

## Effort Estimation Techniques In Software Engineering

The integration of AI with software is an essential enabler for science and the new economy, creating new markets and opportunities for a more reliable, flexible and robust society. Current software methodologies, tools and techniques often fall short of expectations, however, and much software remains insufficiently robust and reliable for a constantly changing and evolving market. This book presents 54 papers delivered at the 20th edition of the International Conference on New Trends in Intelligent Software Methodology Tools, and Techniques (SoMeT\_21), held in Cancun, Mexico, from 21–23 September 2021. The aim of the conference was to capture the essence of a new state-of-the-art in software science and its supporting technology and to identify the challenges that such a technology will need to master, and this book explores the new trends and theories illuminating the direction of development in this field as it heads towards a transformation in the role of software and science integration in tomorrow's global information society. The 54 revised papers were selected for publication by means of a rigorous review process involving 3 or 4 reviewers for each paper, followed by selection by the SoMeT\_21 international reviewing committee. The book is divided into 9 chapters, classified by paper topic and relevance to the chapter theme. Covering topics ranging from research practices, techniques and methodologies to proposing and reporting on the solutions required by global business, the book offers an opportunity for the software science community to consider where they are today and where they are headed in the future. This volume constitutes the refereed proceedings of the 18th EuroSPI conference, held in Roskilde, Denmark, in June 2011. The 18 revised full papers presented together with 9 key notes were carefully reviewed and selected. They are organized in topical sections on SPI and assessments; SPI and implementation; SPI and improvement methods; SPI organization; SPI people/ teams; SPI and reuse; selected key notes for SPI implementation.

This book constitutes the first part of refereed proceedings of the 5th Computational Methods in Systems and Software 2021 (CoMeSySo 2021). The CoMeSySo 2021 Conference is breaking the barriers, being held online. CoMeSySo 2021 intends to provide an international forum for the discussion of the latest high-quality research results. The software engineering, computer science, and artificial intelligence are crucial topics for the research within an intelligent systems problem domain.

CD-ROM includes: Video introduction -- Book overview -- COCOMO II. 2000 software -- Tutorials -- Adobe Acrobat Reader installation package.

Software effort estimation is one of the oldest and most important problems in software project management, and thus today there are a large number of models, each with its own unique strengths and weaknesses in general, and even more importantly, in relation to the environment and context in which it is to be applied. Trendowicz and Jeffery present a comprehensive look at the principles of software effort estimation and support software practitioners in systematically selecting and applying the most suitable effort estimation approach. Their book not only presents what approach to take and how to apply and improve it, but also explains why certain approaches should be used in specific project situations. Moreover, it explains popular estimation methods,

## Get Free Effort Estimation Techniques In Software Engineering

summarizes estimation best-practices, and provides guidelines for continuously improving estimation capability. Additionally, the book offers invaluable insights into project management in general, discussing issues including project trade-offs, risk assessment, and organizational learning. Overall, the authors deliver an essential reference work for software practitioners responsible for software effort estimation and planning in their daily work and who want to improve their estimation skills. At the same time, for lecturers and students the book can serve as the basis of a course in software processes, software estimation, or project management.

Software Project Effort Estimation Foundations and Best Practice Guidelines for Success Springer

To support the broadening spectrum of project delivery approaches, PMI is offering A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Sixth Edition as a bundle with its latest, the Agile Practice Guide. The PMBOK® Guide – Sixth Edition now contains detailed information about agile; while the Agile Practice Guide, created in partnership with Agile Alliance®, serves as a bridge to connect waterfall and agile. Together they are a powerful tool for project managers. The PMBOK® Guide – Sixth Edition – PMI's flagship publication has been updated to reflect the latest good practices in project management. New to the Sixth Edition, each knowledge area will contain a section entitled Approaches for Agile, Iterative and Adaptive Environments, describing how these practices integrate in project settings. It will also contain more emphasis on strategic and business knowledge—including discussion of project management business documents—and information on the PMI Talent Triangle™ and the essential skills for success in today's market. Agile Practice Guide has been developed as a resource to understand, evaluate, and use agile and hybrid agile approaches. This practice guide provides guidance on when, where, and how to apply agile approaches and provides practical tools for practitioners and organizations wanting to increase agility. This practice guide is aligned with other PMI standards, including A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Sixth Edition, and was developed as the result of collaboration between the Project Management Institute and the Agile Alliance.

