

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 1. The Shameful Secret: Fake Agility is the Norm

The lifecycle wars have been fought, and “agile” has won. Right?

I wish that were true. If “agile” had won these wars, these statements would be true:

- All teams are cross-functional, with all the skills the team needs to discover the actual requirements and deliver a product the customers love.
- No team has to plan for more than a month or so because everyone expects to adapt the plan based on customer feedback of interim deliverables.
- Managers focus on creating an environment that enables and rewards collaborative teams, not the effort of any specific individual.

Instead, every week or two, I read a blog post somewhere about how “agile” is terrible. That’s because many organizations use a two-week iteration as a death march for the team. The teams rightly feel they must deliver a finished product every two weeks. And they don’t have time to think about and make critical decisions, such as for the UI or the architecture.

Worse, because teams are in a death march, they don’t have time to integrate customer feedback into their product development. And don’t get me started on trying to measure individual “productivity” or “velocity.”

None of that is an agile approach. It's jamming traditional, control-based thinking into "agile" ideas—the proverbial square peg into a round hole. No wonder people think "agile" is a bad word. That's agility in name, not in action.

"Agile" is an Adjective

Many people have shortened "The Agile Manifesto for Software Development" to the word "Agile." But notice that the word is an adjective that describes an approach or a technique. Not a noun.

It's even worse when someone says, "Agile/Scrum" because they think Scrum is the only agile approach.

Instead, I will use agility because that word describes a property of nimbleness. Or, I'll use agile as an adjective, to describe a team or a culture. I hope you decide to consider this usage, too.

But teams aren't the only ones who pay for fake agility. So do first-level, and middle managers. And, too often, senior managers do, too.

How is this possible? Because while people in the organization change their practices, no one changes the culture.

Real agility—not fake agility—requires a culture of agility at all levels. Let's start with what an agile team culture might look like.

1.1. Visualize a Successful Agile Team Culture

An agile team culture means the cross-functional team delivers useful value often. Then, the team obtains and uses customer feedback to choose what to do next and how. In addition, the team reflects on how they worked and what they want to change for the next increment of value.

Just as important, the team has the autonomy to work the way they want to. That means no one assigns work to any particular person on the team. The team members decide how to best work, as individuals and as a team.

In addition, the team decides how to organize their work, often with some kind of a board. I've seen many kinds of team boards: on corkboards, on whiteboards, with many columns or just three columns. Even if the team has a facilitative leader, that leader does not tell the team what their board should look like.

The team always gets to choose how they work.

However, an agile team has a product leader, who, with the team's perspective, decides what is most important for the team to do next. That product leader ranks the work, so everyone knows what work is first, second, and third. Even better, the team knows what they don't have to do yet. The team might need a look-ahead to keep their design options open, but the team finishes the immediate work and waits to start the future work.

In a real sense, the team keeps these two questions in mind as they develop the product:

- How do we maintain technical excellence on the current work so we can demonstrate and then receive feedback from customers or internal people?
- How do we make future changes easier?

There is no control function, such as a manager, inside an agile team—unless the team members want that function. Most successful agile teams I see do need someone to facilitate the team’s decisions and to protect the team from other people’s multitasking requests. In addition, if the team is missing some necessary capabilities or skills, sometimes that team needs a cross-organization negotiator to hire or obtain those people from other teams.

(I call that facilitator and negotiator an agile project manager—because that person creates a better agile environment for the team.)

The multitasking requests or insufficient staffing most often occur when the product leaders and management are still discovering how to work in an agile way.

That’s why an agile team culture is not enough. The teams and the organization also need an agile product culture.

1.2. Visualize a Successful Agile Product Culture

In traditional organizations, a product manager, often with the help of business analysts, writes a requirements document at the start of the project. Sometimes, those documents take months to write.

However, an agile product culture defines the product strategy, and the product goal, often within a day. Then the product leader asks this question: What is the smallest chunk of value I want the customers to see as soon as possible?

That means agile product leaders do not require long roadmaps or huge backlogs. They determine what is most valuable right now to the ideal customers.

