

Extracted from:

Competing with Unicorns

How the World's Best Companies Ship Software
and Work Differently

This PDF file contains pages extracted from *Competing with Unicorns*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2020 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

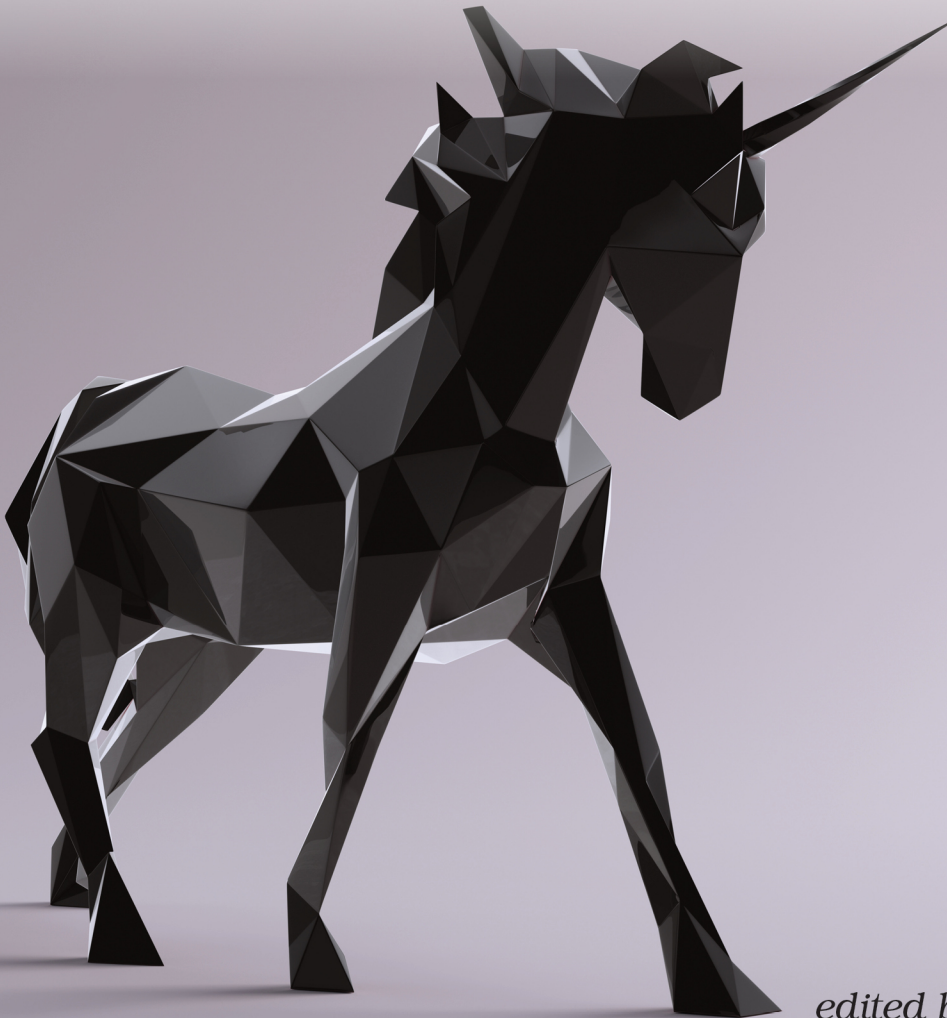
The Pragmatic Bookshelf

Raleigh, North Carolina

The
Pragmatic
Programmers

Competing with Unicorns

How the World's Best Companies
Ship Software and Work Differently



Jonathan
Rasmusson
edited by Michael Swaine

Competing with Unicorns

How the World's Best Companies Ship Software
and Work Differently

Jonathan Rasmusson

The Pragmatic Bookshelf

Raleigh, North Carolina



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 The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at <https://pragprog.com>.

