenserfolg.

CMMI Mary Beth Chrissis 2009

Spiral Development Barry W. Boehm 2000 Abstract: "Spiral development is a family of software development processes characterized by repeatedly iterating a set of elemental development processes and managing risk so it is actively being reduced. This paper characterizes spiral development by enumerating a few 'invariant' properties that any such process must exhibit. For each, a set of 'variants' is also presented, demonstrating a range of process definitions in the spiral development family. Each invariant excludes one or more 'hazardous spiral look-alike' models, which are also outlined. This report also shows how the spiral model can be used for a more cost-effective incremental commitment of funds, via an analogy of the spiral model to stud poker. An important and relatively recent innovation to the spiral model has been the introduction of anchor point milestones. The latter part of the paper describes and discusses these."

The Incremental Commitment Spiral Model Barry Boehm 2014-05-29 "The title makes a huge promise: a way to divide commitment into increments that are both measurable (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.

Scrum für Dummies Mark C. Layton 2019-01-04 Das Schöne an Scrum ist, dass das Regelwerk so überschaubar ist. Es schafft nur so viel Struktur, dass Teams sich ganz und gar auf die Lösung der eigentlichen Herausforderung konzentrieren können und keine Zeit mit der Abarbeitung unnötiger und bereits überholter Prozesse verlieren. Das Buch zeigt Ihnen, wie Sie ein Team zusammenstellen und diese beliebte agile Projektmanagementmethode implementieren, um Projekte reibungsloser zu gestalten, und zwar vom Anfang bis zum Ende. Wenn Sie möchten, auch in Ihrem Privatleben. Scrum wird Ihnen das Leben leichter machen. Warum probieren Sie es nicht einfach aus?

Die Scrum-Revolution Jeff Sutherland 2015-01-08 »Scrum« heißt die revolutionäre Methode, die seit den 90er-Jahren große IT-Projekte zum Fliegen bringt. Und das schneller und kostengünstiger als geplant: Unternehmen, die mit Scrum arbeiten, schaffen die doppelte Arbeit in der Hälfte der Zeit. Gar nicht auszudenken, was geschähe, wenn jede Firma von dieser Methode profitieren könnte! Genau das ist Jeff Sutherlands Mission. Als Scrum-Erfinder zeigt er in seinem neuen Standardwerk ganz normalen Unternehmen, wie sie Scrum-Teams etablieren, ihre Entwicklungsaufgaben vereinfachen und alle ihre Projekte agil, zügig und kostengünstig durchziehen.

Agile Transformation Bas van Lieshout 2020-12-04 Transformation ist eine Reise! Agile Transformation bedeutet eine völlige Kehrtwendung und damit eine völlige Umgestaltung der gesamten Organisation. Wie wird nun eine solche unternehmensweite Umstellung auf Agilität umgesetzt? Was sind die Voraussetzungen für den Erfolg einer solchen Transformation? Das Buch bietet Transformationsteams eine praktische Anleitung zu allen Aspekten eines Transformationsprozesses. Die Autoren, ausgewiesene Experten auf diesem Gebiet, beschreiben anhand vieler Fallbeispiele einen Schritt-für-Schritt-Plan zur Vorbereitung und Durchführung der agilen Transformation, wahlweise, dass dieser Plan kontinuierlich angepasst werden muss, basierend auf den Erkenntnissen und Ergebnissen der vorherigen Lernerfahrungen. Auch Themen, die während der Transformation zusätzliche Aufmerksamkeit erfordern und gerne vergessen werden, wie u.a. Personalentwicklung, Finanzen und Compliance, wird ein eigenes Kapitel gewidmet. Das Buch ist Pflichtlektüre für alle, die sich mit agilen Transformationen und Veränderung im Allgemeinen befassen und ihr Unternehmen in eine nachhaltige und bewegliche Zukunft führen möchten.

