Preface

This book is the blend of a software project cost/schedule expert (Dan Galorath) with a software risk expert's (Mike Evans) work. This book comes from years of experience in software project management, and building and applying tools to understand and manage software cost, schedule, and risk. Galorath's disciplines of project planning and control fit like a glove with the risk management and project assessment approaches of Evans.

How This Book Came about from a Galorath Viewpoint

From a Galorath viewpoint, this book is a product of a commitment to better software project planning and control. Many years ago a project was never developed because of my realistic, but politically incorrect estimate. I wondered if I had done the right thing, which was try to provide an achievable project plan and a realistic estimate, even though that estimate was longer than the company desired. This experience made me wonder if I had failed as a manager. However, some years later I tried to reproduce the estimate using SEER-SEM and discovered that I had significantly underestimated the project. SEER-SEM enabled me to understand that I had not failed and that my refusal to give in to the division head's pressure had been the best thing for that company.

This experience was the beginning of my mission to understand how long software development should take and how many people are needed. A few years later Don Reifer and I were tasked with developing a concept for software estimation within NASA JPL. The results of the study prompted the development of the JPL Softcost software estimation model.*

* Tausworth, R. *Softcost*. Pasadena: Jet Propulsion Laboratory, 1981.

In 1981, when Dr. Barry Boehm's classic book, *Software Engineering Economics*, was published, I was operating a consulting business that remedied failing projects. I still used the early Softcost model which was automated, and performed risk and trade-off analysis. In 1984 I began consulting for Computer Economics, Inc., where I met Dr. Randall Jensen and was introduced to the "Jensen model." This model had been implemented as the JS-2. I recognized its strengths, which were its ability to conduct uncertainty estimation and minimum time estimation, but from a product viewpoint it had challenges. After redesign, it was released as the CEI System-3, which was relatively successful in helping people answer the difficult questions surrounding software development.

In 1988 Galorath Incorporated began developing SEER-SEM, deciding to implement it under Microsoft Windows version 2 (version 2.03 to be exact). The initial product, which was approximately 22,000 lines of code, relied heavily on the mathematics that served as the foundation of Jensen's public domain model. We shipped a run-time version of Windows on 13, 5.25 in. floppies. DOS was pervasive at this time, and some people complained about being required to use Microsoft Windows, but this decision was critical to SEER-SEM's early success. At one time there were Macintosh and Sun Solaris versions of SEER-SEM as well, but both were based on the Windows code baseline.

As the state of software development progressed, and as user demand grew more sophisticated, SEER-SEM continually required more robust and advanced mathematics and refined knowledge bases to handle the situations our users found themselves confronting. Users wanted to know the answers to questions such as: "How do I plan with my real-world staffing constraints?"; "How do I estimate my COTS software?"; "Do risk and uncertainty affect schedule independently of effort?"; "How can I calibrate for special situations or to my particular environment?"; and a host of other situations. SEER-SEM has also evolved from using lines of code as the only size input to function points, developing its own function-based sizing and the ability to accommodate objects, Web pages, use cases, classes, and a host of other sizing methods.

Most software estimation models have common ancestors, and SEER-SEM is no different. SEER-SEM, which was based on the Jensen model developed at Hughes Aircraft Company, diverged significantly in the early 1990s. Earlier work by Doty Associates introduced the idea of factoring in development environment influences via parameters. Barry Boehm's COCOMO work contributed to the original Jensen model technology parameters and has been a valuable cross-check to the SEER work. Research with Don Reifer that stimulated work on the NASA Softcost model also influenced SEER-SEM, as did Halstead's software science

metrics and McCabe's complexity metrics. Today, through the process of Galorath Incorporated's "continuous product improvement," the SEER-SEM suite has grown to nearly 200,000 lines of code. Using this process, Galorath engineers analyze and begin to address software industry trends even before the trends become visible in the mainstream. The SEER models continue to evolve with data collection, application, research and development being conducted continuously.

At a high level, software sizing and estimation risks are the root of numerous project failures: unachievable commitments made by projects based on incorrect projections of cost, schedule, and resources. Thus Mike Evans entered the picture.

This book explores the various software sizing techniques, how these size projections are used to project cost and schedule, how projects can manage against these constraints and what risks result from constraints that are unrealistic or not achievable. This book describes how the risks can be managed and how tools, models, and other automated facilities can be used to enable better estimation and lower project risk.

In the chapters that follow, software sizing metrics and other software estimation factors are discussed. This book also covers the software estimation state of the practice, and leading trends and practices in software sizing technology.