The team that produced this book includes:

Publisher: Andy Hunt

VP of Operations: Janet Furlow

Executive Editor: Dave Rankin

Development Editor: Michael Swaine

Copy Editor: Sakhi MacMillan

Indexing: Potomac Indexing, LLC

Layout: Gilson Graphics

For sales, volume licensing, and support, please contact support@pragprog.com.

For international rights, please contact rights@pragprog.com.

Copyright © 2020 The Pragmatic Programmers, LLC.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

ISBN-13: 978-1-68050-723-2

Encoded using the finest acid-free high-entropy binary digits.

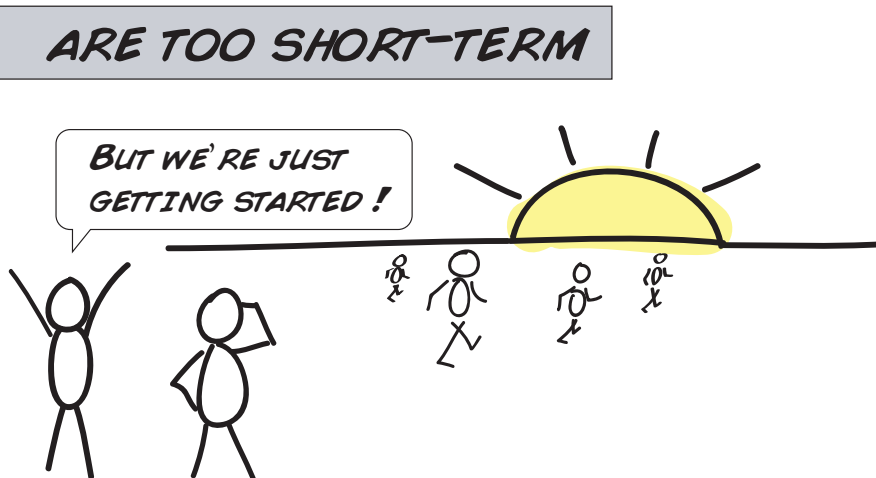
Book version: P1.0—March 2020

Tech companies don't do projects. They do missions instead. And in this chapter you're going to learn why. Learning how to define work in terms of missions will not only give your teams purpose and encourage them to find their own answers. It will put the responsibility and accountability for fulfilling the mission where it belongs—with them.

By the end of this chapter you'll know what missions are, why they're a better vehicle for building product, and how they enable tech companies to move so fast.

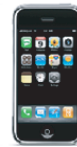
The Problem with Projects

Projects are great when you need to plan out a year in advance what next year's work is going to look like, but they're a lousy way for building anything new. For one thing, they are too short-term.



Projects, by their very definition, have a beginning and an end. And when the project is over—that's it. Everyone packs up and goes home. Product development, however, doesn't work like that. In product development, the first version of your product isn't the end. It's the beginning—which leads to the next thing projects aren't particularly good at. Iterating.

NO OPPORTUNITY FOR FEEDBACK

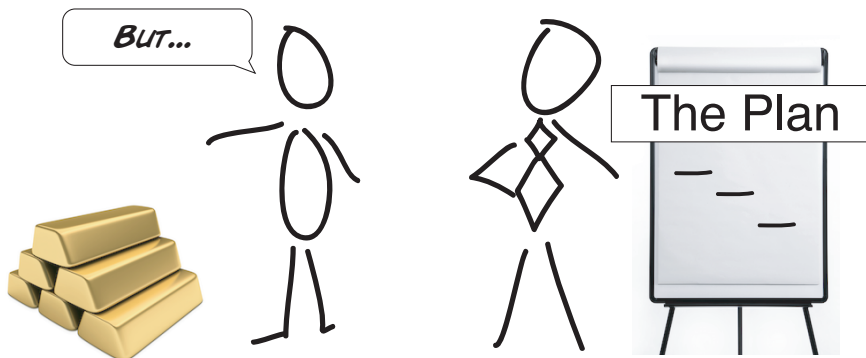


Projects

Product

Product development is a highly iterative process. You build one version. Ship it. Get feedback. And repeat twenty-seven times. Projects aren't built for that. With projects you ship the first version. Declare victory and call it a day. They aren't built for acting on feedback or applying learnings.