Die Essenz von Kanban - kompakt David J. Anderson 2017-11-30

Agiles Projektmanagement mit Scrum Ken Schwaber 2007

The Future of Software Engineering Sebastian Nanz 2010-10-20 This book focuses on defining the achievements of software engineering in the past decades and showcasing visions for the future. It features a collection of articles by some of the most prominent researchers and technologists who have shaped the field: Barry Boehm, Manfred Broy, Patrick Cousot, Erich Gamma, Yuri Gurevich, Tony Hoare, Michael A. Jackson, Rustan Leino, David L. Parnas, Dieter Rombach, Joseph Sifakis, Niklaus Wirth, Pamela Zave, and Andreas Zeller. The contributed articles reflect the authors' individual views on what constitutes the most important issues facing software development. Both research- and technology-oriented contributions are included. The book provides at the same time a record of a symposium held at ETH Zurich on the occasion of Bertrand Meyer's 60th birthday.

Software Engineering Ian Sommerville 2018-09-21

Vom Mythos des Mann-Monats Frederick P. Brooks 2019-05-08 Nur wenige Bücher über das Projektmanagement bei Software haben sich als so einflussreich und zeitlos gültig erwiesen wie "Vom Mythos des Mann-Monats": Fred Brooks bietet hier mit einem Mix aus harten Fakten und provokanten Ideen jedem tiefe Einsichten, der komplexe Projekte zu managen hat. Die Essays in diesem Buch stellen die Quintessenz seiner Erfahrungen als Projektmanager erst für die Hardware der IBM/360-Computerfamilie, dann als Leiter der Entwicklung des - wahrhaft gigantischen - Betriebssystemes OS/360 dar. Die Besonderheit dieses Buches liegt aber auch darin, dass Brooks, 20 Jahre nach Erscheinen des Originals, seine ursprünglichen Vorstellungen und Visionen noch einmal überdacht und sie um neue Erkenntnisse und Ratschläge bereichert hat. Dieses Buch ist ein Muss sowohl für Kenner seiner Arbeiten als auch Leser, die Brooks nun zum ersten Mal entdecken.

Parallel Agile – faster delivery, fewer defects, lower cost Doug Rosenberg 2020-01-03 From the beginning of software time, people have wondered why it isn't possible to accelerate software projects by simply adding staff. This is sometimes known as the "nine women can't make a baby in one month" problem. The most famous treatise declaring this to be impossible is Fred Brooks' 1975 book *The Mythical Man-Month*, in which he declares that "adding more programmers to a late software project makes it later," and indeed this has proven largely true over the decades. Aided by a domain-driven code generator that quickly creates database and API code, Parallel Agile (PA) achieves significant schedule compression using parallelism: as many developers as necessary can independently and concurrently develop the scenarios from initial prototype through production code. Projects can scale by elastic staffing, rather than by stretching schedules for larger development efforts. Schedule compression with a large team of developers working in parallel is analogous to hardware acceleration of compute problems using parallel CPUs. PA has some similarities with and differences from other Agile approaches. Like most Agile methods, PA "gets to code early" and uses feedback from executable software to drive requirements and design. PA uses technical prototyping as a risk-mitigation strategy, to help sanity-check requirements for feasibility, and to evaluate different technical architectures and technologies. Unlike many Agile methods, PA does not support "design by refactoring," and it doesn't drive designs from unit tests. Instead, PA uses a minimalist UML-based design approach (Agile/CONIX) that starts out with a domain model to facilitate communication across the development team, and partitions the system along use case boundaries, which enables parallel development. Parallel Agile is fully compatible with the Incremental Commitment Spiral Model (ICSM), which involves concurrent effort of a systems engineering team, a development team, and a test team working alongside the developers. The authors have been researching and refining the PA process for several years on multiple test projects that have involved over 200 developers. The book's example project details the design of one of these test projects, a crowdsourced traffic safety system.

A Review of the Next Generation Air Transportation System National Research Council 2015-06-17 The Next Generation Air Transportation System's (NextGen) goal is the transformation of the U.S. national airspace system through programs and initiatives that could make it possible to shorten routes, navigate better around weather, save time and fuel, reduce delays, and improve capabilities for monitoring and managing of aircraft. A Review of the Next Generation Air Transportation provides an overview of NextGen and examines the technical activities, including human-system design and testing, organizational design, and other safety and human factor aspects of the system, that will be necessary to successfully transition current and planned modernization programs to the future system. This report assesses technical, cost, and schedule risk for the software development that will be necessary to achieve the expected benefits from a highly automated air traffic management system and the implications for ongoing modernization projects. The recommendations of this report will help the Federal Aviation Administration anticipate and respond to the challenges of implementing NextGen.