While software size has the biggest impact on estimation, other factors such as technology, environment, complexity factors, staffing, scheduling, risk, and probability are also key. This book focuses on the techniques of the SEER family of cost, schedule risk, reliability models from Galorath Incorporated, as well as a brief look at the software project management life cycle and how software estimation makes a perfect management tool. The book is unique in that it doesn't stop at describing the sizing and estimation process but goes further in describing the core risks that result from the estimation process and commonly cause programs to not perform in accordance with the initial estimates. The specific methods for managing these risks, mitigating them, and getting the program completed successfully are provided.

Audience

This book is written for people who manage, engineer and assure software, stakeholders who need to understand software estimation techniques and metrics, and, finally, how to identify, manage, and mitigate estimate risks and their project impacts. The book is important to individual readers as follows:

- ***Senior Management*** —— Provides insight into the estimation process; the effects that management and corporate decisions, attitudes, and culture have on the integrity of estimates; and the risks that result from these estimates, which place systems and software intensive projects in jeopardy.
- ***Customers and Users*** —— Provides customers and users essential insight into how estimates are developed; and cultural factors and management attitudes, which could impact integrity of the estimate and the resulting risks.
- ***Stakeholders*** —— Provides insights into the processes and risks associated with software estimates and methods that can be used to monitor project progress and determine the risk of receiving a product that will meet the needs and expectations of the user developed within the cost and schedule constraints.
- ***Engineering Management*** —— Provides engineering management insight into the estimation process and how engineering decisions, shortcuts, trade studies and trade-offs, and other engineering factors affect the integrity of the estimate and the resultant risk to the project.
- ***Project Management*** —— Provides specific factors which can be used to understand how estimates are developed and the essential relationship that exists between the initial estimates and the need to perform at a productivity rate consistent with the estimate.
- ***System Engineers*** —— Provides specific factors to consider when establishing the essential system relationships and operational specifications that impact the estimate and increase the risk of meeting plans.
- ***Software Engineers*** —— Provides comprehensive information that software engineers can use to develop more accurate estimates, project resource requirements and schedule, and determine potential threats to software integrity, quality, and system effectiveness.
- ***Quality Assurance*** —— Provides specific information concerning the cost of adding various quality factors to the product, and the added costs of assuring the quality of the product or the effectiveness of the processes used.
- ***Test Personnel*** —— Identifies cost factors which should be considered when planning, implementing, or evaluating the results of a test program, and the estimation of defects insertion and removal.
- ***Risk Management Personnel*** —— Identifies specific risks related to estimates and describes risk management strategies that can be used to identify and control their impacts.
- ***Students*** —— Helps students understand how estimation and risk management are important to the complete understanding of the costs associated with management, engineering, assurance, and monitoring of software products and systems.

Structure of the Book

This book is divided into four major parts, which describe the estimation issues, the various estimation steps, the various metrics and special estimation cases, and the use of the SEER family of tools to assist in developing more accurate estimates faster.

Part I: Estimation Issues. Chapter 1 discusses specific issues that impact the validity of estimates and the difficulty of the process.
Part II: Estimation Process. Chapters 2, 3, and 4 discuss the various steps involved in developing and validating estimates.
Part III: Metrics and Special Estimation Cases. Chapters 5 through 9 discuss the various size measures and special estimation cases.
Part IV: Risk Management and Estimation Tools and Techniques. Chapters 10 through 12 discuss the risk management process and how the SEER family of tools can be used to expedite the estimation process and generate more accurate estimates.

What Can You Expect from the Book?

Before discussing what can be expected from the book it is important that it is clear what it is not. It is not an academic text describing estimation theories or concepts. It is a practical, hands-on discussion of critical factors and considerations that impact estimates, how to select and apply the appropriate measures to project and document size, and how to identify and manage risks. The authors, Mike Evans and Dan Galorath, are recognized experts in estimation, process engineering, and risk management. The book captures much of their unique experience providing practical solutions to many of the difficult problems that make estimates invalid or high risk. The information is presented in a way that will help readers identify and deal with project actions and attitudes that can result in an invalid estimate or high risk projects because of inadequate estimates. The content of the book will provide insight not readily available through other sources, which will help organizations recognize and avoid downstream impacts that can be caused by poor estimates.

Without a map (or GPS) there is no way to determine which way to go or how long it will take to get there. This book provides a map for successful software project planning and control — helping developers *plan* which way to go, how long it will take to get there, and minimize the side trips, detours, and flat tires. While writing this book, the motto for the text became: "Preparation precedes performance. When performance is measured performance improves."