Introduction to Agile Software Development



Dario Campagna

Head of Research

Agile Development is **adaptive** rather than predictive; **people-oriented** rather than process-oriented.

Martin Fowler







Agile, a bit of history



Some of the major events since 1950s

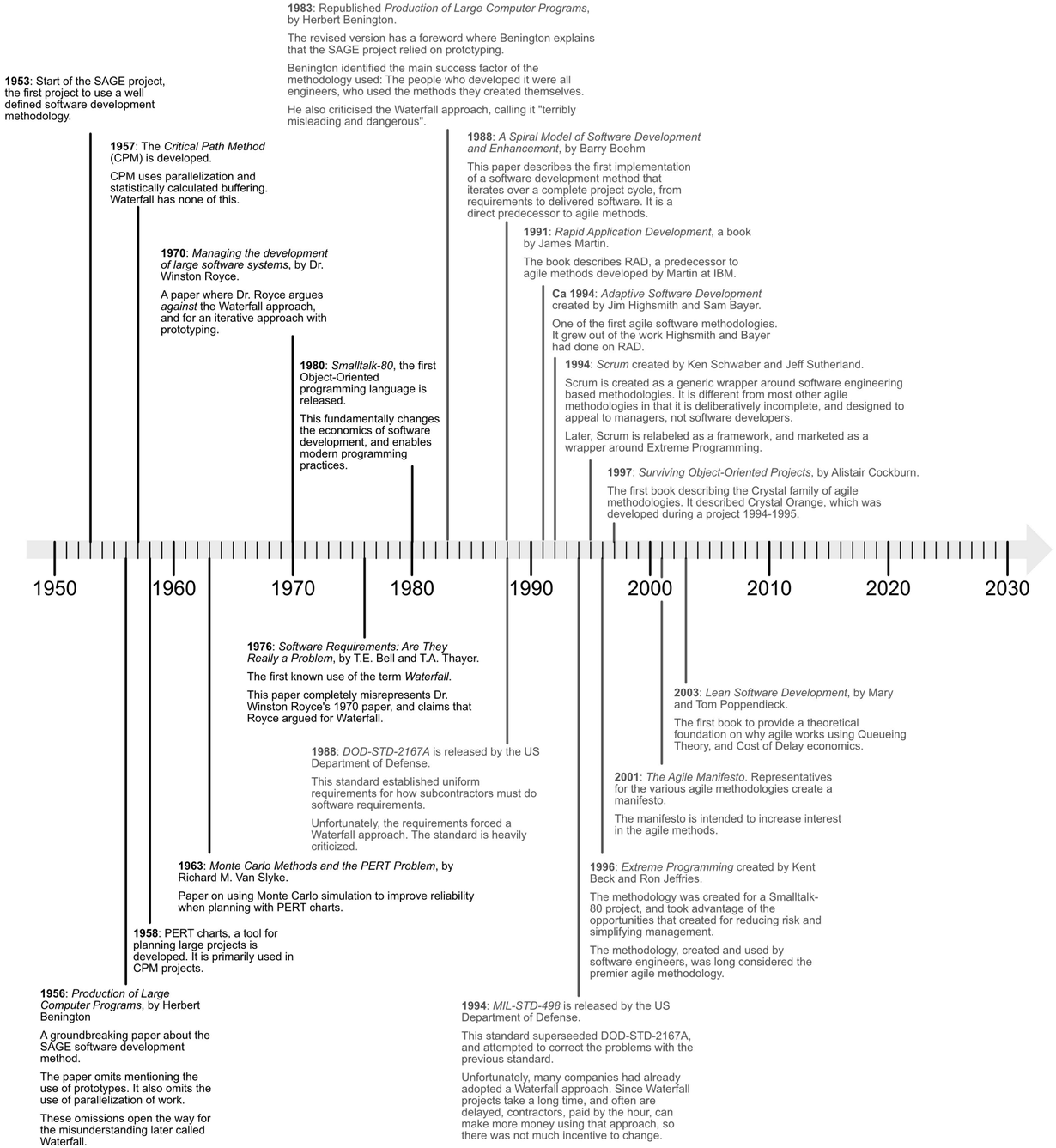


Figure from [“Waterfall vs. Agile: Battle of the Dunces or A Race to the Bottom?”].

