

Extracted from:

Create Your Successful Agile Project

Collaborate, Measure, Estimate, Deliver

This PDF file contains pages extracted from *Create Your Successful Agile Project*, 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 © 2017 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

Create Your Successful Agile Project

Collaborate, Measure,
Estimate, Deliver



Johanna Rothman
edited by Katharine Dvorak

Create Your Successful Agile Project

Collaborate, Measure, Estimate, Deliver

Johanna Rothman

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

Development Editor: Katharine Dvorak

Indexing: Potomac Indexing, LLC

Copy Editor: Candace Cunningham

Layout: Gilson Graphics

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

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

Copyright © 2017 Johanna Rothman.

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.

Printed in the United States of America.

ISBN-13: 978-1-68050-260-2

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

Book version: P1.0—October 2017

To Mark, as always.

Every project needs some sort of project management. Not every project, especially agile projects, needs a project manager. In fact, if you have a cross-functional feature team and everyone in the organization has the agile mindset, you might not need any project leadership. However, I have yet to see this circumstance in organizations that are new to agile approaches.

When you first start using an agile approach, you may need a servant leader in the form of an agile project manager or a coach to help the team understand the process it needs. That servant leader might need to help the team recognize what to change about the team's process and how the team could change. You might need that person to protect the team from what I call "management mayhem" in [Avoid Management Mayhem, on page ?](#). Agile project managers facilitate the team's process and remove impediments that prevent the team from delivering finished value.

You might be accustomed to more of a hierarchy in your project: the project manager took responsibility for the team's work and the team's process. Everyone reported to the project manager. Conversely, in agile approaches, team members report to each other. The team commits to and takes responsibility for its work and its process. That changes the kind of leadership a team needs.

Agile teams do not require anyone to control them. Agile teams don't need any hierarchy to work or to solve problems. In fact, hierarchy and control often prevent team members from feeling safe and collaborating, which many teams depend on for learning and experimenting. Agile teams require servant leadership—a different kind of project management.

Agile project-management activities facilitate the team's ability to work and to deliver value. In this chapter, I'll discuss what servant leaders are, the servant leaders you need, and the servant leaders you might consider for your team.

How Leaders Serve the Team

If you are accustomed to traditional project management, you might wonder who tells people what to do and when. The short answer is the team decides what to do as a team. The team takes its work in the way the team prefers. No one assigns work to anyone else.

When one person assigns work to another, that's called "command and control." It's inefficient and often leads to less-than-desired outcomes. Agile teams are in charge of their own deliverables and interactions. The product owner explains the results he or she wants. Then, the team decides how to perform the work.

Agile project managers, coaches, product owners, and even managers are all servant leaders. These people serve the team, not the other way around.

In *The Case for Servant Leadership [Kei08]*, Kent Keith defines seven practices of servant leaders:

1. They are self-aware.
2. They listen.
3. They serve the people who work “for” them. (Keith calls this “changing the pyramid.”)
4. They help other people grow.
5. They coach people, not control them.
6. They unleash the energy and intelligence of others.
7. They work to develop their foresight so they can act, not react.

Servant leaders serve the team by facilitating the team’s work.

I Serve the Team, Not Management

by: James, New Product Owner

I got training as a product owner, but it was nothing like what awaited me at work.

My team didn’t have anyone in a position of leadership except for me. The functional managers thought it was just fine for them to add to any given person’s backlog instead of telling me what they wanted.

I got fed up and called a meeting. I told them that adding more work to anyone’s list of work was no longer acceptable. They would funnel requests through me or the team wouldn’t do any of that work.

One of the managers asked, “Aren’t you a servant leader?” I said I was. The manager then said, “Well, you need to serve us.”

I saw red. I said, “No, I serve the team. You want me to serve you? Act like a team.” I walked out.

After I took a walk around the block, I went back and knocked on my manager’s door. I told him what I’d done. He said, “Good for you!” We discussed who served whom more and I was much happier about the entire situation. Oh, and we got someone to be the agile project manager so I didn’t have to fight those battles.

Servant leaders are not wimps or pushovers. They serve the team, doing what the team needs them to do. Most people don’t need to manage the management team. On the other hand, don’t be afraid to do so. Your team will thank you for your service.

Agile Project Managers Facilitate to Serve

An agile project manager facilitates and serves the entire team. Agile project managers identify and manage risks that the team needs someone to manage.

