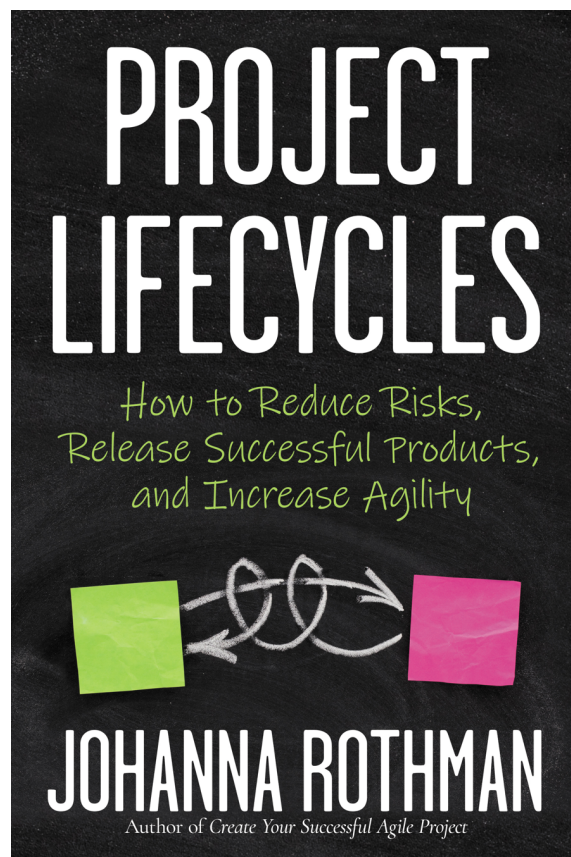


# Project Lifecycles Audiobook Supplemental Materials

## Project Lifecycles: How to Reduce Risks, Release Successful Products, and Increase Agility



## Table of Contents

<b>Chapter 1, Fake Agility .....</b>	<b>3</b>
<b>Chapter 2, How Culture Influences Risks .....</b>	<b>3</b>
<b>Chapter 3, Serial Lifecycle Figures .....</b>	<b>4</b>
<b>Chapter 4, Iterative Lifecycle Figures.....</b>	<b>6</b>
<b>Chapter 5, Incremental Lifecycle Figures .....</b>	<b>8</b>
<b>Chapter 6, Combination Lifecycle Figures .....</b>	<b>10</b>
<b>Chapter 7, Agile Approaches Figures.....</b>	<b>12</b>
<b>Annotated Bibliography.....</b>	<b>14</b>
<b>Glossary .....</b>	<b>16</b>
<b>More from Johanna.....</b>	<b>18</b>
Management Books: .....	18
Product Development:.....	18
Personal Development:.....	18

## Chapter 1, Fake Agility

Figure 1. Little's Law

$$\text{Work in Progress (WIP)} = \text{Cycle Time} * \text{Throughput}$$

## Chapter 2, How Culture Influences Risks

Figure 2. The Project Pyramid

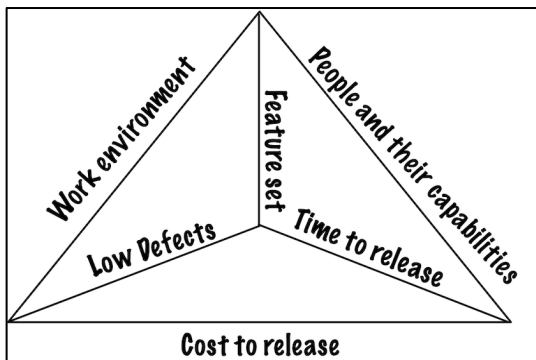
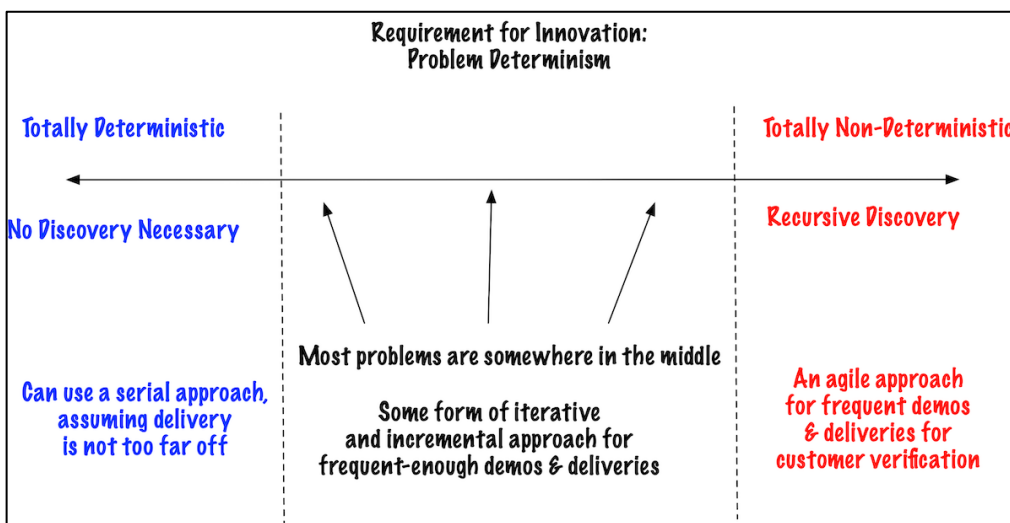


Figure 3. Problem Determinism Continuum



### Chapter 3, Serial Lifecycle Figures

Figure 4. Serial Lifecycle



Figure 5. Requirements Take Too Long

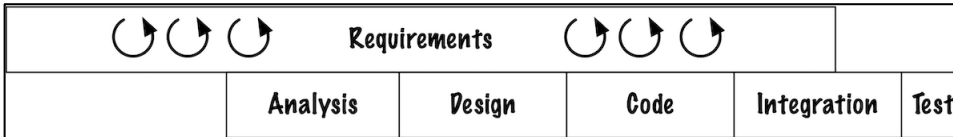


Figure 6. Freezes are Slush

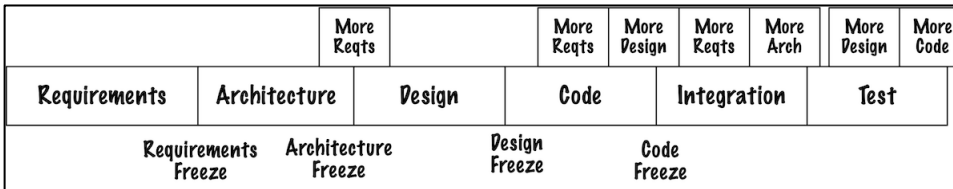


Figure 7. Many Unplanned Feedback Loops

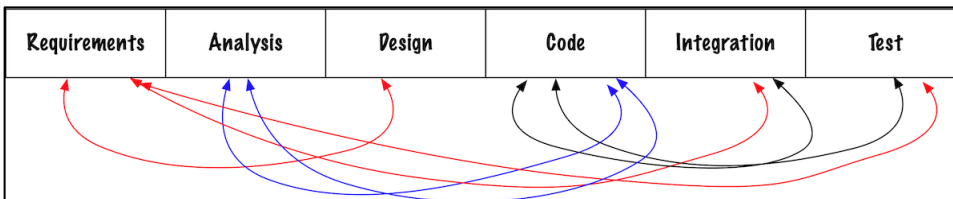
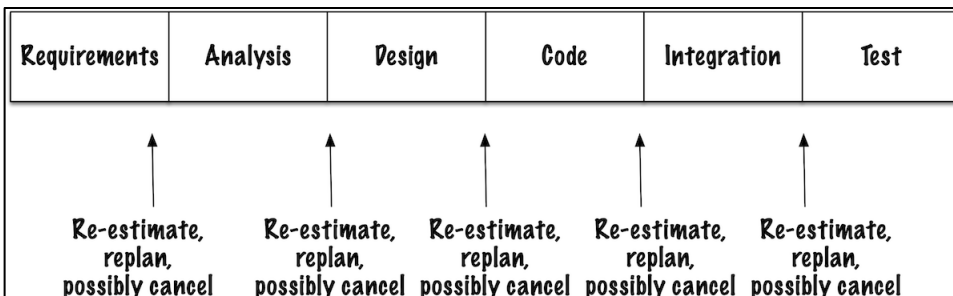
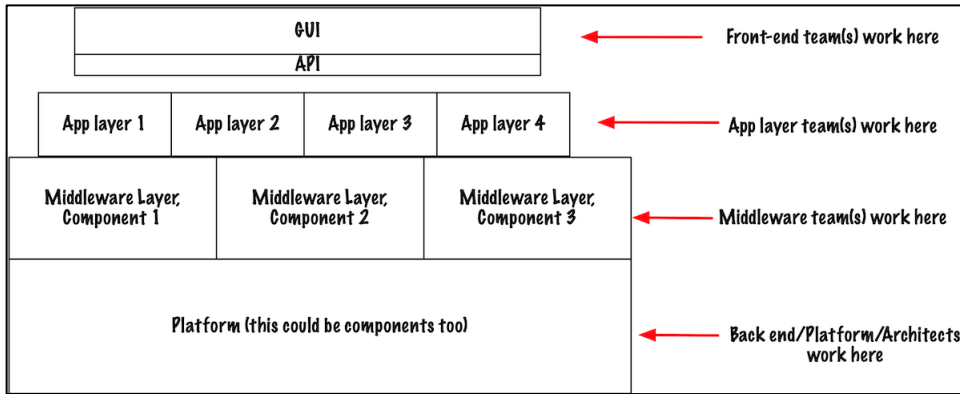


Figure 8. Serial Lifecycle with the Option to Cancel



**Figure 9. Implement Across the Architecture**



## Chapter 4, Iterative Lifecycle Figures

Figure 10. Boehm's Spiral Model

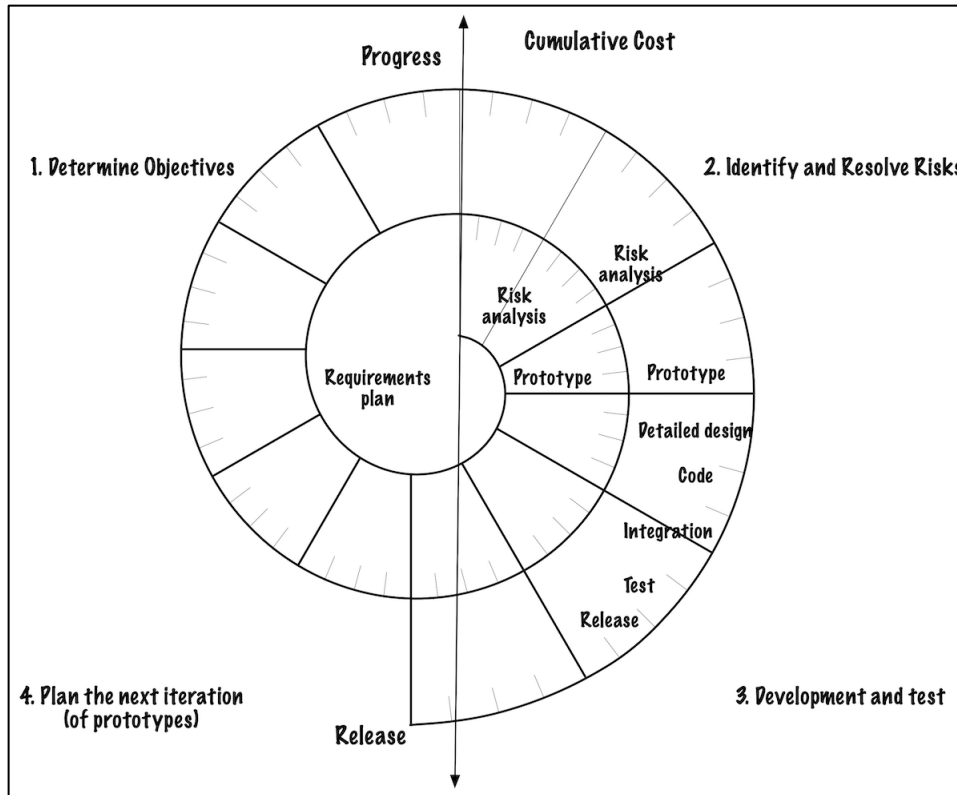
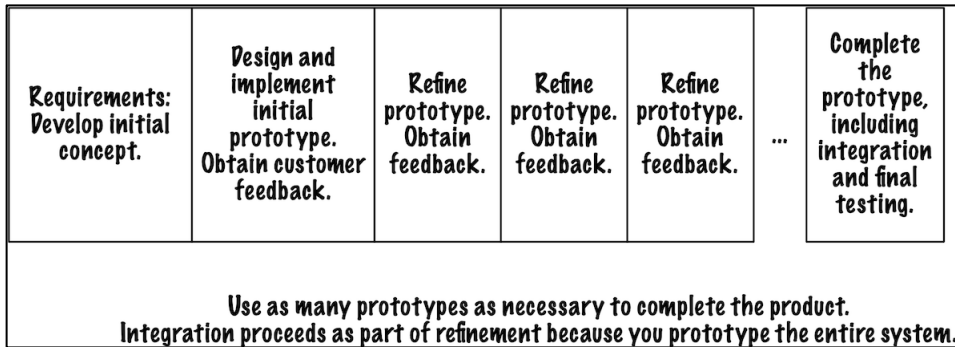


Figure 11. My Experiences with Spiral Lifecycles

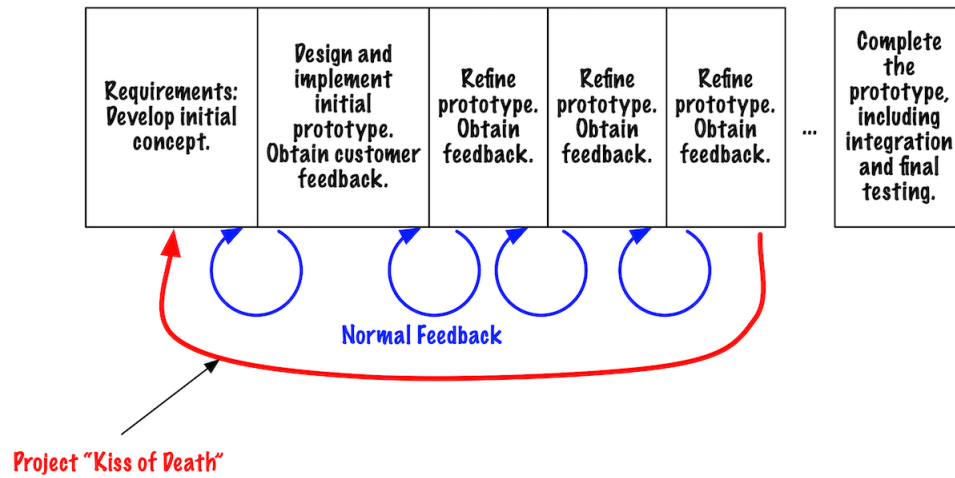
Requirements	Prototype: Analysis, design, code. Check with customer. Replan based on feedback	Prototype: Analysis, design, code. Check with customer. Replan based on feedback	Prototype: Analysis, design, code. Check with customer. Replan based on feedback	...	Finish the prototypes: Complete the development work	Integration	Test
				As many prototypes as necessary			

Each prototype can be a different duration.  
Remaining boxes not to scale with time.

**Figure 12. Evolutionary Prototyping**



**Figure 13. Late Iterative Lifecycle Feedback Loops**



## Chapter 5, Incremental Lifecycle Figures

Figure 14. Design to Schedule

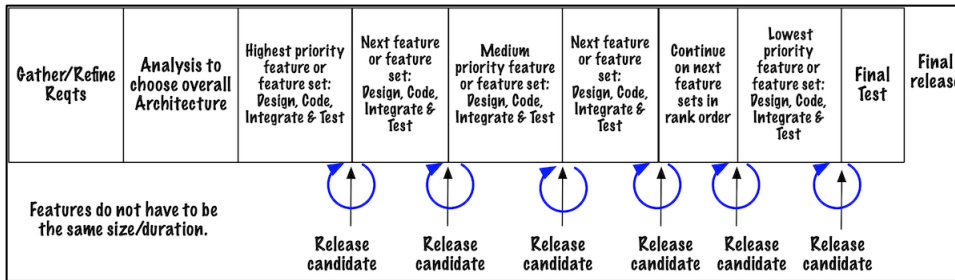


Figure 15. Feedback Loops in Design to Schedule

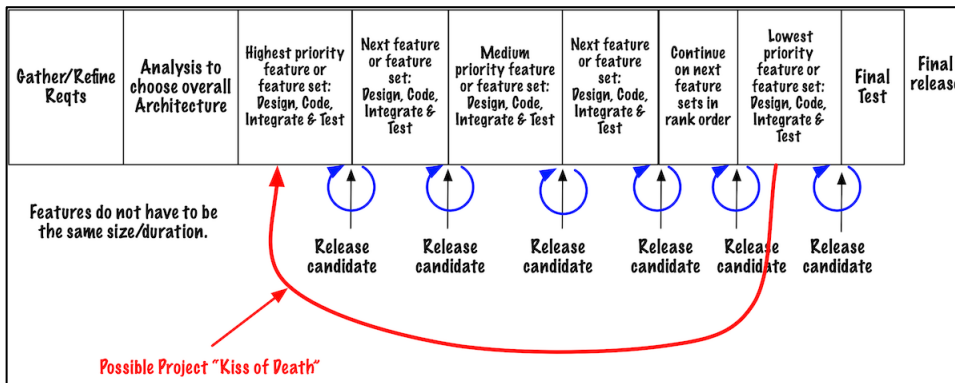


Figure 16. Staged Delivery

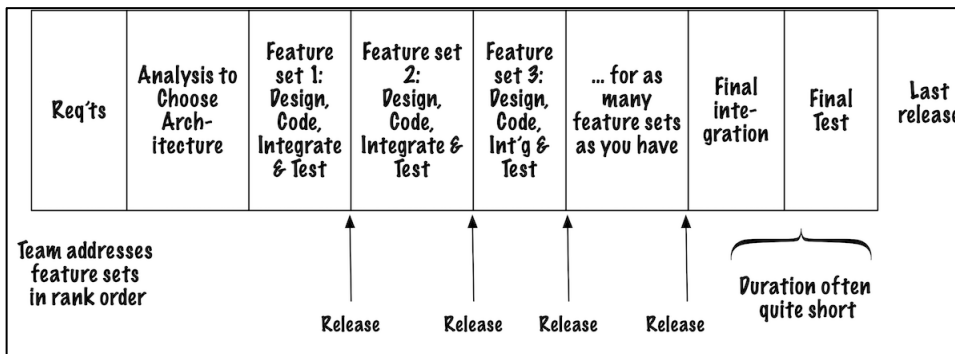


Figure 17. Staged Delivery Feedback Loops

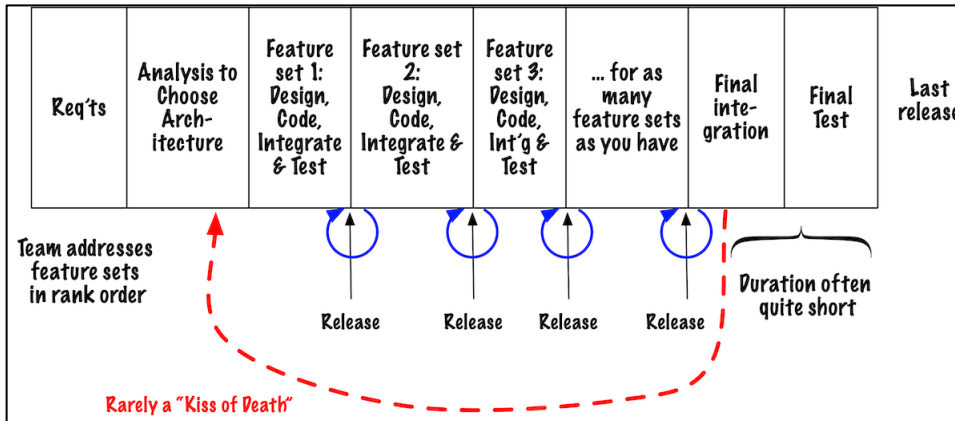
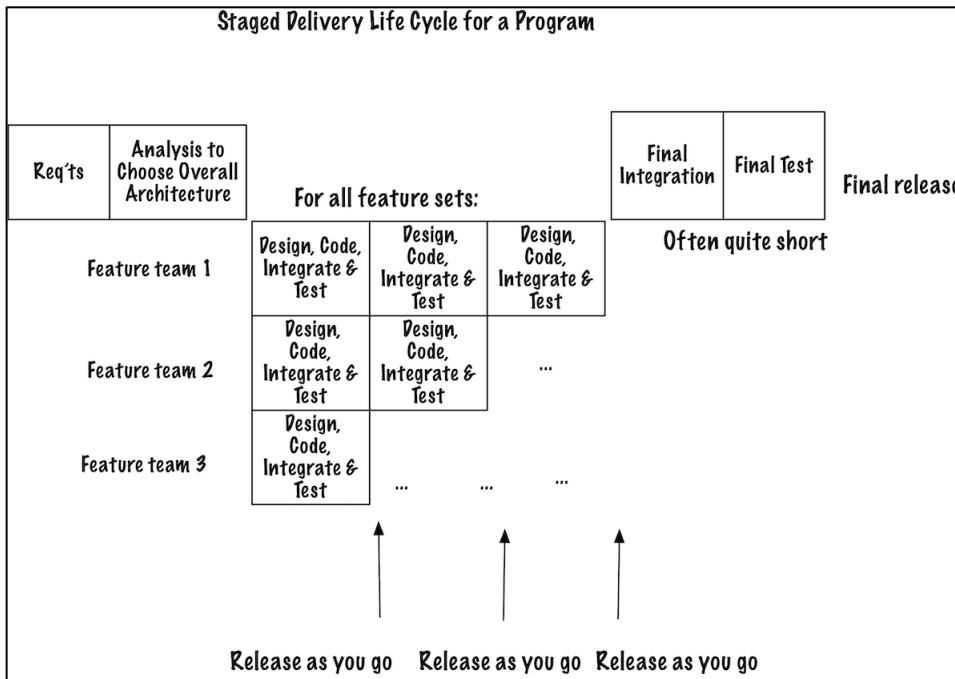