The Semi-Automated Ground Environment (SAGE)

The first software project to use a well defined development methodology



Part of the SAGE computer room.



A large computer program, written on punch cards.

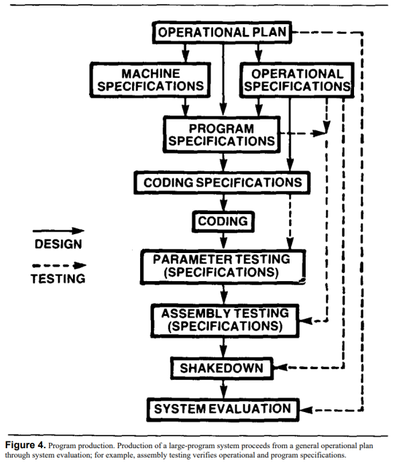


Figure 4. Program production. Production of a large-program system proceeds from a general operational plan through system evaluation; for example, assembly testing verifies operational and program specifications.

An outline of the software development process from Benington's 1956 paper.

The most misunderstood methodology paper of all time

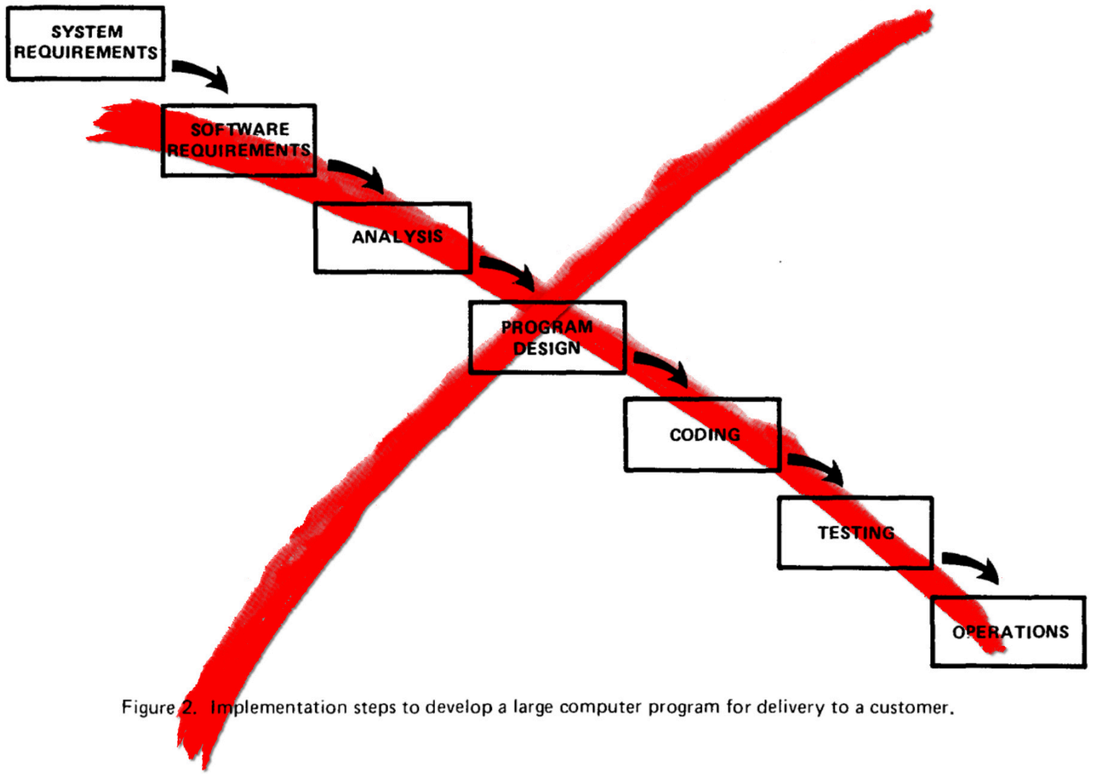


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

This is Waterfall. It is what Dr. Winston Royce argued *against* in his 1970 paper *Managing the Development of Large Computer Systems*.