Here is a list of what an agile project manager might do at the team level:

- Facilitate the team's process. New-to-agile teams don't have the agile mindset and culture baked into their DNA. Who will learn more about how they can make agile work for them? Who will schedule the demo and the retrospective and make sure the two meetings occur?
- Remove or enable removal of impediments the team members can't remove. Many impediments are at the organization level. The team isn't going to tackle them. The team "delegates" the impediment to the project manager.
- Assist the team in measuring the team's velocity, cycle time, and any other measurements. This might be as simple as creating the space for the team to measure as they walk the board.
- Assist the product owner in writing stories for the next iteration. New product owners may not understand how to write small-enough stories for iterations.
- Facilitate the team's workshoping of the project vision.
- Facilitate the team's workshoping of the release criteria.
- Facilitate the team's working agreements, including the team's definition of "done."

In addition, the team may need help in identifying and managing the team's risks. This can be many different things:

- Managing sponsor expectations. Too many senior managers think agile is a way to get "more, faster, cheaper" without realizing the team needs to learn—to learn how the product needs to work and to learn how to work as an agile team.
- Managing the project portfolio so the team has no context switching and is cross-functional and stable.
- Obtaining more funding if necessary.
- Making sure the long-lead-time items show up on time. This is especially helpful for projects with hardware or mechanical parts.

In addition, sometimes it makes sense for an agile project manager to represent the team and its status to more senior management. (See [Chapter 14, Report Your Project State, on page ?](#), for what to report.)

This is not an exhaustive list. You might need other actions from your agile project manager.

If you use Scrum, you might say the agile project manager is a Scrum master. That is correct. I am not fond of “master” or “chief” names for roles. And if you use any other approach to agile, aside from Scrum, you can still use the term “agile project manager” for the role.

Servant Leadership Changed My Expectations of Myself

by: Sherry, New Agile Project Manager

I was accustomed to asking people what they would do and when we would see their deliverables. I didn’t demand that people do anything, but I did ask for dates and status.

When we moved to agile, I had to change everything I did. It took me a while to figure it out.

I started to ask the product owner about the sequencing of the features in the roadmap and the backlog. I made sure he understood the difference between feature sets and features. We wanted features!

I had to educate my management about what they could expect from me and from the team. No more Gantt charts; they could see the roadmap. No velocity charts; we had product backlog burnup charts.

I did have to nudge—okay, push—my management to give us our own UX person. I had to explain about flow efficiency, which blew their minds.

I’m sure I did more. I think it took me about three months to change my expectations of myself. I was no longer the center of the team in terms of my deliverables. Instead, I made it possible for other people to deliver their work.

Project managers who act as servant leaders fulfill useful functions. Project managers can be especially helpful if managers don’t understand the difference between resource efficiency and flow efficiency.

Controlling project managers? Not so helpful.

Here’s what agile project managers do not do:

- The agile project manager does not assign work.
- The agile project manager does not estimate work on behalf of the team.
- The agile project manager does not commit to features, stories, or tasks on behalf of the team.
- The agile project manager does not agree to dates.

- The agile project manager does not agree to constraints on the project.

For many new-to-agile teams, this is a huge change. In serial life cycles, such as waterfall or phase-gate, there is no such role as the product owner. (In serial life cycles, the team works on all the requirements, then all the analysis, then all the design, and so on. Phase-gate life cycles have gates—check-points—after each phase before a team can proceed to the next phase.)

In the serial life cycle, the project manager assesses the requirements and decides what features/requirements/whatever the team should work on first, second, third, and so on. The project manager decides on the deliverables.

In agile approaches, the product owner decides the rank order of what the team will do, often using rolling-wave deliverable-based planning. Rolling-wave planning says to plan a little now—say up to four weeks of work—and as the team completes one week, add a week to the end. The team always has a four-week plan, a rolling wave. See [Manage It! \[Rot07\]](#) for a larger explanation and see [Create Rolling-Wave Roadmaps, on page ?](#), for more details as to how to apply rolling-wave planning to agile projects.

The product owner decides which features (deliverables) the team needs to implement now, and what rank they are. The product owner decides when to replan. The agile project manager might assist/suggest/facilitate, but the deliverable-based planning is the product owner's job.

The product owner now performs some of the work a project manager might have done:

- The product owner (or the product manager) manages the “commitments” to external requests.
- The product owner defines deliverables for the team to focus on and deliver.
- The product owner defines the rolling-wave planning so the team can look ahead a bit.

This can be a large change for traditional project managers, who were accustomed to making their deliverable-based rolling-wave plan work. (Yes, you could make a waterfall project work with deliverable-based rolling-wave planning. It was difficult but possible.) Some project managers have a difficult time reconciling their role to be one of servant leader, facilitating the team's work rather than directing it.

This also means every team needs a product owner. If two teams work off the same backlog, it's possible those two teams can share a product owner. A

product owner has the ability to decide about features in the moment, to define acceptance criteria, and to explain a story for the team at any time.

Beware of the *Trap: Your Team Has a Business Analyst Masquerading, on page ?*. Every team needs a product owner.