That’s a huge problem because everything is uncertain, especially at the start of a project. But that uncertainty can help a product

leader make better decisions faster. Here are some questions agile product leaders can answer with short experiments:

- Do I know who the ideal customers are?
- Do I know what problems they need to solve?
- What is the first (and then next) deliverable to help all of us learn if we are solving the right problems for the right customers?

An agile product leader focuses on small deliverables so everyone can learn as fast as possible. Those small deliverables allow a team to finish something valuable, release it, and then assess the results of that value.

That's why agile product leaders have a bias for shorter feedback loops based on small experiments.

However, an agile team and an agile product leader can't create short feedback loops if everyone works separately. That separate working has a name: resource efficiency.

1.3. Resource Efficiency Creates an Anti-Agility Culture

In a more traditional approach, managers assign, review, and sometimes control, the work each person performs. There's a name for that: resource efficiency.

Resource efficiency thinking assumes that product development is closer to factory work than to innovation and learning. If you've ever been part of a team that divided features into tasks, and then assigned specific tasks to specific people, you might have been working in resource efficiency.

The idea behind resource efficiency is seductive: It's possible to divide all the work so the "team" can conquer it. At the end of

all that division, the features (or worse, the entire product) will magically come together. That looks efficient, both to the team and to the managers.

Divide and conquer *can* work for very small, straightforward problems and projects. However, it does not work for projects that take more than a few weeks or require innovation.

When managers think in resource efficiency, they tend to think about predictions vs. reality. They tend to measure schedule and schedule variation; cost and cost variation; and the busy-ness of the people, what they call “resource allocation.”

However, until the team sees a completed feature, all those predictions are just that—predictions. They have no basis in reality.

Worse, when everyone realizes they’re late, the team stops managing their technical excellence. Instead, they take shortcuts to meet the schedule, but they create future problems. Those problems result in late and unplanned feedback loops.

When managers focus on resource efficiency, they create multitasking at all levels: individual, team, and project. In turn, that creates very long feedback loops, high WIP (Work in Progress), and slow delivery of value.

Team members might cooperate, but they rarely collaborate, often because they have individual goals and objectives that might not have anything to do with this project.

That’s why agile managers create a culture of flow efficiency.

1.4. Flow Efficiency Creates an Agile Culture

Flow efficiency thinking allows a team to focus on the flow of work through their system as individuals and as a team. You might have heard the maxim, “Watch the work, not the people.” That’s thinking

in flow efficiency terms. (See *This is Lean: Resolving the Efficiency Paradox* [MOA13] for more details.)

When team members focus on the flow of work, they tend to collaborate more. Instead of handing off work to each other, they work together, finishing a feature. Teams that use flow efficiency tend to finish work and maintain that work's technical excellence.

In addition, managers can focus on the flow of work through the project portfolio by assigning teams to projects, not trying to play Tetris by assigning people to different projects. That allows managers to change their idea of the unit of work from an individual to a team.

When managers focus on flow, they create a culture where agility can thrive.

But if managers remain focused on the individual, as in resource efficiency, agility dies. That's when we see fake agility, where the teams live in agile death marches.

Teams and managers can't just *think* in flow efficiency. Instead, they need to change the measures they use at all levels.

1.4.1. Measures Change in Flow Efficiency

Here's what teams and managers can use for flow efficiency measures:

- WIP, the current number of work items in progress.
- Throughput, the number of work items completed per unit of time.
- Cycle time, the time to release value, at minimum internally, as a trend.
- Aging, how long a piece of work has been in progress.

Teams can use these measures for features while managers can use precisely the same measures for decisions. Once managers start to

measure their decision aging, they realize that they tend to slow the work of everyone else because of their lack of decisions. (See *Why Minimize Management Decision Time* [ROT20] for actual data I measured in organizations.)

No balance sheet uses these measures—but they should. When teams and managers don't minimize WIP, reduce unnecessary waste in their cycle time, and address old work, their throughput goes way down, often to effectively zero.

As you might imagine, there's a relationship between WIP, throughput, and cycle time. That's called Little's Law.

Work in Progress (WIP) = Cycle Time * Throughput

Figure 1. Little's Law

WIP is a function of cycle time multiplied by the throughput. (See *The Kanban Pocket Guide: What No One Has Told You About Kanban Could Kill You* [VSJ22] for more information and all the caveats that go with this equation.)