ARE TOO RIGID



Projects are also too rigid. When given a project, the goalposts have been set, the path forward is clear, and you don't have a lot of room to deviate or take advantage of new insight, regardless of what you discover along the way. If it's not part of the plan, we don't do it.

But even more frustrating is how disempowering projects are. They encourage you not to think.

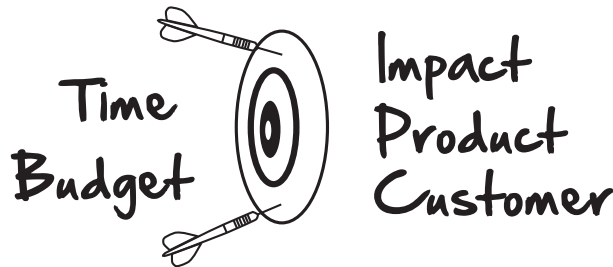
ARE DISEMPOWERING



Not being able to follow through on your instincts and not being able to incorporate your learnings in the name of time and budget make teams not think or care—which is exactly the opposite of what we need when building product.

And that's really what it comes down to—projects just focus on the wrong things.

FOCUS ON THE WRONG THINGS



Projects vs Product

At a tech company, no one cares if your project is +/-10%. That question doesn't even make sense. What does make sense is proof, and that we are building something our customers want, and that we are heading in the right direction toward something valuable.

For these reasons tech companies don't use projects. They simply don't work. What they use is something else. Something that:

- Gives teams an incentive to think longer-term
- Gives teams the time and space to explore
- Lets them incorporate learnings as they go
- Puts the focus back on the work
- Values impact over conformance to plan

Enter the Mission

Missions are high-level goals, given to teams, to direct them in their work toward fulfilling the company's greater purpose.



Google, for example, has a North Star goal of organizing the world's information. That's their purpose. Missions in aid of that would be making search fast, making search accurate, and making all the world's information easy to find.

Tech companies use missions instead of projects to define work. For example, at Spotify we had missions like these:

- Make discovering new music easy
- Win the living room
- Own the morning commute

But missions don't just have to be about product. They can be about non-product things as well:

- Enable other engineering teams to go faster
- Make setting up cloud storage easy
- Prepare the company for the upcoming IPO

In other industries, missions could be something like these:

Industry	Example Missions
Cable company	Win connectivity in the home
Banks	Own casual payments
Financial technology	Detect fraud instantly

The advantages of defining work in terms of missions, instead of projects, are numerous.

Missions Engage the Team

Instead of using people for their backs, missions engage people for their brains. As we will see in the next chapter on squads, [Chapter 3, Empower Through Squads, on page ?](#), the people doing the mission are the ones creating the work. This engages the team, gets them thinking about the solution, and lets them drive the work.

Missions Give People a Sense of Purpose

Purpose is huge. When you come into work knowing that what you do today is going to make someone's life easier, that puts wind in your sails and makes work meaningful. It makes you want to come in. Do your best work. And look forward to doing it all over again tomorrow and the next day after that.

Missions Better Align Incentives

Because teams know they are in it for the long haul, and they are the ones who are going to be iterating and maintaining what they build, they put more care and attention into their work. Which is one of the big reasons why tech company product is of higher quality. It's because the people building the systems are the ones who are maintaining it. That's big.

Not only does it result in better product, it leverages the collective experience and wisdom people develop over time. Instead of disbanding the team at the end of a project, teams on missions stay together.

Finally, missions put the focus back on what really matters. The work. Not the budget and the schedule, which in and of themselves offer no value.

You can sum up the differences between projects and missions like this:

Projects	Missions
Have a budget	The team is the budget
Have an end	Go on indefinitely
Short term	Long term
Have project managers	Have no project managers
Hand off work	Maintain what they build
Disband upon completion	Stay together
Focus on the plan	Focus on the customer
Values meeting expectations	Values impact
Driven from the top down	Driven from the bottom up