"This reference is a broad, multi-volume collection of the best recent works published under the umbrella of computer engineering, including perspectives on the fundamental aspects, tools and technologies, methods and design, applications, managerial impact, social/behavioral perspectives, critical issues, and emerging trends in the field"--Provided by publisher.

Computational Intelligence Techniques and Their Applications to Software Engineering Problems focuses on computational intelligence approaches as applicable in varied areas of software engineering such as software requirement prioritization, cost estimation, reliability assessment, defect prediction, maintainability and quality prediction, size estimation, vulnerability prediction, test case selection and prioritization, and much more. The concepts of expert systems, case-based reasoning, fuzzy logic, genetic algorithms, swarm computing, and rough sets are introduced with their applications in software engineering. The field of knowledge discovery is explored using neural networks and data mining techniques by determining the underlying and hidden patterns in software data sets. Aimed at graduate students and researchers in computer science engineering, software engineering, information technology, this book: Covers various aspects of in-depth solutions of software engineering problems

## Get Free Effort Estimation Techniques In Software Engineering

using computational intelligence techniques Discusses the latest evolutionary approaches to preliminary theory of different solve optimization problems under software engineering domain Covers heuristic as well as meta-heuristic algorithms designed to provide better and optimized solutions Illustrates applications including software requirement prioritization, software cost estimation, reliability assessment, software defect prediction, and more Highlights swarm intelligence-based optimization solutions for software testing and reliability problems

An effective, quantitative approach for estimating and managing software projects How many people do I need? When will the quality be good enough for commercial sale? Can this really be done in two weeks? Rather than relying on instinct, the authors of *Software Measurement and Estimation* offer a new, tested approach that includes the quantitative tools, data, and knowledge needed to make sound estimations. The text begins with the foundations of measurement, identifies the appropriate metrics, and then focuses on techniques and tools for estimating the effort needed to reach a given level of quality and performance for a software project. All the factors that impact estimations are thoroughly examined, giving you the tools needed to regularly adjust and improve your estimations to complete a project on time, within budget, and at an expected level of quality. This text includes several features that have proven to be successful in making the material accessible and easy to master:

- \* Simple, straightforward style and logical presentation and organization enables you to build a solid foundation of theory and techniques to tackle complex estimations
- \* Examples, provided throughout the text, illustrate how to use theory to solve real-world problems
- \* Projects, included in each chapter, enable you to apply your newfound knowledge and skills
- \* Techniques for effective communication of quantitative data help you convey your findings and recommendations to peers and management

*Software Measurement and Estimation: A Practical Approach* allows practicing software engineers and managers to better estimate, manage, and effectively communicate the plans and progress of their software projects. With its classroom-tested features, this is an excellent textbook for advanced undergraduate-level and graduate students in computer science and software engineering. An Instructor Support FTP site is available from the Wiley editorial department.

"This book assists its readers in recommending formulation of ICT strategies for e-government implementation and maintenance from the perspective of acknowledging the importance of e-Governance for building institutions to achieve transparency and accountability, and eventually democratic governance"--Provided by publisher.

Planning of software projects has become complex and challenging more and more as time has passed. Software project managers are dealing with issues such as rapid change of customer demands, dynamic advancements of hardware platforms and intangible nature of software. Development effort estimation plays a vital role in software project planning because accurate estimation is needed when dealing with the complexity of software projects. Numerous researches

have been conducted to improve the accuracy of development effort estimation in recent years and various methods have been proposed in this field. Extensive diversity of existing methods and the lack of unified investigation of which, have inspired us to write this book. We have tried to collect the most significant studies in terms of software development effort estimation based on the latest findings in this area. It is expected that this book can be an initial reference for researchers who want to work on the mentioned field. This initial reference might help researchers to have a simple overview of available effort estimation methods in software projects.