Here's how I use Little's Law: When a team or a manager takes "too long" to finish work or make a decision, I might use a value stream map to see where the work is stuck. Or, I'll ask questions such as, "Does the team have enough skills and capabilities to finish their work?"

Sometimes, I review the arrival rate of new work. If more work arrives than the team can complete, the WIP increases. I worked with one team that completed three stories every week, but more than three stories arrived every week. Their WIP and cycle time grew, even though their throughput remained the same.

Flow metrics can help everyone—regardless of their position—reason about the actual project and product progress. Everyone can see the flow, or lack thereof, of all the work.

However, these measures don't count in cost accounting.

1.4.2. Cost Accounting Reinforces Resource Efficiency

Cost accounting focuses on the incremental value of building inventory and the variance between the projections and the actuals.

That can work for manufacturing, where the cost of adding value at each step in the manufacturing process can contribute to the overall added value of the product.

Software product development is not manufacturing. Teams create successful software and hardware products by learning and delivering together. While product development uses words from construction, the words don't mean the same thing.

Cost accounting reinforces resource efficiency thinking, where everyone focuses on the individual. That's why many organizations where everyone is overloaded and crazy-busy, but no one can finish anything, that low throughput problem the flow metrics assess.

Every organization needs to use cost accounting to *report* their profit and loss, pay taxes, and all the other external accounting activities. However, the more managers use flow efficiency to *manage* the business, the more agility can thrive.

That means that the measures organizations use to *report* their business state have limited to no use for *managing* the projects.

That's a huge cultural challenge for real agility. Because it feels "inefficient" to maintain two sets of measures. Except, it's actually easy to measure the flow metrics for any project or management effort.

But that's not the only problem. Flow efficiency thinking changes everything: career ladders, performance reviews, and how internal money flows from and to cost centers. (Cost centers and shared

services “teams” take resource efficiency thinking to an extreme—and are anti-agile.)

In my experience, flow efficiency thinking is the most challenging change for an agile culture. And only the managers can change the organizational culture to focus on flow efficiency.

It’s no wonder that agility-in-name, fake agility has won.

However, you can take advantage of agility where you find it.

1.5. Design Your Work to Take Advantage of Agility

With the exception of a strict waterfall lifecycle, you and your team can use any other approach or combination of approaches to design your project’s lifecycle. That design will allow you to take advantage of agility where you can.

You might not be literally “agile,” but you can be more effective. Especially when you can recognize and avoid the fake agile ideas.

1.6. Recognize and Avoid Fake Agility

In general, fake agile cultures overemphasize time to finish and de-emphasize learning. That leads to unplanned and long feedback loops. Worse, this emphasis on time often means the team has no autonomy to experiment with their process or the product.

Here are some examples of fake agility:

- The team creates a backlog for a week or two, but a manager wants a project manager to create a Gantt chart for the entire project—including these next two weeks.
- The team assigns all the stories to individuals in the team planning session.

- The team uses relative estimation instead of cycle time. As a result, all of their predictions are wrong—and they don't know why. Then, since the team doesn't deliver what they estimated, management asks them to do twice as much the next time.

Another example of fake agility is “Scrum-But” or “Water-Scrum-Fall.” In Scrum-But, someone plans for the team, not with the team. Or the team doesn't perform demos or retrospectives on a regular basis.

In Water-Scrum-Fall, someone creates a year's worth of roadmaps or backlog, the team is supposed to use Scrum to deliver, and then a different team releases or deploys.

In both of these cases, the team doesn't learn from feedback and integrate that feedback into their work. This can occur with any agile approach, but because Scrum is the most “used” approach, this occurs most often with Scrum.

Fake agility removes the ease, joy, and flow of work. Worse, fake agility does not manage the customers' needs or the project's risks.

Instead, you can assess all the needs and risks and then choose a lifecycle that will work for you. You can even combine aspects of each lifecycle.

And if you have a culture where you can't call it a “whatever” lifecycle, you can even call it “our agile approach.”

But whatever your approach, you can create some ease and joy in your work. That's the promise of choosing or designing your own lifecycle.

1.7. Remember This About True Agility

True agility is a cultural change. It requires: