

# Adaptive Processes

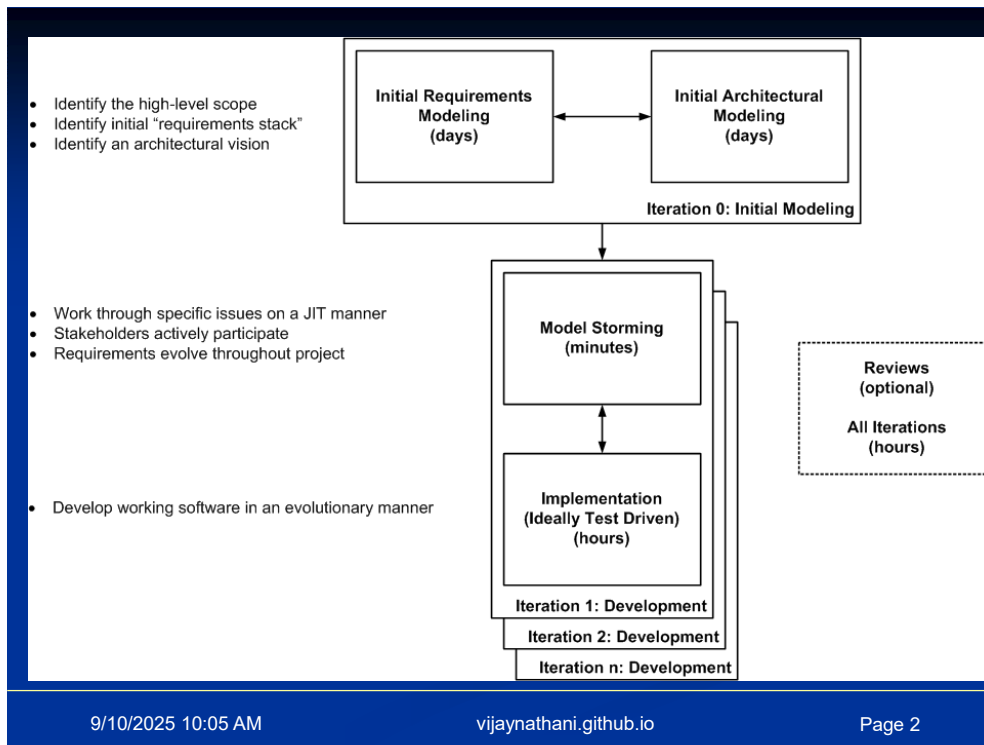
Continuous development, not episodic

9/10/2025

[vijaynathani.github.io](https://vijaynathani.github.io)

1

Ebay has 1000's of servers. It does a production release every two weeks.



## What is Agile?

“Agility is the ability to both create and respond to change in order to profit in a turbulent business environment.”—*Jim Highsmith, Agile Software Development Ecosystems*

Nimbleness and flexibility balanced with structure

=====

Some architectural decisions need to be taken at the start e.g.

1. File or Database
2. Scalability of 100 or 10 million users
3. Database size less than 10GB or thousands of GBs
4. Java or .NET or Ruby

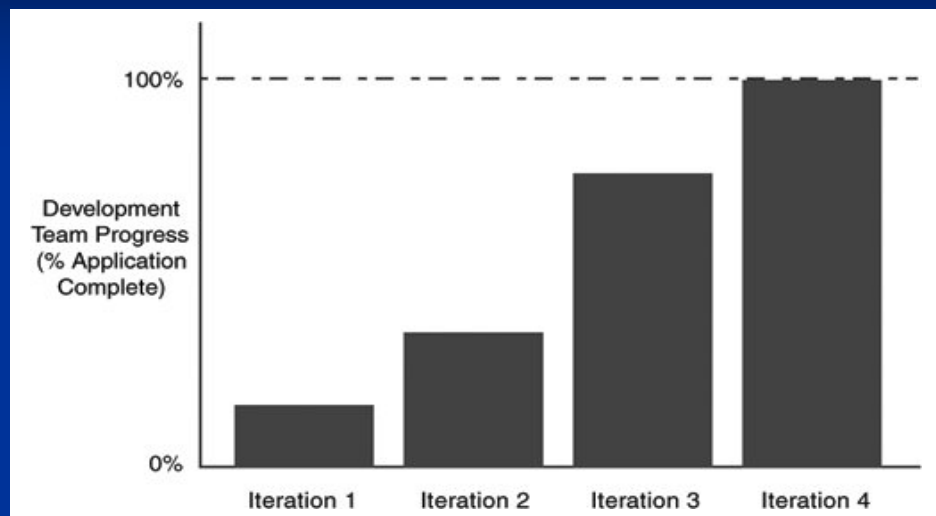
A decision that is reversible need not be taken at the start. An decision that is difficult to change later is an architectural decision.

## Cadence



Do as little as possible with the highest possible value as fast as possible.  
Cadence means Rhythm.

## Progress



9/10/2025 10:05 AM

[vijaynathani.github.io](https://vijaynathani.github.io)

Page 4

Iteration refers to the process

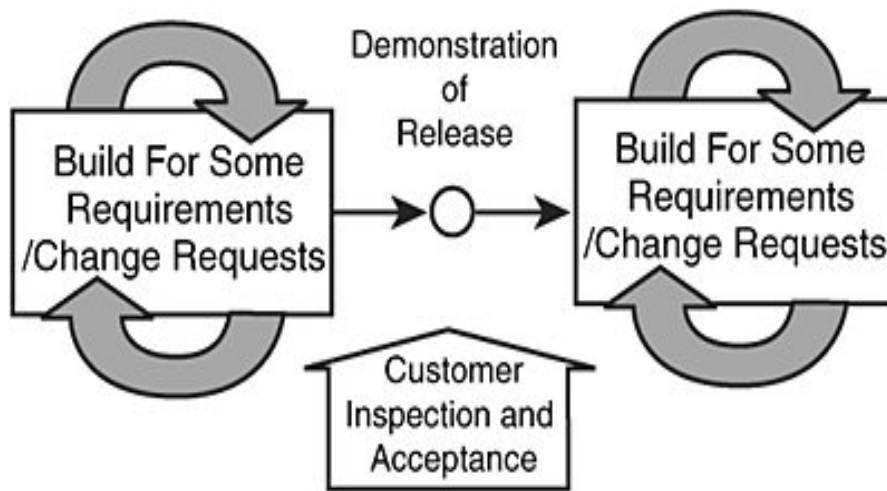
- Cyclic process model is based on development periods of same duration.
- Duration is typical one to four weeks. The smaller the better.

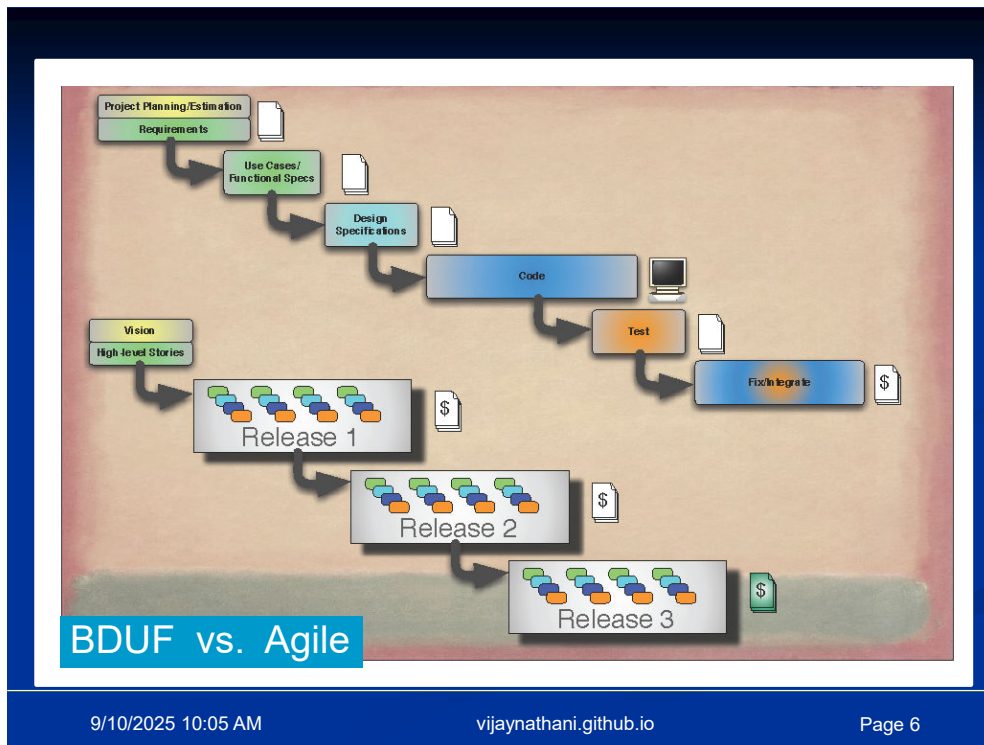
Iterations are time-boxed and have ritual-enforced boundaries. Progress feedback based on increments become meaningless, if the iteration duration is variable on demand. Everyone knows and feels when an iteration ends. It is signaled with a demo and a retrospective.