Figure 18. Program-Level Staged Delivery



## Chapter 6, Combination Lifecycle Figures

Figure 19. Date-Driven Combination Lifecycle

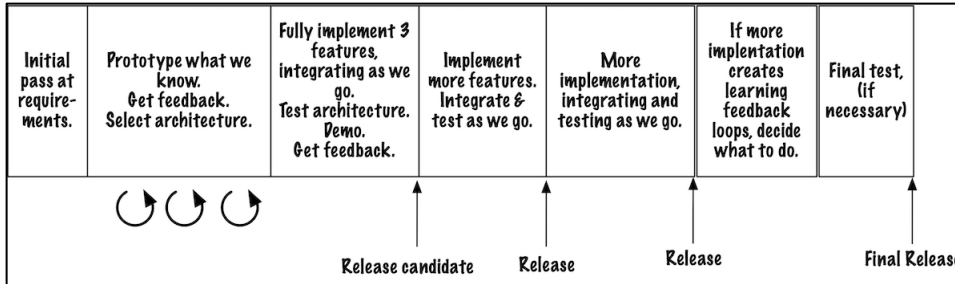
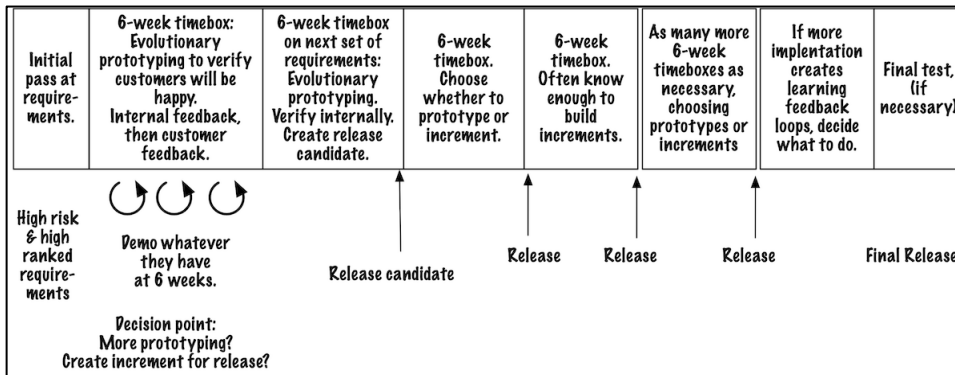
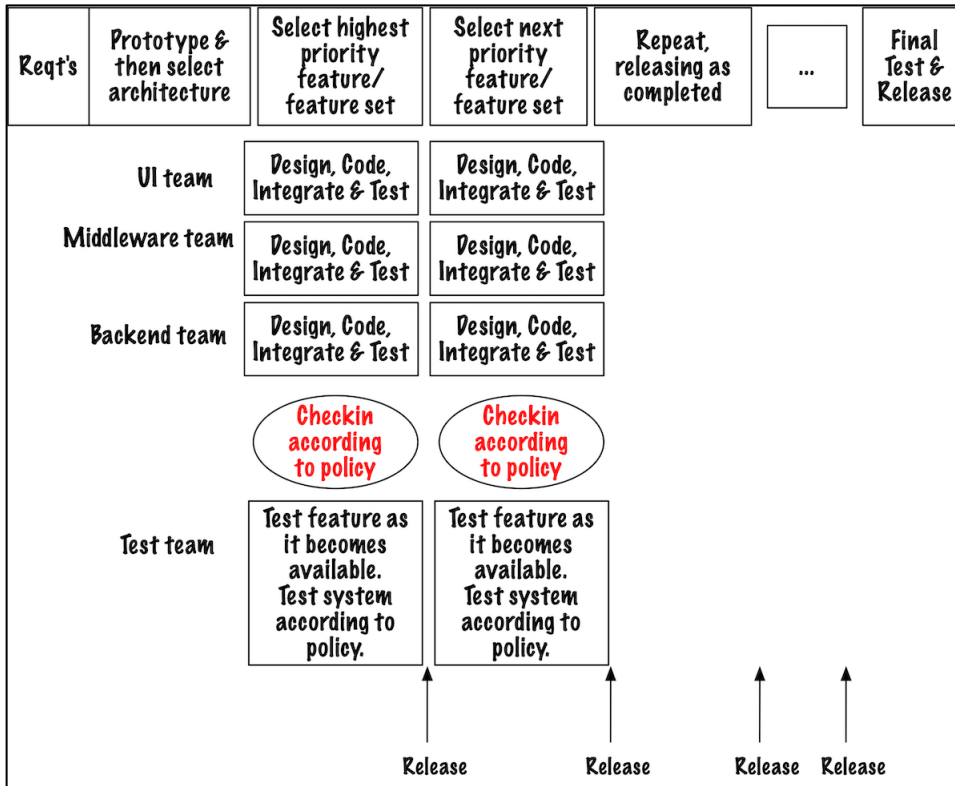


