

PREDICTING THE UNPREDICTABLE

PRAGMATIC APPROACHES TO ESTIMATING PROJECT SCHEDULE OR COST



AUTHOR OF "MANAGE IT!"
YOUR GUIDE TO MODERN, PRAGMATIC PROJECT MANAGEMENT"

JOHANNA ROTHMAN

Predicting the Unpredictable

Pragmatic Approaches to
Estimating Project Schedule or Cost

Johanna Rothman

This version was published on 2015-05-26

ISBN 978-1-943487-03-5



Practical ink

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the author.

Every precaution was taken in the preparation of this book. However, the author and publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information contained in this book.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Practical Ink was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals.

©2015 Johanna Rothman

*To everyone who was ever asked,
“How much will this project cost?” or
“How long will this project take?”*

Contents

- Acknowledgements i
- 1. Introduction 1
 - 1.1 Estimates Are Guesses or Predictions 1
 - 1.2 Estimates Change 2
 - 1.3 Estimates Expire 3
- 2. What Estimates Are 5
 - 2.1 Provide an Accurate but Not Precise Estimate 6
- 3. Why Do We Estimate Anyway? 8
 - 3.1 Why Do You Estimate? 9
 - 3.2 Ask This Question Before You Estimate 11
- 4. Software is Learning, Not Construction 12
 - 4.1 Inch-Pebbles or Small Stories Show Progress 14
 - 4.2 Learn With Spikes 14
- 5. Think About Estimation 15
 - 5.1 Estimating the Unknown: Dates or Budgets 16
 - 5.2 Determine Your Degrees of Freedom 17
 - 5.3 Insist on a Ranked Backlog 19
 - 5.4 The Team Doing the Work Provides the Estimate 20
- 6. How to Estimate 22

CONTENTS

6.1	Your First Best Bet: Make Your Stories and Chunks Small	22
6.2	Your Second Best Bet: SWAG and Refine . . .	23
6.3	Collect Data	24
6.4	When You Have a Decreed Date	26
6.5	Wrap Up	29
6.6	Estimating a Program	32
6.7	Beware of These Program Estimation Traps . .	35
7.	Rolling Wave Planning	38
7.1	Learn as the Project Proceeds	38
8.	There Is No Correct Estimation Model	40
8.1	We Invent; We Don't Repeat	42
9.	But I Need to Know When the Project Will Be Done	44
9.1	What You Can Say to Managers Who Think They Need to Know	45
10.	NoEstimate or Do Your Estimates Provide Value?	47
11.	Use All Four Parts of Project Estimation	49
11.1	Part 1: Create an Initial Estimate	49
11.2	Part 2: Track Estimation Quality Factor to Understand the Project Estimate	51
11.3	Part 3: Use EQF to Manage Project Concerns .	52
11.4	Part 4: Update Your Estimate as You Know More	52
12.	Show Your Status and Update Your Estimate . . .	54
12.1	Probabilistic Scheduling	54
13.	Refocusing: 90% Done Is Not Almost Done	57
14.	Future Fixes	60

CONTENTS

14.1 Plan Ahead	60
14.2 Hindsight is 20-20	61
14.3 Track Your Estimates	62
14.4 Measure Bad Fixes	63
14.5 Manage Your Buffers	65
14.6 Incorporate Agility	66
 Troubleshooting Your Estimation Problems	68
15.Avoid Multitasking	69
16.Avoid Student Syndrome	71
17.Estimation Units Predict Schedule Slippage	73
18.Edit Those Epics	75
19.What You Can Do For Estimation	79
20.Estimation Depends On...	82
21.Estimating Testing Time	85
22.Need to Learn More About the Work You're Doing? Spike It!	90
22.1 How Does a Spike Work?	90
22.2 Spikes Are About Learning	92
22.3 What Happens to the Code at the End of the Spike?	92
22.4 "Use the Code As Is..."	93
22.5 How Many People Were Involved in the Learn- ing?	94
23.Use Targets as Estimates	95
23.1 How to Use Targets	95

CONTENTS

23.2 When the Target Is a Trap 96

24. How to Avoid Three Big Estimation Traps 97

25. Understanding Multitasking and the Cost of De-
lay on Estimation 100

26. What You Know About Estimation Now 106

26.1 Transition to an Agile Approach or an Incre-
mental Approach for Your Projects 106

26.2 Make Your Features Small 107

26.3 Iterate on your Estimate 107

26.4 Don't Multitask 108

26.5 Don't Let Defects Dictate Your Estimate 108

26.6 Final Thoughts 108

Glossary 109

References 112

More from Johanna 113

Acknowledgements

I thank my [Managing Product Development](#) blog readers. You asked me enough questions that I had to write the answers.

I thank my editors, Rebecca Airmet and Nancy Groth. I thank Jean Jesensky for her indexing for the print book. I thank Karen Billipp for her layout for the print book.

Any errors are mine.

Troubleshooting Your Estimation Problems

You know your estimates are guesses. They will be wrong. Here are some ideas for actions you can take to prevent estimation problems or when you realize your estimates are wrong.

15. Avoid Multitasking

Let's talk about the effects of multitasking on estimation.

Many managers want people to work on several projects at one time. This is the multitasking problem. I've said before that multitasking introduces defects. We have research that says it makes you stupid. Still, some managers don't want to manage the project portfolio and assign people or teams to projects. They think people can work on multiple projects at one time. We still have multitasking.

What can you do?

In *Manage It! Your Guide to Modern, Pragmatic Project Management*, (ROTPM), I said you shouldn't even bother estimating when people are multitasking. Why?

- You have no idea when people will be available to work. A person might estimate a task will take four hours. But, if they need to spread those four hours over five days, you have no idea when that person can deliver the results you want.
- Your project will incur a delay. The more multitasking, the more delay you will have.
- You can't tell how much context switching people have when they multitask. That four-hour task might take eight, because it takes so much time to return to the context.

When you have people multitasking, you cannot estimate with any accuracy.

If you are using estimates to decide which projects to do, you can see that your estimate is wrong, and does not provide you useful information.

Worse, if you incur a cost of delay because your team members are multitasking, you won't realize the value of the project when you work on it and finally release it.

Multitasking is not just the quickest way to waste a team member's time, it makes it impossible to estimate with any accuracy. Even if your ideal time is accurate, the duration cannot be accurate.

If you want an accurate estimate, make sure the project team members are full time on one project at a time.

16. Avoid Student Syndrome

Student Syndrome occurs when the person with the task waits until the last possible moment to start. Some people spend their entire academic career waiting until the night before a project is due and then starting it, pulling an all-nighter, and getting some (hopefully adequate) grade. Student Syndrome isn't for me, but I know lots of people who do this.

I use these techniques to avoid Student Syndrome:

- Ask each person to develop inch-pebbles so that person (and the project manager, if necessary) can track progress.
- Use Estimation Quality Factor to continuously predict the end of their current task (not just the end of the project).
- Ask “What have you completed today?” Just asking can help jiggle people into starting the work.

These techniques work for me too, not just when I manage other people. Just because I don't wait until the last possible moment doesn't mean I don't procrastinate every so often. (In English, that means I procrastinate too :-)

Student Syndrome isn't the same as being stuck, although if I'm stuck, it can look like I'm procrastinating instead of working on the task. I use a timeout to see if I'm stuck. For any

given task, if I can't make progress on it in about 30 minutes, I ask for help. Thirty minutes may be too short or too long for your tasks, so adjust accordingly.

If you're a project manager or a functional manager, notice if your staff are waiting until the last possible moment to start. If so, try something to help people start earlier. Late-as-possible starts lead to late projects.

You most often see Student Syndrome on non-agile projects. You *can* see Student Syndrome on agile projects where each person takes a backlog item individually—or when people multitask.

When people take backlog items individually, try these ideas:

- Pair or swarm on each backlog item. Reduce the team's WIP.
- Make each item small, so a person can finish it in one day.
- Decide on a team norm about what to do when people get stuck on work. How long can they remain stuck without asking for help?

If someone is multitasking, understand why. Multitasking makes everything late and creates a cost of delay in your project.

17. Estimation Units

Predict Schedule Slippage

I've been teaching a project management workshop, and one of the participants said something brilliant: "If you estimate in days, you'll be off by days. If you estimate in weeks, you'll be off by weeks." If you estimate in months, you will be off by months.

Here's why. The more you can break a big task apart, the more likely you are to remember all the pieces and estimate each piece well. The less you know about a task, the more gotchas you'll encounter, and the longer the task will take. And, the bigger the task, the more likely you are to fall into [Student Syndrome](#).

If you're a PM and you don't understand why your schedule is slipping, look at the general task duration. Got a lot of week-long tasks? Or multiple-week-long tasks? Those tasks are slipping, and you won't know why or by how much until the time is almost done. I bet your project will slip for a duration of several of those multi-week tasks. Replan now, breaking all those tasks into [inch-pebbles](#). Then you'll have a much better idea of what it will really take to finish this project. And maybe, just maybe, you won't have that much of a delay, because delays of weeks are very different than delays of days.

Does small planning take time? Yes. That's why I recommend [rolling wave planning](#) so you don't have to do lots of estimation at one time. Or, if you're agile, you know you don't do a lot anyway.

Remember, you can't estimate epics, large compound features. They are too large. You need to break them apart into small features.