

PROJECT LIFECYCLES

*How to Reduce Risks,
Release Successful Products,
and Increase Agility*



JOHANNA ROTHMAN

Author of Create Your Successful Agile Project

Project Lifecycles

How to Reduce Risks, Release
Successful Products, and Increase
Agility

Johanna Rothman

Project Lifecycles

How to Reduce Risks, Release
Successful Products, and Increase
Agility

Johanna Rothman



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.

© Johanna Rothman. This book is available in these formats:
Ebook: 978-1-943487-31-8; Paper: 978-1-943487-32-5; Hardcover:
978-1-943487-33-2

Chapter 2. How Culture Influences Risks and Tradeoffs

Culture influences all aspects of projects, products, and management. That includes the risks each organization can tolerate, and the tradeoffs they might choose.

Each lifecycle differs in the risks they optimize for and against.

2.1. Lifecycles Manage Risks

A lifecycle is an idealized model of how to organize a project's work. That's it. While George Box referred to statistical models, his aphorism is correct for project lifecycles, too:

“All models are wrong, but some models are useful.”

Each lifecycle—as a model—addresses specific risks. Since most projects have multiple risks at different times, teams can choose the elements of each model to decide when to address their project's risks.

There are three major kinds of risks:

- Project-based risks, where the team needs to learn what customers think they want, learn together to create a shared understanding of how the emerging product works, and learn from what they did to plan for the next bit of work.

- Product-based risks, where the organization decides which customers they want to attract and when with the various features and feature sets.
- Portfolio-based risks, where management decides to start, transform, or stop a project. Effective portfolio management supports leadership's more effective corporate strategy.

Lifecycles help the team deliver its work so the product leader and managers can manage their decisions.

Let's start with the typical project risks.

2.2. Assess Your Project's Risks

For many years, we heard about the “iron triangle.” Sometimes, the triangle was “Scope, Quality, Cost.” Sometimes, it was “Scope, Date, Cost.” It was always three things out of a minimum of four possibilities.

In my decades of participating in and managing projects, I've never seen a triangle of risks in practice. Instead, I've seen a pyramid of interconnected risks, as in the following figure.

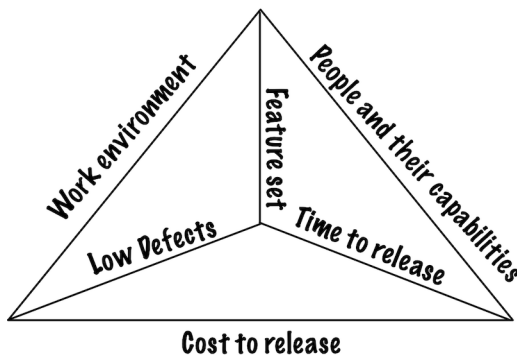


Figure 2. The Project Pyramid

Start on the inside of the pyramid, with the risks the project team can address directly. These risks often feel like pressure on the team. So how much pressure does your team feel to:

- Complete all the features (the requirements)
- With few or zero defects
- By the desired release date?

Then, there are the constraints management often attempts to impose on the project, on the outside of the pyramid:

- Cost to release (the costs to complete the project).
- People assigned to the project and the skills they have.
- Work environment, which includes where people are in the world and the project culture.

In my experience, management can and will change its mind about these supposed constraints, but that requires specific questions.

The more pressure the team feels, the less freedom the team has to manage its work. No lifecycle will work with an overconstrained project. That's why all projects must identify their necessary tradeoffs, so they can decide when they need feedback loops and decisions.

2.3. Project Tradeoffs Clarify Feedback Loops

Since every project has different risks, each project will require different feedback loops. That means each project needs to clarify its feedback loops before the project starts.

Before the project kickoff, I work to clarify the project's drivers, boundaries, and constraints, originally defined in *Manage It! Your*

Guide to Modern, Pragmatic Project Management [ROT07]. I use this scenario:

“Assume you’re three weeks from the desired release date. The team hasn’t finished all the features and won’t in the next three weeks. The testers continue to find problems, so the quality won’t be what we need. What do you want to do?”

I ask managers to rank the importance of each of the six sides of the pyramid. Some sponsors want to claim each side is equally important, but that is untrue. I remind them of the times the sponsors added more people to the project or changed the team’s location, or invested in tooling as late project changes.

Only one side of the pyramid can *drive* the project. One or two more sides might *bound* the project. But the remaining sides are floats, and degrees of freedom that the project can use to accomplish the work.

That’s why I recommend project teams don’t start to work until they have a separate ranking for each side of the pyramid. That information clarifies the necessary feedback loops.

When sponsors have to rank the tradeoffs, they might change their minds about the outside of the pyramid. As a project manager, I want management to see the reality of the project from the start. When we discuss options to manage our reality, we can expose the real constraints and break the fake constraints.