Figure 20. Feature Driven Combination Lifecycle



**Figure 21. Program-Level Combination with Component Teams**



## Chapter 7, Agile Approaches Figures

Figure 22. Three-Column Scrum Board

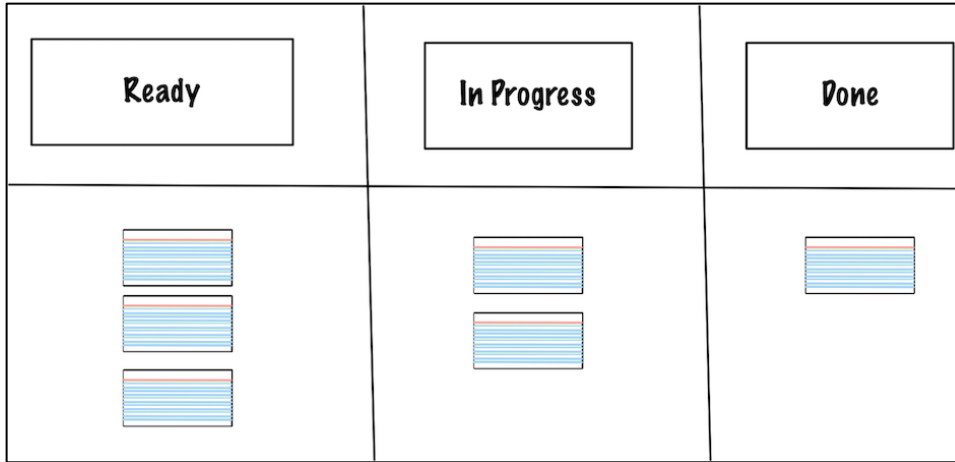


Figure 23. Multiple Column Kanban Board

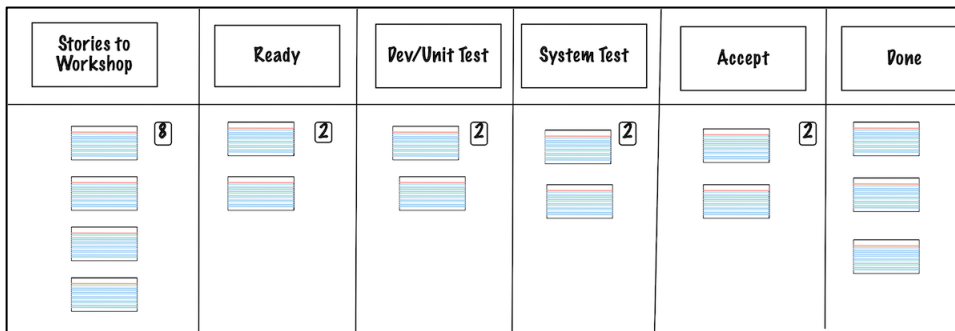
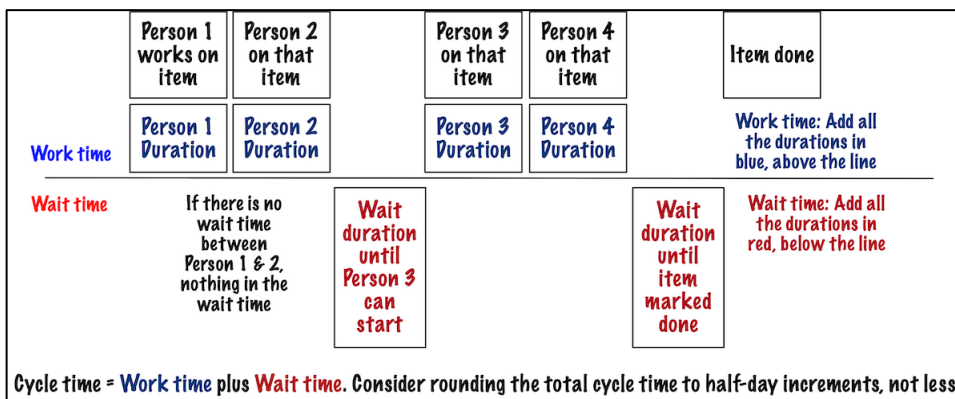
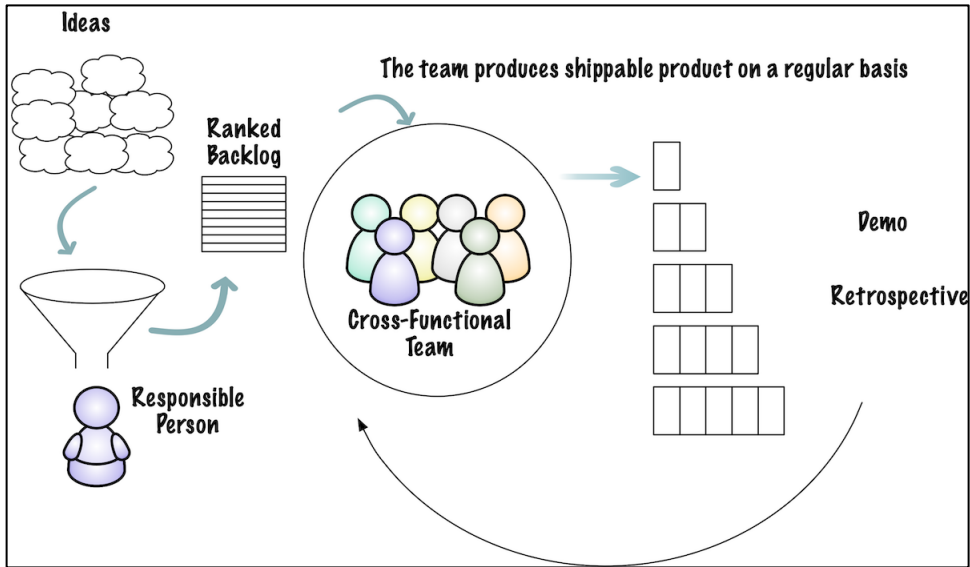