This book constitutes the refereed proceedings of two joint events: the 25th International Workshop on Software Measurement (IWSM) and the 10th International Conference on Software Process and Product Measurement (Mensura), referred to as IWSM?Mensura 2015 and held in Kraków, Poland, in October 2015. Software measurement is a key methodology in estimating, managing, and controlling software development and management projects. The 13 papers presented in this volume were carefully reviewed and selected from 32 submissions. They present various theoretical and empirical results related to software measurement and its application in industrial projects.

Two issues continually plaguing the software industry are software size calculation and project effort estimation. Incorrect estimates may lead to inappropriate allocation of resources (people), shortage of time, insufficient funds, and possibly project failure. The purpose of this study is to explore current methods of software effort estimation and the applicability to low-code application development. The study seeks to answer the research question: Can the current methods be used to estimate effort for applications built with low-code platforms? The goal is to analyze some of the popular estimation methods to see if they can be applied to low-code application development.

Project estimating plays a vital role in project management. Typically completed in the initial planning stages, accurate project estimation can be a difficult task. Organizations and project managers should use these initial estimates to baseline the project schedule and cost, then refine these estimates as the project develops. Accurate estimation and refinement of the estimates leads to better and earlier decision making, thus maximizing value. Developed within the framework of A Guide to the Project Management Body of Knowledge (PMBOK® Guide) &– Sixth Edition and other PMI standards, the Practice Standard for Project Estimating &– Second Edition focuses on providing models for the project management profession in both plan-driven and change-driven adaptive (agile) life cycles. This practice standard describes the aspects of project estimating that are recognized as good practice on most projects most of the time and that are widely recognized and consistently applied. PMI practice standards describe processes, activities, constraints, inputs, and outputs for specific discipline subject areas and are targeted to all practitioners within projectized organizations, not just project managers.

## Get Free Effort Estimation Techniques In Software Engineering

Businesses today are faced with a highly competitive market and fast-changing technologies. In order to meet demanding customers' needs, they rely on high quality software. A new field of study, soft computing techniques, is needed to estimate the efforts invested in component-based software. *Component-Based Systems: Estimating Efforts Using Soft Computing Techniques* is an important resource that uses computer-based models for estimating efforts of software. It provides an overview of component-based software engineering, while addressing uncertainty involved in effort estimation and expert opinions. This book will also instruct the reader how to develop mathematical models. This book is an excellent source of information for students and researchers to learn soft computing models, their applications in software management, and will help software developers, managers, and those in the industry to apply soft computing techniques to estimate efforts.

*Unbiased Stereology, Second Edition* is a practical guide to making unbiased 3-D measurements via the microscope. Only those stereological techniques which have been tried and tested by real application are included. Although this technology is essentially mathematical and statistical, the authors do not immerse the reader in complex analysis, but rather provide simple heuristic explanations and references to the original proof, and illustrate the theory by analogies drawn from everyday experience. To give practical experience in application of the techniques, exercises are provided at the end of each chapter, complete with detailed worked answers.

Designed for system developers, project managers and teachers of software engineering and related topics. This guide to using macro-estimation techniques to improve estimations of software development effort and duration includes a diskette of project data and estimating tools. Topics covered include deriving the size of a project and creating an estimating framework.

This is the first handbook to cover comprehensively both software engineering and knowledge engineering -- two important fields that have become interwoven in recent years. Over 60 international experts have contributed to the book. Each chapter has been written in such a way that a practitioner of software engineering and knowledge engineering can easily understand and obtain useful information. Each chapter covers one topic and can be read independently of other chapters, providing both a general survey of the topic and an in-depth exposition of the state of the art. Practitioners will find this handbook useful when looking for solutions to practical problems. Researchers can use it for quick access to the background, current trends and most important references regarding a certain topic. The handbook consists of two volumes. Volume One covers the basic principles and applications of software engineering and knowledge engineering. Volume Two will cover the basic principles and applications of visual and multimedia software engineering, knowledge engineering, data mining for software knowledge, and emerging topics in software engineering and knowledge engineering.