Increment refers to the product

- How much more product has been produced since the last iteration?
- Increment needs to be consumable rather than a software-centric artifact

## Regular Customer Feedback





De-emphasizing dependencies between features

Time boxed more than Phased

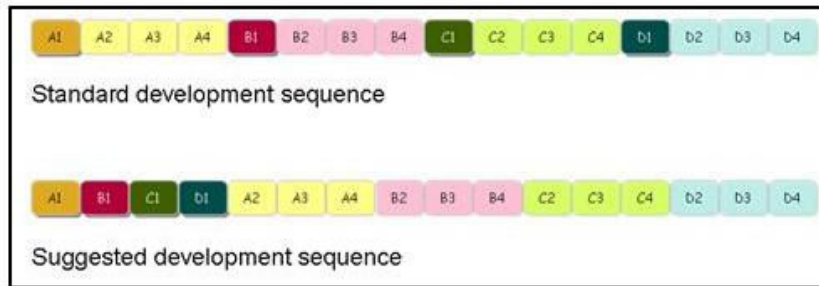
Self-similar through the lifecycle

Measured project velocity

Continuous specifications, design, coding and testing activities.

Early development validates planning estimates

# Predictive vs. Agile



## Planning development to add value quickly

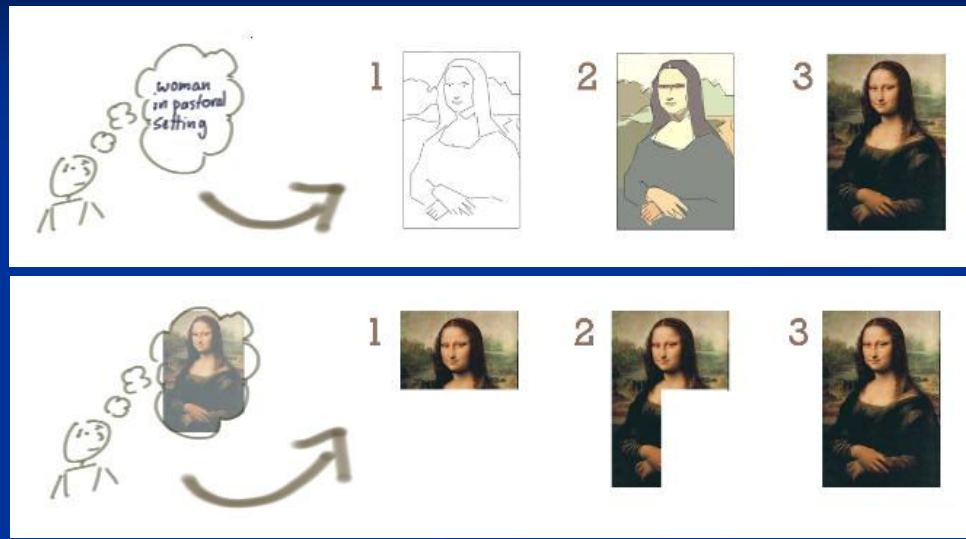
A,B,C,D are components

A1,B1,C1,D1 add 25% value for each component

Together, these 4 add 60% of overall value to customer

Do them first

## Which one is Iterative Development?



Only the first one.



## Incremental Development

- Version 1 was a static web page that described the product and provided a telephone number to call.
- Version 2 was a web form that captured the data the customer filled in.
- Version 3 hooked the web form into the back-end system directly.

<http://martinfowler.com/bliki/RollerSkateImplementation.html>

9/10/2025 10:05 AM

vijaynathani.github.io

Page 9

A key property of agile development is figuring out how to make a system go live with a small subset of features. We build software for the business value it offers, the quicker we go live, the faster we get at least some of that business value.

My colleague Dave Leigh-Fellows told me one of my favorite examples of this kind of thinking. It came when we had working for a brokerage firm. They had a new kind of product that they wanted to get into the market. The full software support for this was a web page that the customer filled in that generated the necessary transactions against the back-end system. But Dave came up with a way to get the product into the market faster than that.