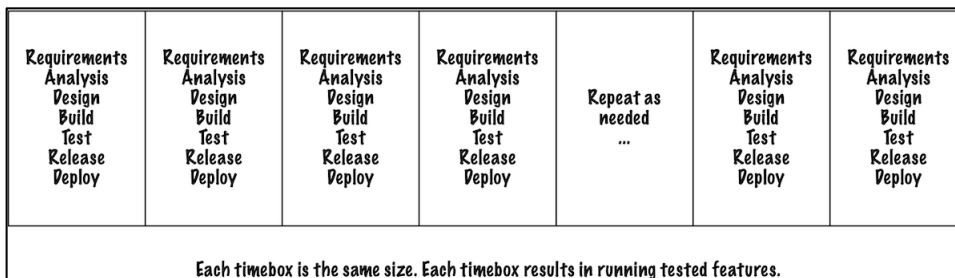
Figure 24. Blank Value Stream Map



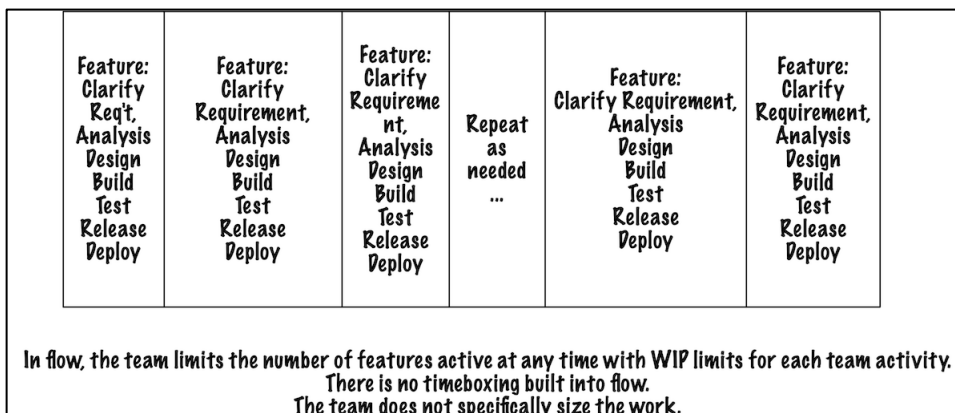
**Figure 25. General Agile Picture**



**Figure 26. Iteration-Based Agile Approach**



**Figure 27. Flow-Based Agile Approach**



## **Annotated Bibliography**

[BAT75] V. Basili and J. Turner, "Iterative Enhancement: A Practical Technique for Software Development," IEEE Trans. Software Eng., Dec. 1975, pp. 390- 396. Available online at <https://www.cs.umd.edu/~basili/publications/journals/J04.pdf>. I was still in college when one of my professors assigned this paper. (I don't remember which professor or which class.) That's when I realized I wasn't stupid for not understanding how the product would unfold—I was normal. That, while some people could conceive of a product's architecture totally in their head, most of us needed to iterate and increment our way through the architecture and design.

[BOE86] Boehm, Barry. *A Spiral Model of Software Development and Enhancement*. Available online at <https://www.cse.msu.edu/~cse435/Homework/HW3/boehm.pdf>. If you read Boehm's paper, he references one of Royce's feedback visuals for the waterfall lifecycle.

[MOA13] Modig, Niklas and Pär Åhlström. *This is Lean: Resolving the Efficiency Paradox*. Rheologica Publishing, 2013. Possibly the best book about how managers should consider agile and lean. A wonderful discussion of resource efficiency vs. flow efficiency.

[MEY93] Meyer, Christopher. *Fast Cycle Time: How to Align Purpose, Strategy, and Structure for Speed*. The Free Press, New York. 1993. I read this in 1997, when a client in a regulated industry said, "What if we just finish one feature at a time? We would have complete traceability, right?" Uh, yes. The first book I read that encapsulated all the ideas behind agility at every level.

[ROT07] Rothman, Johanna. *Manage It! Your Guide to Modern, Pragmatic Project Management*. Pragmatic Bookshelf, Raleigh, NC. 2007. Read this book to become a facilitative project manager, in any lifecycle. One of the reviewers called it, "Agile for the rest of us." Learn to create an environment where the project team can thrive.

[ROT15] Rothman, Johanna. *Predicting the Unpredictable: Pragmatic Approaches to Estimating Project Cost or Schedule*. Practical Ink. 2015. Ways to think about prediction and what might make sense for you and your team. And what to do when your estimate is "wrong."

[ROT16A] Rothman, Johanna. *Agile and Lean Program Management: Scaling Collaboration Across the Organization*. Practical Ink. 2016. Scale collaboration, not process. Everything you need to know for a facilitative agile and lean approach based on principles, not practices.