## Get Free Effort Estimation Techniques In Software Engineering

Effort estimation is used for planning and managing all phases of software development. Traditional estimation methods rely on having a mostly ?complete? and ?fixed? specification of a system. In agile software development, cost and effort estimation is challenging, as the requirements cannot be specified entirely upfront and are evolved as development progress and users interact with the product. Current practice in this context relies heavily on human judgment which has several limitations. It is based on limited information (subjective opinion), hampered by wishful thinking and prone to human judgment bias (individual and group effects). Estimates produced by these methods are often inaccurate. Experts recognize the need and importance of more objective information that includes historical data from previous iterations. Moreover, current estimation methods do not objectively consider the potential impact of a change on existing software and custom factors that contribute to the effort overhead. Thus, considerable improvement potential exists concerning systematic effort estimation in this environment and marks the contribution of this research to the body of knowledge.

This book features selected research papers presented at the International Conference on Advances in Information Communication Technology and Computing (AICTC 2019), held at the Government Engineering College Bikaner, Bikaner, India, on 8–9 November 2019. It covers ICT-based approaches in the areas ICT for energy efficiency, life cycle assessment of ICT, green IT, green information systems, environmental informatics, energy informatics, sustainable HCI and computational sustainability.

"If you're looking for solid, easy-to-follow advice on estimation, requirements gathering, managing change, and more, you can stop now: this is the book for you."--Scott Berkun, Author of *The Art of Project Management* What makes software projects succeed? It takes more than a good idea and a team of talented programmers. A project manager needs to know how to guide the team through the entire software project. There are common pitfalls that plague all software projects and rookie mistakes that are made repeatedly--sometimes by the same people! Avoiding these pitfalls is not hard, but it is not necessarily intuitive. Luckily, there are tried and true techniques that can help any project manager. In *Applied Software Project Management*, Andrew Stellman and Jennifer Greene provide you with tools, techniques, and practices that you can use on your own projects right away. This book supplies you with the information you need to diagnose your team's situation and presents practical advice to help you achieve your goal of building better software. Topics include: Planning a software project Helping a team estimate its workload Building a schedule Gathering software requirements and creating use cases Improving programming with refactoring, unit testing, and version control Managing an outsourced project Testing software Jennifer Greene and Andrew Stellman have been building software together since 1998. Andrew comes from a programming background and has managed teams of requirements analysts, designers, and developers. Jennifer has a testing background and has managed teams of architects, developers, and testers. She has led multiple large-scale outsourced projects. Between the two of them, they have managed every aspect of software development. They have worked in a wide range of industries, including finance, telecommunications, media, nonprofit, entertainment, natural-language processing, science, and academia. For more information about them and this book, visit [Page 6/12](http://stellman-</a></p></div><div data-bbox=)

greene.com

Effort estimation is a key factor for software project success, defined as delivering software of agreed quality and functionality within schedule and budget. Traditionally, effort estimation has been used for planning and tracking project resources. Effort estimation methods that grew upon those objectives focus on providing exact estimates and usually do not support systematic and reliable analysis of the causal effort dependencies. Moreover, existing estimation methods are typically based either on large data sets or on the extensive involvement of domain experts (human expertise), which, in practice, significantly reduces their applicability in software industry. In order to handle those problems the thesis proposes a WelCoMe method that integrates data analysis and human judgment to extracting context-specific causal effort dependencies. When applied in the context of two industrial companies WelCoMe contributed to 17% reduction in cost of building an effort model and 50% reduction in complexity of a resulting model, while increasing its predictive performance by 43%-56%. Moreover, proposed estimation method allowed identifying crucial improvement potentials with respect to organizational measurement processes that had not been identified by domain experts.

This book constitutes the thoroughly refereed post-conference proceedings of the Second IFIP TC 2 Central and East-European Conference on Software Engineering Techniques, CEE-SET 2008, held in Brno, Czech Republic, in October 2008. The 20 revised full papers presented together with a keynote speech were carefully reviewed and selected from 69 initial submissions. The papers are organized in topical sections on requirements specification, design, modeling, software product lines, code generation, project management, and quality.

