SOFTWARE PROJECT ESTIMATION

INTELLIGENT FORECASTING, PROJECT CONTROL, AND CLIENT RELATIONSHIP MANAGEMENT

Dimitre Dimitrov

Apress®

Software Project Estimation: Intelligent Forecasting, Project Control, and Client Relationship Management

Dimitre Dimitrov
Toronto, ON, Canada

ISBN-13 (pbk): 978-1-4842-5024-2 ISBN-13 (electronic): 978-1-4842-5025-9
https://doi.org/10.1007/978-1-4842-5025-9

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Shiva Ramachandran
Development Editor: Rita Fernando
Coordinating Editor: Rita Fernando

Cover designed by eStudioCalamar

Author photo by Caroline Acton

To a friend.

"Invert, always invert"
—Carl Jacobi, 19th-century mathematician

Contents

About the Author

Dimitre Dimitrov is a software industry professional with 20 years of experience in project management, information systems development, and agile team coaching and facilitation. Dimitre has helped companies in a wide range of industries. He explores the forces and relationships that shape the lives of modern software development teams and their clients.

Acknowledgments

I am thankful to many of my colleagues for getting to this wonderful place of having written a book. I met some very bright programmers and managers who helped me see the right ways of approaching project planning and management—want to thank all the teams and clients I have worked with for being open to new ideas. I want to thank those who showed me the wrong ways as well.

And I want to thank my family, my lovely wife and girls, for having the chance to enjoy life together, and my sister and brother-in-law, for their deck, on which I wrote some of the pages of my book and felt as a true book author, basking in the summer sun and smelling the air of the Atlantic Ocean.

Special thanks to my editors and proofreaders. And special thanks to the owner of the Page One cafe in Toronto, whose place and atmosphere are inspiring many people to start their first page and where I claimed "I'm going to write a book."

Introduction

Why This Book?

Have you been on projects where halfway down the road it seems increasingly unlikely that you will finish as desired, but you can't put your finger on it and simply push through with a growing resentment? Have you been in meetings where scope discussions get increasingly difficult and depressing, ultimately sucking the energy out of everyone instead of enabling them to move forward with certainty and determination? Wouldn't it be good if a confident assertion for the project's ability to deliver is made as early as 1 or 2 months into the effort? What additional value and quality of working relationships can you generate if the time for estimation, tracking, and change management is slashed by a factor of 10 while simultaneously providing clients and team members with true peace of mind so they can focus on other important business activities? This book provides a practical tool that will help with all of this—it is a how-to book. But ultimately it is about the relationships we develop during projects and the appreciation of our colleagues and business partners.

It is 2019! Yet estimation and forecasting in the software development industry are still considered mystifying at best, and an archaic and obsolete concept at worst. Reliable forecasting continues to present a challenge for many teams and software development organizations. The practices associated with the two predominant software development methodologies are inadequate. Methods related to waterfall development are notoriously bad for long-term forecasting because they encourage too much information processing too early and have a tendency to skew reality into a Gantt chart. And methods that relate well to agile software development are not as notoriously bad, however, mostly because long-term forecasting is avoided altogether. This is problematic in many cases because it pushes important decisions too late in the project, adds unnecessary stress on people's relationships, and ultimately diminishes the chances for successful projects.

When forecasting is done on projects utilizing agile software development, it is often with short-term commitments—an iteration or a few at best. Such small-span commitments have minimal value for business people who want to look a year or two ahead. When teams practicing agile software development commit to a longer-term delivery, they suffer many of the same issues as people working on waterfall projects do—the desired "project targets" get missed, and there is lack of confidence throughout the project about what will be delivered and when. Senior businessmen and businesswomen lose the capacity to trust the delivery teams and organizations. People who work together for years remain at a distance and never build true partnerships. Many business teams and development teams are openly adversary.

One reason behind much of these issues is the inability for meaningful long-term commitment. And while reliably providing accurate and precise software estimates is close to impossible, reliably providing accurate and precise forecasts is not. This book shows you how to do this and commit to a purpose early. A satisfactory solution can be reached with a small investment in understanding people's problems and through the adoption of simple statistical principles. The security and reliability that you will be able to contribute to projects will improve your team's performance, morale, and motivation and will have a positive impact on long and healthy client relationships.

Who Is the Reader of This Book and What Can You Find in It?

Software Project Estimation is for anyone who wants to have a fresh and adequate outlook on the process of software estimation and forecasting, and how these activities facilitate the conversations and relationships among people. It is directly relevant to the roles of scrum masters and project managers and provides practical tools for intelligent project control. The book is also valuable for business people who want insight into the type of problems that delivery teams face, and for programmers and other delivery team members who want to gain an understanding of the project manager's day-to-day challenges. While life experience as a member of a software project is useful for quickly recognizing some of the situational nuances, this book will appeal to curious people who are early in their careers.

The described forecasting method relies on the ability of a technical team to deliver working software with consistency. Many of the technical practices enabling such delivery have been perfected and popularized by what came to be known as agile software development. Thus, in practice, the forecasting and control methods align well with the operational mode of many teams delivering software with an agile development methodology. However, this is not a book about agile software development. If you and your team are practicing solid software development and do not label yourself as agile or extreme, then this book will be valuable for you too.

Estimation by itself is of limited use. We do it to forecast and plan. Forecasting and planning are in turn done to improve the chances of making good decisions and taking appropriate actions. A forecast does not improve a project's performance. It is only a tool for visualizing that performance. This book shows how to use an intelligent forecast for making timely decisions and applying measured project control for steering toward a valuable goal.