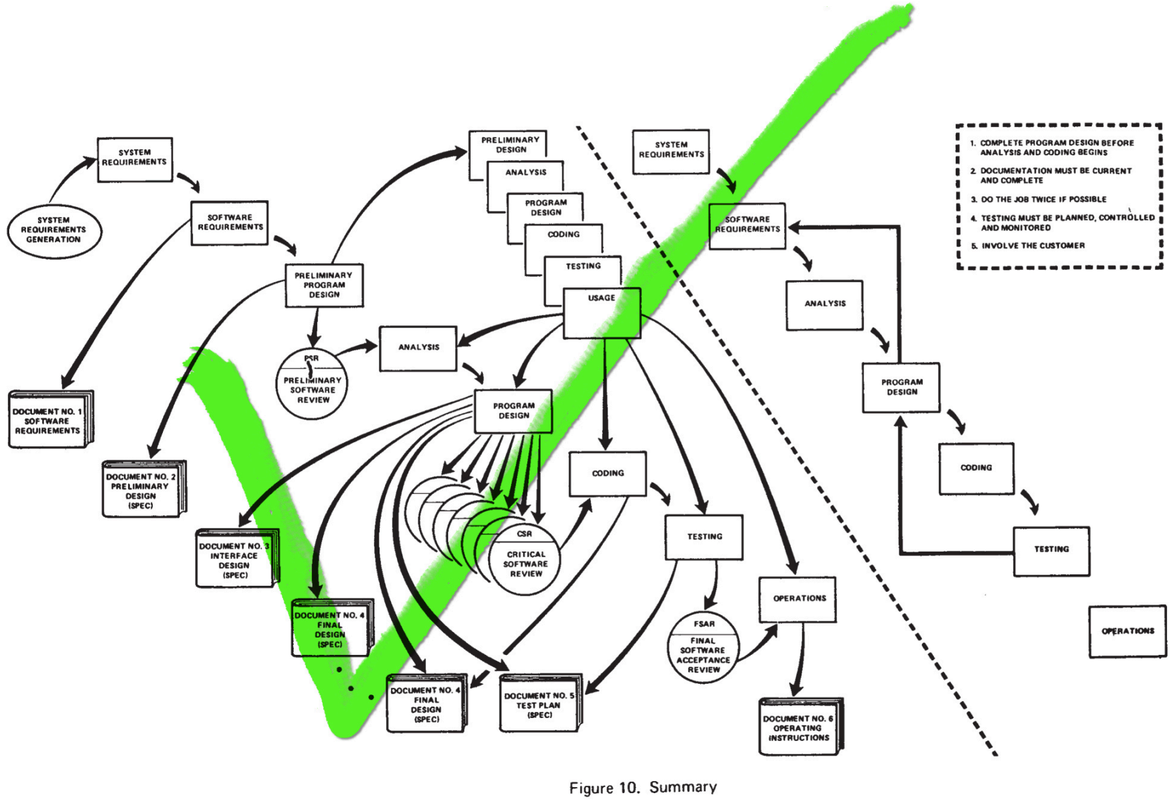


Figure 10. Summary

This is what Dr. Royce argued *for* in his 1970 paper: An iterative method, with prototyping, and close continuous contact with customers.

Figure from "Waterfall vs. Agile: Battle of the Dunces or A Race to the Bottom?".