This book presents high-quality, original contributions (both theoretical and experimental) on Information Security, Machine Learning, Data Mining and Internet of Things (IoT). It gathers papers presented at ICETIT 2019, the 1st International Conference on Emerging Trends in Information Technology, which was held in Delhi, India, in June 2019. This conference series represents a targeted response to the growing need for research that reports on and assesses the practical implications of IoT and network technologies, AI and machine learning, data analytics and cloud computing, security and privacy, and next generation computing technologies.

Applying methodologies of Software Process Improvement (SPI) is an effective way for businesses to remain competitive in the software industry. However, many organizations find implementing software process initiatives challenging. Agile Estimation Techniques and Innovative Approaches to Software Process Improvement reviews current SPI techniques and applications through discussions on current and future trends as well as the presentation of case studies on SPI implementation. Ideal for use by academics, students, and policy-makers, as well as industry professionals and managers, this publication provides a complete overview of current tools and methodologies regarding Software Process Improvement.

Product verifiable, defensible, and achievable software estimates Based on data collected by the International Software Benchmarking Standards Group (ISBSG), Practical Software Project Estimation explains how to accurately forecast the size, cost,

and schedule of software projects. Get expert advice on generating accurate estimates, minimizing risks, and planning and managing projects. Valuable appendixes provide estimation equations, delivery rate tables, and the ISBSG Repository demographics. Verify project objectives and requirements Determine, validate, and refine software functional size Produce indicative estimates using regression equations Predict effect and duration through comparison and analogy Build estimation frameworks Perform benchmarks using the ISBSG Repository Compare IFPUG, COSMIC, and FiSMA sizing methods Peter Hill is the chief executive officer and a director of the ISBSG. He has been in the information services industry for more than 40 years and has compiled and edited five books for the ISBSG.

This volume constitutes the refereed proceedings of the 18th International Conference on Software Process Improvement and Capability Determination, SPICE 2018, held in Tessaaloniki, Greece, in October 2018. The 26 full papers presented were carefully reviewed and selected from 40 submissions. The papers are organized in the following topical sections: SPI systematic literature reviews; SPI and assessment; SPI methods and reference models; SPI education and management issues; SPI knowledge and change processes; SPI compliance and configuration; SPI and agile; industry short papers.

The ecological footprint of our every day activities has become one of the greatest problems facing mankind today. Reducing the environmental footprint of IT, which has become a major contributor to green house gas emissions in the last decade, is therefore a major research challenge for both industry and academia. The focus of this book is to provide the foundation for expressing and estimating ecological costs of software systems in all life-cycle phases including the earliest phases of software development, thus helping all stakeholders make well-informed choices. In particular, common eco-cost drivers as well as new metrics are defined which allow meaningful and precise descriptions of ecological costs. Based on these concepts, auxiliary models are introduced which allow ecological costs and the circumstances causing them to be expressed in a unified, unambiguous, and easy-to-understand way. These models provide input to algorithmic eco-cost estimation models which show the impact of development decisions on ecological costs even in early phases of software development. Both the estimation and auxiliary models accommodate different input, allowing them to deliver the most accurate eco-cost estimates possible based on the information available. The approach presented in this book can therefore be used as the fulcrum of green software engineering methods to explore the eco-costs of applications' features and architectures at early stages in the development process before significant resources have been invested in their implementation. It therefore facilitates the creation of eco-friendlier software systems and by that contributing to the solution of one of the greatest problems facing mankind today.

Almost every software project begins with the utterances, "What will this cost?" and "When will this project be done?" Once those words are spoken, project stakeholders begin to wrestle with how to produce an estimate. Accurately estimating the cost or time to complete a software project is a serious problem for many software engineers, developers and project managers who struggle with costs running double original estimates, putting their careers at risk. It is reported that nearly 50% of all software projects are shelved and that one of the major causes is poor estimation practices. If developing software for internal use, poor estimates can represent a significant drain on corporate profits.