Version 1 was a static web page that described the product and provided a telephone number to call. Some temporary staff then spoke to the customer and entered the information into the back end system.

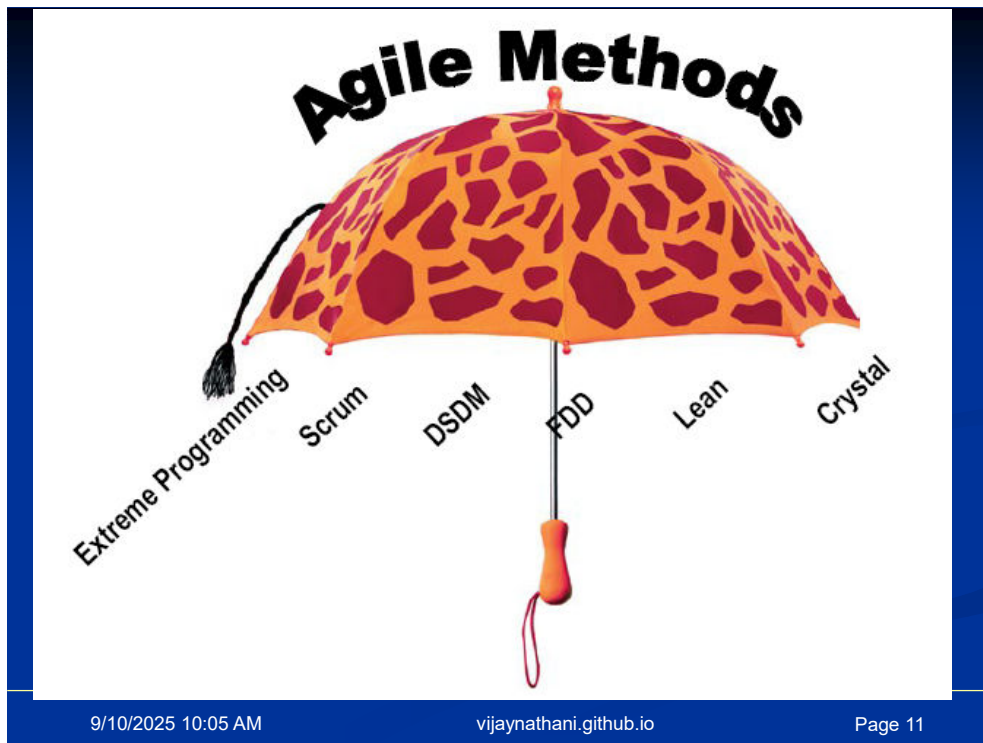
Version 2 was a web form that captured the data the customer filled in. However this version didn't load that data into the back end system. Instead the web form generated a fax. They hired some more temps to get the orders from the fax machine to the people that keyed the information into the back end system. Since the fax machines were a bit of a distance away, this is where the roller-skates came in.

Version 3 hooked the web form into the back-end system directly.

The first two versions may not have been the most elegant solutions ever conceived, but they did get the product into the market much more quickly. I've not come across any other examples of iterative development that use roller-skates, but that may be more due to lack of imagination rather than lack of need.

# Very Small Exposure Requires Very Little Ceremony





Ken Schwaber on MSF (Microsoft Solutions Framework), “[MSF for Agile] defines roles within the development team, so it isn’t cross-functional. It tells the team what to do, so it misses self-management. It defines what steps to take, so it isn’t empirical. It has increments that aren’t potentially shippable, so it isn’t lean. Lots of artifacts and no emerging list of requirements and architecture, so it is wasteful and not lean/agile.”

Removing many of the core principles that make Agile the most effective way to work isn’t the only issue. Microsoft chose to roll this process into their tools and the marketplace without independent peer review, without testing it with real teams, and with no comparative analysis on this new software development process vs. proven Agile methods such as Scrum. Should we expect better from the world’s largest software development tool vendor? Microsoft has enormous market presence, and as such has the opportunity to influence many billions of dollars of IT projects. Microsoft had the opportunity to make proven Agile methods the centerpiece of a tool suite that could help customers make the transition to much improved productivity. Instead Microsoft co-opted an important brand (Agile), but delivered a process that is missing key Agile principles and practices, and cannot provide the business value seen by teams adopting Scrum, XP, or other market tested and proven methods.