DoD-Std-2167

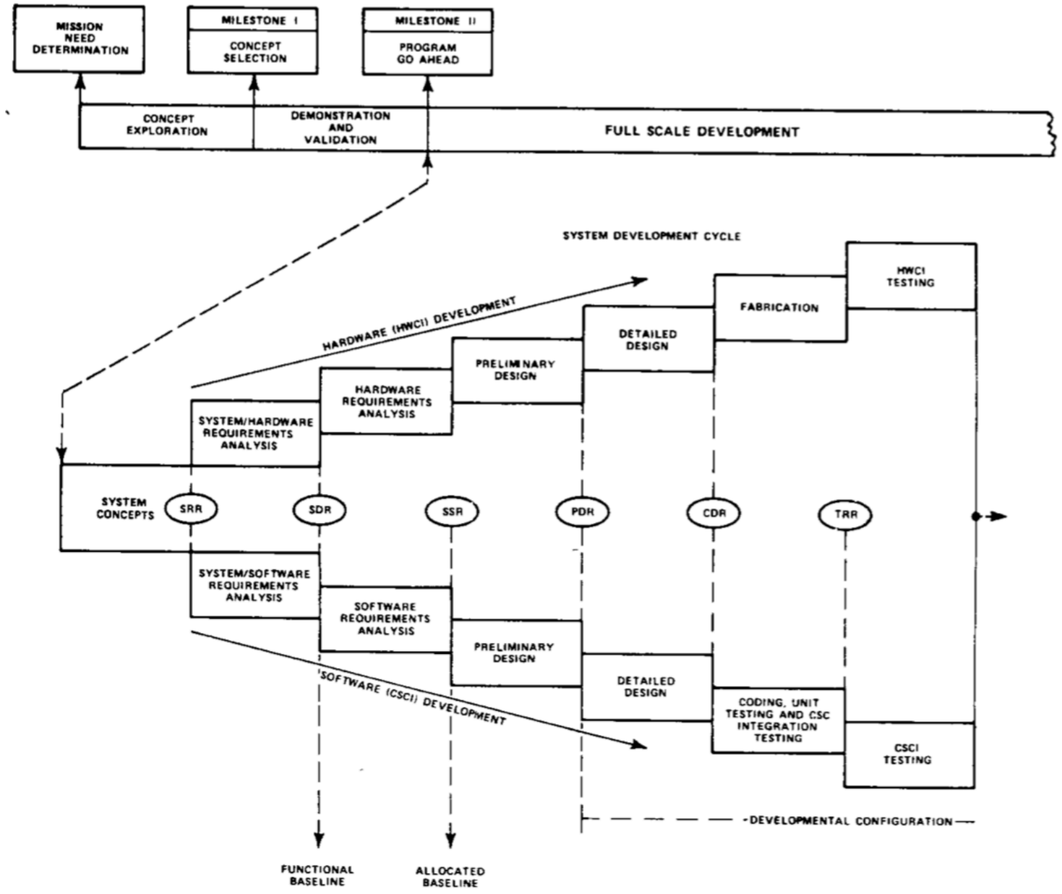


FIGURE 1. System development cycle within the system life cycle.

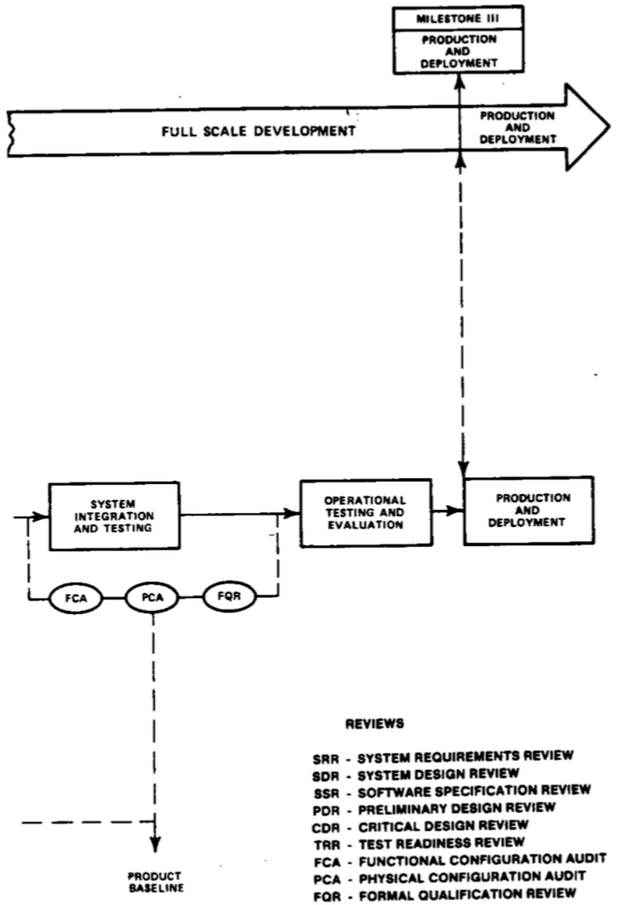


FIGURE 1. System development cycle within the system life cycle. (continued)

DoD-Std-2167 and project failures

1987 report of the Defense Science Board Task Force on Military Software.