Worldwide growth in the number of companies specializing in the development of software for use by other companies is staggering. India alone has nearly 20,000 such companies. Intense competition has led to an increased demand for fixed-bid pricing in client/vendor relationships, and has made effective cost estimation even more important and, in many cases, critical to a firm's survival. There are many

## Get Free Effort Estimation Techniques In Software Engineering

methods of estimation. Each method has its strengths and weaknesses, proponents and opponents. Knowing how and which one to use on a given project is key to developing acceptable estimates for either internal or external projects. Software Estimation Best Practices, Tools, & Techniques covers all facets of software estimation. It provides a detailed explanation of the various methods for estimating software size, development effort, cost, and schedule, including a comprehensive explanation of Test Effort Estimation. Emphasizing that software estimation should be based on a well-defined process, it presents software estimation best practices and shows how to avoid common pitfalls. This guide offers direction on which methods are most appropriate for each of the different project types commonly executed in the software development space and criteria for selecting software estimation tools. This comprehensive desk reference explains software estimation from scratch to help the beginner and features advanced techniques for more experienced estimators. It details project scheduling, including resource leveling and the concept of productivity, as applicable to software estimators, demonstrating the many benefits of moving from the current macro-productivity approach to a micro-productivity approach in software estimation. Software Estimation Best Practices, Tools, & Techniques: A Complete Guide for Software Project Estimators caters to the needs of all software project stakeholders, from novice to expert. It provides the valuable guidance needed to estimate the cost and time required to complete software projects within a reasonable margin of error for effective software development.

Agile Estimating and Planning is the definitive, practical guide to estimating and planning agile projects. In this book, Agile Alliance cofounder Mike Cohn discusses the philosophy of agile estimating and planning and shows you exactly how to get the job done, with real-world examples and case studies. Concepts are clearly illustrated and readers are guided, step by step, toward how to answer the following questions: What will we build? How big will it be? When must it be done? How much can I really complete by then? You will first learn what makes a good plan-and then what makes it agile. Using the techniques in Agile Estimating and Planning, you can stay agile from start to finish, saving time, conserving resources, and accomplishing more. Highlights include: Why conventional prescriptive planning fails and why agile planning works How to estimate feature size using story points and ideal days—and when to use each How and when to re-estimate How to prioritize features using both financial and nonfinancial approaches How to split large features into smaller, more manageable ones How to plan iterations and predict your team's initial rate of progress How to schedule projects that have unusually high uncertainty or schedule-related risk How to estimate projects that will be worked on by multiple teams Agile Estimating and Planning supports any agile, semiagile, or iterative process, including Scrum, XP, Feature-Driven Development, Crystal, Adaptive Software Development, DSDM, Unified Process, and many more. It will be an indispensable resource for every development manager, team leader, and team member.

This book empowers readers to know the thought-provoking field of software effort estimation. It discusses how requirement change effort estimation using algorithmic and impact analysis techniques is used to optimize the estimation accuracy prediction of software development effort. It is a worthy read for researchers and practitioners to estimate the change effort required to develop traditional and agile-based software systems.

This book features selected papers presented at the Fifth International Conference on Nanoelectronics, Circuits and Communication Systems (NCCS 2019). It covers a range of topics, including nanoelectronic devices, microelectronics devices, material science, machine learning, Internet of things, cloud computing, computing systems, wireless communication systems, advances in communication 5G and beyond. Further, it discusses VLSI circuits and systems, MEMS, IC design and testing, electronic system design and manufacturing, speech signal processing, digital signal processing, FPGA-based wireless communication systems and FPGA-based system design, Industry 4.0, e-

## Get Free Effort Estimation Techniques In Software Engineering

farming, semiconductor memories, and IC fault detection and correction.