For example, if the sponsor says, “We’re shipping on April 10. Period. No excuses,” then the release date is driving the project. I often choose an incremental approach to make sure the project completes as many features as possible in the time we have.

If the sponsor says, “We *must* have all these whiz-bang new features,” I can be sure we have plenty of technical risks. I’ll use an iterative approach to reduce the technical risks.

When sponsors say, “We must not break anything,” I’ll use an incremental approach with plenty of demos and tests. And to manage

The more innovative the product is (or the team believes it to be), the less the team can plan at any one time. That's because the team will need to iterate more frequently over the various features to obtain feedback and new decisions.

The more a team iterates over features, the more the team can review what they completed. They have the chance to replan future work every time they deliver some increment of value. Innovative products require more customer feedback and more frequent small planning.

The less innovative the product, the more the team can plan at one time. They don't have to worry as much about the changing nature of the product or what the customers expect. It's okay if their feedback loops are longer and there's more time between decisions.

These product risks are more subtle than the tradeoffs between features, defects, and time to release. However, product risks also affect any project's lifecycle choices.

2.5. Product Risks Can Drive Lifecycle Choice

Totally deterministic projects, those on the left of the continuum, can choose a serial approach. Those problems tend to be these kinds of work:

- Defect fixes, where the requirements won't change, or
- Small and short projects that take only one or two people for a maximum of four weeks.

The longer any project continues, the more likely someone will want something new or different. Even small, deterministic projects can morph into something closer to the middle of the continuum if they take more time than anyone expected.

For example, one of my clients completed a project to port an older product to new hardware. Originally, they thought they could use a serial approach. However, after the fifth week, they uncovered previously invisible problems in the old product that they did not want to port to the new product. They learned new information. They had a product squarely in the middle of the continuum.

Now, they needed more feedback loops and decisions. They still didn't have a totally innovative product, but they could not use a serial approach.

All the problems on the right of the continuum require fast feedback loops and frequent decisions to allow the project recursive discovery and delivery. That's where agile approaches shine. However, if your team can't create an agile culture, consider a combination lifecycle, where the team can spend some time iterating over features with sufficiently frequent deliveries.

But all those products in the middle of the innovation continuum? That's where teams can choose how and when to manage the risks. How can the product leader and the team slice and dice the feature sets into useful pieces of value? Which customers need which slices sooner or later? How often do the customers see the work?

That's why the product innovation risks require more subtle lifecycle choices—and probably, more combinations of lifecycles in the same project.

The more innovation your product requires, the more feedback loops and decision points your project needs.

Your Organization's Culture Might Reflect Its Products

Years ago, I worked with an insurance company. By nature—because of the products they sold—they were risk averse. Even

though they continued to lose market share, they had a terrible time using agile ideas because they were so focused on the statistical modeling of risk.

Organizations attract people who want to work on the organization's products. That might reinforce a culture that does not appear to invite agility. However, it's possible to explain the benefits and the risks in ways that will encourage people to reconsider their risk exposure.

Once they realized they could purposefully iterate on prototypes, and then move to incremental deliveries, they did. That allowed them to innovate on their next generation of products.

When product leaders and teams choose which features to expose and when, they can create the necessary project feedback loops and decision points. Those decisions allow the team to gather project and product feedback.

In turn, all that feedback can support how the team organizes the next part of the project. That's how teams can use product risks to help manage project risks.

2.6. Product Feedback Loops Reduce Risks

The most common product risks revolve around identifying the ideal (buying) customer who has problems that this product solves. The next set of risks is the timing: when the team needs to release which features for these ideal customers.

The more often the team experiments and delivers something useful, the faster the product leader can manage those risks.

That means that as the teams iterate over the requirements and

deliver something useful, a product leader needs to be able to change the list of features and the order in which the team delivers those features.

That means the project team needs to deliver increments of value to verify the product solves the customers' problems. As the product leader and the team learn together, the product leader might change the order and type of deliverables as everyone learns what customers need.

An agile culture supports frequent customer identification and feedback.

Finally, organization-based risks, such as project portfolio management can affect a team's lifecycle choice.

2.7. Portfolio-Based Risks Can Drive Lifecycle Choice

Imagine a startup, where they're searching for the right combination of features for the right customers, so they can grow fast. These organizations need the project(s) to deliver something fast, and frequently.

Or, imagine a large organization that realizes its market is totally changing and they need to cancel projects and decide where to place their bets.

Or, imagine an established innovative organization whose current customers have requested a variety of features. After reviewing those requests, the leaders realize all those requests can create a new product that requires several teams to deliver. The leaders decide to create a program, to deliver that one business objective.

The organization's leaders can't take advantage of these opportunities unless each team learns as the team proceeds. That learning allows the team to inform management of the product's