> Directive 5000.29 not only does not encourage this best modern practice, it essentially forbids it. We recommend that it be revised immediately to mandate and facilitate early prototyping before the baseline specifications are established (Rec. #23).
>
> DoD-STD-2167 likewise needs a radical overhaul to reflect best modern practice. Draft DoD-STD-2167A is a step, but it does not go nearly far enough. As drafted, it continues to reinforce exactly the document-driven, specify-then-build approach that lies at the heart of so many DoD software problems.

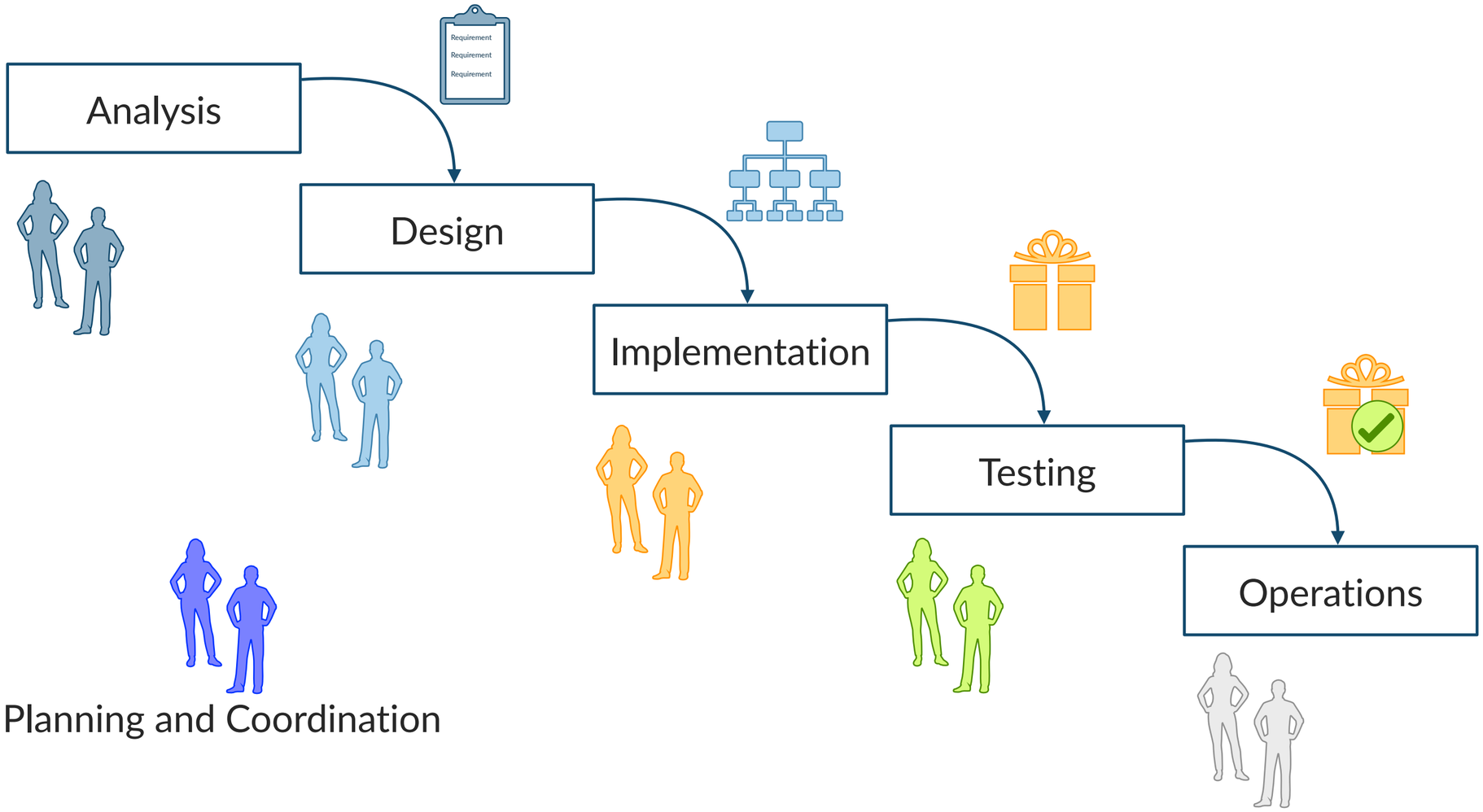
Manifesto for Agile Software Development

Utah, 2001. A group of 17 experts (with different backgrounds) met to discuss the growing field of what used to be called lightweight methods.