"A clearly written book that is a useful primer for a very complicated set of topics." --Capers Jones, Chief Scientist Emeritus, Software Productivity Research LLC

Practical Software Estimation brings together today's most valuable tips, techniques, and best practices for accurately estimating software project efforts, costs, and schedules. Written by a leading expert in the field, it addresses the full spectrum of real-world challenges faced by those who must develop reliable estimates. M. A. Parthasarathy draws on the immense experience of Infosys, one of the world's largest and most respected providers of IT-enabled business solutions, to bring you the only book with detailed guidance on estimating insourced and outsourced software projects, as well as projects that blend both approaches. He demonstrates how to successfully utilize Function Point (FP) methods, the industry's leading estimation model. Then, using real case studies, he systematically identifies pitfalls that can lead to inaccurate estimates--and offers proven solutions. Coverage includes

- How to estimate all types of software projects, including "fresh" development, reengineering, and maintenance
- How to incorporate the impact of core project elements on estimates: scope, environment, experience, and tools
- FP analysis from start to finish: data and transaction functions, general system characteristics, and more FP methods for any platform or business function
- Innovative re-estimation methods to track progress
- How to quote RFPs and prepare contracts: fixed price, time/material, and project execution lifecycle models
- Alternatives to FP: Delphi, COCOMO II, and COSMIC-FFP
- How to choose the right estimation tools

Practical Software Estimation is the definitive reference for anyone who must estimate software projects accurately: project and IT managers, individual developers, system designers, architects, executives, consultants, and outsourcers alike.

List of Figures  
List of Tables  
Foreword  
Preface  
Acknowledgments  
Chapter 1: Introduction  
Chapter 2: Role of Estimation in Software Projects  
Chapter 3: A Study of Function Point Analysis  
Chapter 4: Data Functions  
Chapter 5: Transactional Functions  
Chapter 6: General System Characteristics  
Chapter 7: Size, Effort, and Scheduling of Projects  
Chapter 8: Estimation Flavors  
Chapter 9: A Sense of Where You Are  
Chapter 10: Tips, Tricks, and Traps  
Chapter 11: Insourcing versus Outsourcing  
Chapter 12: Key Factors in Software Contracts  
Chapter 13: Project Estimation and Costing  
Chapter 14: Other Estimation Methods  
Chapter 15: Estimation Tools  
Chapter 16: Estimation Case Study  
Appendix A: Reference Tables: Transaction Function Counts  
Appendix B: Reference Tables: Data Function Points  
Bibliography  
Index

Estimating software development often produces more angst than value, but it doesn't have to. Identify the needs behind estimate requests and determine how to meet those needs simply and easily. Choose estimation techniques based on current needs and available information, gaining benefit while reducing cost and effort. Detect bad assumptions that might sink your project if you don't adjust your plans. Discover what to do when an estimate is wrong, how to recover, and how to use that knowledge for future planning. Learn to communicate about estimates in a healthy and productive way, maximizing advantage to the organization and minimizing damage to the people. In a world where most developers hate estimation and most managers fear disappointment with the results, there is hope for both. It requires giving up some widely held misconceptions. Let go of the notion that "an estimate is an estimate" and estimate for the particular need you, and your organization, have. Realize that estimates have a limited shelf-life, and reestimate frequently if it's important. When reality differs from your estimate, don't lament; mine that disappointment for the gold that can be the longer-term jackpot. Estimate in comparison to past experience, by modeling the work mathematically, or a hybrid of both. Learn strategies for effective decomposition of work and aspects of the work that likely affect your estimates. Hedge your bets by comparing the results of different approaches. Find out what to do when an estimate proves wrong. And they will. They're estimates, after all. You'll discover that you can use estimates to warn you of danger so you can take

## Get Free Effort Estimation Techniques In Software Engineering

appropriate action in time. Learn some crucial techniques to understand and communicate with those who need to understand. Address both the technical and sociological aspects of estimation, and you'll help your organization achieve its desired goals with less drama and more benefit. What You Need: No software needed, just your past experience and concern for the outcomes.

A collection of previously published articles from a variety of publications.