Before going any deeper in this book, we need to establish that estimation and forecasting are two different things. This book is more about forecasting and control. However, the term "estimation" has been misused for so long that it has become the normal term for describing what amounts to forecasting. Estimation is only the initial guess about the size of something. Forecasting is the activity of processing input data, including the estimates, and formulating an intelligent prediction.

Where this book ventures into a longer treatment on a subject, or establishes an explicit view on the meaning of a concept, it is not done with the intent to educate the reader, rather it is done with the intent of providing context so the reader can align their understanding with the ideas in the book. For more information on some of the relevant concepts, you can refer to the literature provided in the Bibliography at the end.

The method presented here scales well over software projects with different sizes, provided the organizational structure and the development methods are such that continuously delivering working software is achieved.

Small projects consisting of a few developers working for a couple of months are harder to control through the approach provided by this book, but you can still find useful ideas to guide your future discussions with clients and teammates.

In the context of a legacy system, the applicability of the approach depends on whether the system is healthy or not, or at least on how homogeneously unpredictable is working on the system. For projects within healthy legacy systems, you can apply the concepts almost directly. For projects within legacy systems with compromised codebase integrity, you can cherry-pick ideas that make sense in your context and apply them when there is an opportunity.

This book will be of particular value to people who provide software project delivery as a service to other companies or departments within larger organizations. Teams utilizing modern development techniques and automation have often achieved the technical excellence required for reliable forecasting and intelligent project control. These teams often serve big corporate clients who are not well dispositioned or are downright allergic to the concept of no estimation and no long-term planning. By providing a viable alternative to the Gantt chart as a control method, this book will help delivery teams speak the language of business people without sacrificing software development performance or quality of delivery practices.

Finally, *Software Project Estimation* will provide clues for achieving a mind-set that allows us to bridge an important gap—to see each other (delivery teams and clients) as human and to appreciate each other's and our own needs, abilities, and expectations.

#NoEstimates

Since this book has the word "estimation" in its title, it seems appropriate to address the idea of #NoEstimates.

#NoEstimates is about searching for alternatives to estimates. The method described in this book might actually fall in this category of alternative to the broadly accepted methods of estimation. However, it can be also seen as a logical extension of traditional estimation. I will leave it to the reader to make their own judgment as to whether the method described here is alternative or not—if that categorization matters to the reader at all.

Exploring alternatives is interesting when context is maintained. Still, much of the discussions in the #NoEstimates blogs and articles move haphazardly from project delivery to product development to software development. These are vastly different contexts. The book you are reading is about estimating, forecasting, and controlling software projects.

In contrast, the concepts of business value and guided product experiments are primarily about product development. Software quality, continual integration, and automation are primarily about software engineering. All three—product delivery, software delivery, and project delivery—are complementary, and when done right each draws on the strengths of the others. While product delivery and software delivery present just as interesting and important problems, they are not the main subject of this book and are only examined where it facilitates the discussion of intelligent project forecasting.

How Is This Book Organized?

The concepts in the book build upon each other. Concepts covered earlier are needed to fully appreciate those covered later. Ideally the reader will take on a sequential approach and read the book from front to back.

That said, many of the sections in the book can be read without having read the previous sections or chapters. Some readers might be familiar with a few of the ideas presented in this book and could dive straight into a later chapter. The sections within chapters are kept brief, making it easier to locate topics when reading nonsequentially.

Chapter 0 initializes a minimalistic context for the rest of the book.

Chapters 1–3 lay out the topography.

Chapter 4 describes the core estimation and forecasting techniques.

Chapters 5 and 6 add important details.

Appendix A is a compilation of ideas and techniques that are important for supporting the practical application of this method.

Appendix B contains a few spreadsheet examples.

Frequently Used Pronouns, Collective Nouns, and Terms

A few words appear repeatedly throughout the text:

"People" is often used as all the people on a project, which includes both client and the software delivery team. Or, "people" can sometimes refer to only one of these groups—either the client team or the software delivery team.

"We" is mainly from the perspective of the software delivery team, but it is also used from the perspective of all the people involved on a software project. Occasionally "we" stands for project managers and scrum masters only.

"Business people" and "client" are used for the people who request the software solution. Sometimes these people are external clients, and sometimes they are from another department within the same company.

"Software developers" is generally meant as the programmers, designers, and testers from the delivery team, but is sometimes used to describe all the people who contribute to the delivery of a software project, including the scrum master and project manager.

"The team" and "project team"—these and similar expressions are used to mean all the people on a project, including people from the client team who are actively engaged with the project.

"Delivery team" is basically "the project team" less the "business people."

If there are five scrum teams working on the same project, then all of these are considered "the delivery team," since the major discussion in this book focuses on estimating and forecasting complete projects and not individual team's performance.

"Accurate" is generally used to mean both accurate and precise, since we are only interested in the pragmatic side of forecasting.

"Project" is a collaborative enterprise for achieving a particular valuable goal over a set period (and within certain limitations).

“Project management” is meant as the activities related to decision making and application of control over project parameters in a way that is most conducive to the project’s success. It also includes the activities related to securing the people’s well-being within the project boundaries.

“Project manager” is generally meant to describe a role and not a job title. Anyone who is participating in project management activities on the project can play the project manager’s role. A scrum master can often be a project manager, but so can a team lead, a director, or a manager. Occasionally, the whole team can be the project manager.

“Project performance” captures the project’s progress toward success in terms of delivered functionality and expanded effort. It is not exactly the same as team performance, although both are certainly related. It is important to keep in mind that the definition of success might shift during a typical project as people adjust to reality.