[ROT16B] Rothman, Johanna. *Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects, 2nd ed.* Pragmatic Bookshelf, Dallas, TX and Raleigh, NC, 2016. Sometimes, program managers encounter project portfolio decisions with the feature set, or the request for people to multitask. This book helps you manage all the work in your project portfolio. I also have more references about why multitasking is crazy in here.

[ROT17] Rothman, Johanna. *Create Your Successful Agile Project: Collaborate, Measure, Estimate, Deliver*. Pragmatic Bookshelf, Dallas, TX and Raleigh, NC, 2017. You don't need to adopt a specific framework for any given agile project. Instead, use the agile and lean principles to adjust for your project's context.

[ROT20] Rothman, Johanna. *Why Minimize Management Decision Time*. Available online at <https://www.jrothman.com/wmmdt>. An explanation and value stream map of the delays I see too often, just for portfolio decisions.

[ROY87] Royce, Winston. MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS. ICSE '87: Proceedings of the 9th international conference on Software Engineering March 1987, Pages 328–338. Available online at <https://dl.acm.org/doi/10.5555/41765.41801>

[SCH10] Schein, Edgar H. *\*Organizational Culture and Leadership\**. Jossey-Bass. San Francisco 2010. Culture is not about the color of the walls or the foosball tables. Culture is about us, as humans. A fascinating look at what culture means.

[[VSJ22] Vacanti, Daniel, Singh Prateek, and Colleen Johnson. *The Kanban Pocket Guide: What No One Has Told You About Kanban Could Kill You*. When teams use flow efficiency to think about how their work flows through the team, they are much more likely to create an agile approach. This short book explains how to use kanban to do so.

## **Glossary**

Definitions for terms I've used in this book.

### **Backlog**

Ranked list of items that need to be completed for the product.

### **Behavior-Driven Development or BDD:**

Specify examples so everyone understands the story under discussion.

### **Flow**

The team takes a limited number of items to complete, and uses the WIP limit instead of a timebox as a way to control how much work the team takes.

### **Iteration**

A specific timebox, often of one to four weeks, so a team can limit the work they consider now.

### **Pairing**

When two people work together on one item.

### **Mob**

The team works all together on one item. Only one team member types (the driver), while one other team member (the navigator) takes all the input from the rest of the team (the mob). Teams often use a short timebox of five to eight minutes before everyone rotates position.

### **Refactor**

Cleaning the code or test while it is in development.

### **Servant Leadership**

An approach to managing and leading where the leader creates an environment in which people can do their best work. The leader doesn't control the work; the team does. The leader trusts the team to provide the desired results.

### **Sprint**

An iteration in Scrum.

### **Swarming**

The team works all together on one item. However, each person works according to their specific skill. The team often synchronizes every hour to see where they are. When one person is done, that person assists anyone else on the team. The one rule of swarming is this: No one starts anything new until this one item finishes.

### **Timebox**

A specific amount of time in which the person or team will attempt to accomplish a specific task.

### **WIP or Work in Progress**

Any work that is not complete.

## **More from Johanna**

People know me as the “Pragmatic Manager.” I help leaders and teams see simple and reasonable alternatives that might work in their context—often with a bit of humor. Equipped with that knowledge, they can decide how to adapt how they work.

See [www.jrothman.com](http://www.jrothman.com) for my blogs and other writing.

If you liked this book, you might also like some of my other books.

### **Management Books:**

- *Practical Ways to Manage Yourself: Modern Management Made Easy, Book 1*
- *Practical Ways to Lead and Serve—Manage—Others: Modern Management Made Easy, Book 2*
- *Practical Ways to Lead an Innovative Organization: Modern Management Made Easy, Book 3*
- *Behind Closed Doors: Secrets of Great Management*
- *Hiring Geeks That Fit*

### **Product Development:**

- *From Chaos to Successful Distributed Agile Teams: Collaborate to Deliver*
- *Create Your Successful Agile Project: Collaborate, Measure, Estimate, Deliver*
- *Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects, 2nd ed*
- *Agile and Lean Program Management: Scaling Collaboration Across the Organization*
- *Diving for Hidden Treasures: Uncovering the Cost of Delay Your Project Portfolio (with Jutta Eckstein)*
- *Predicting the Unpredictable: Pragmatic Approaches to Estimating Project Cost or Schedule*
- *Project Portfolio Tips: Twelve Ideas for Focusing on the Work You Need to Start & Finish*
- *Manage It!: Your Guide to Modern, Pragmatic Project Management*

### **Personal Development:**

- *Successful Independent Consulting: Relationships That Focus on Mutual Benefit*
- *Free Your Inner Nonfiction Writer: Educate, Influence and Entertain Your Readers*
- *Write a Conference Proposal the Conference Wants and Accepts*
- *Manage Your Job Search*

I'd like to stay in touch with you. If you don't already subscribe, please sign up for my email newsletter, the Pragmatic Manager. Once a month, I send content you can use, plus announcements of workshops and more.

I'm on the various social media:

- LinkedIn: <https://www.linkedin.com/in/johannarothman>

- X/Twitter: @johannarothman (I'm no longer active there, but my account is still there.)
- Mastodon: <https://mastodon.sdf.org/@johannarothman>
- YouTube: <https://www.youtube.com/@johannarothman>

Did this book help you? If so, please consider writing a review of it. Reviews help other readers find books. Thanks!

Johanna