The development of software has expanded substantially in recent years. As these technologies continue to advance, well-known organizations have begun implementing these programs into the ways they conduct business. These large companies play a vital role in the economic environment, so understanding the software that they utilize is pertinent in many aspects. Researching and analyzing the tools that these corporations use will assist in the practice of software engineering and give other organizations an outline of how to successfully implement their own computational methods. *Tools and Techniques for Software Development in Large Organizations: Emerging Research and Opportunities* is an essential reference source that discusses advanced software methods that prominent companies have adopted to develop high quality products. This book will examine the various devices that organizations such as Google, Cisco, and Facebook have implemented into their production and development processes. Featuring research on topics such as database management, quality assurance, and machine learning, this book is ideally designed for software engineers, data scientists, developers, programmers, professors, researchers, and students seeking coverage on the advancement of software devices in today's major corporations.

Often referred to as the "black art" because of its complexity and uncertainty, software estimation is not as difficult or puzzling as people think. In fact, generating accurate estimates is straightforward—once you understand the art of creating them. In his highly anticipated book, acclaimed author Steve McConnell unravels the mystery to successful software estimation—distilling academic information and real-world experience into a practical guide for working software professionals. Instead of arcane treatises and rigid modeling techniques, this guide highlights a proven set of procedures, understandable formulas, and heuristics that individuals and development teams can apply to their projects to help achieve estimation proficiency. Discover how to: Estimate schedule and cost—or estimate the functionality that can be delivered within a given time frame Avoid common software estimation mistakes Learn estimation techniques for you, your team, and your organization \* Estimate specific project activities—including development, management, and defect correction Apply estimation approaches to any type of project—small or large, agile or traditional Navigate the shark-infested political waters that surround project estimates When many corporate software projects are failing, McConnell shows you what works for successful software estimation.

Having realistic estimates of effort at an early stage in a Web project's life is vital to the successful management of

resources. The principles of the prediction process are identifying the influencing factors, gathering past project data, generating an effort prediction model, and assessing the effectiveness of such prediction model. Cost Estimation Techniques for Web Projects provides a step-by-step methodology to improving cost estimation practices for Web projects. Utilizing such techniques as stepwise regression modeling, case-base reasoning, classification and regression trees, and expert opinion, this book is a powerful tool for scholars, researchers, and practitioners in the areas of Web development, Web engineering, project management, and software engineering.

Data Science for Software Engineering: Sharing Data and Models presents guidance and procedures for reusing data and models between projects to produce results that are useful and relevant. Starting with a background section of practical lessons and warnings for beginner data scientists for software engineering, this edited volume proceeds to identify critical questions of contemporary software engineering related to data and models. Learn how to adapt data from other organizations to local problems, mine privatized data, prune spurious information, simplify complex results, how to update models for new platforms, and more. Chapters share largely applicable experimental results discussed with the blend of practitioner focused domain expertise, with commentary that highlights the methods that are most useful, and applicable to the widest range of projects. Each chapter is written by a prominent expert and offers a state-of-the-art solution to an identified problem facing data scientists in software engineering. Throughout, the editors share best practices collected from their experience training software engineering students and practitioners to master data science, and highlight the methods that are most useful, and applicable to the widest range of projects. Shares the specific experience of leading researchers and techniques developed to handle data problems in the realm of software engineering Explains how to start a project of data science for software engineering as well as how to identify and avoid likely pitfalls Provides a wide range of useful qualitative and quantitative principles ranging from very simple to cutting edge research Addresses current challenges with software engineering data such as lack of local data, access issues due to data privacy, increasing data quality via cleaning of spurious chunks in data

This book presents new approaches and methods to solve real-world problems as well as exploratory research describing novel approaches in the field of software engineering and intelligent systems. It particularly focuses on modern trends in selected fields of interest, introducing new algorithms, methods and application of intelligent systems in software engineering. The book constitutes the refereed proceedings of the Software Engineering Trends and Techniques in Intelligent Systems Section of the 6th Computer Science On-line Conference 2017 (CSOC 2017), held in April 2017.

[Copyright: ae15da28761e78496fb2a1c876649cba](https://doi.org/10.1007/978-1-4939-9876-1)