- Captures common ground about software development
- Sets out the values and principles of these lightweight methods
- Covers technical, organizational and people aspect of software development



Specify-then-build approach and People





Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

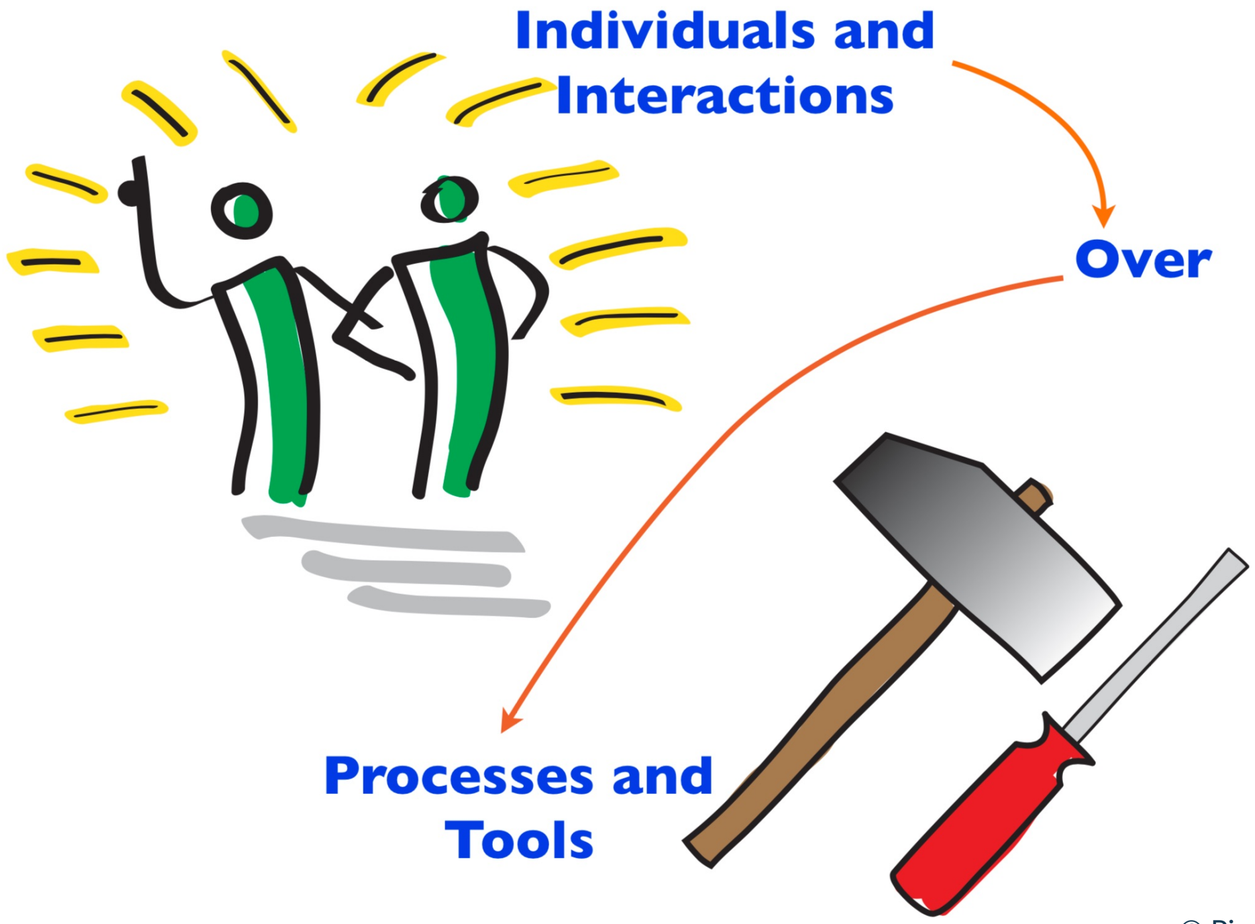
Through this work we have come to value:

[X over Y]

That is, while there is value in the items on the right, we value the items on the left more.

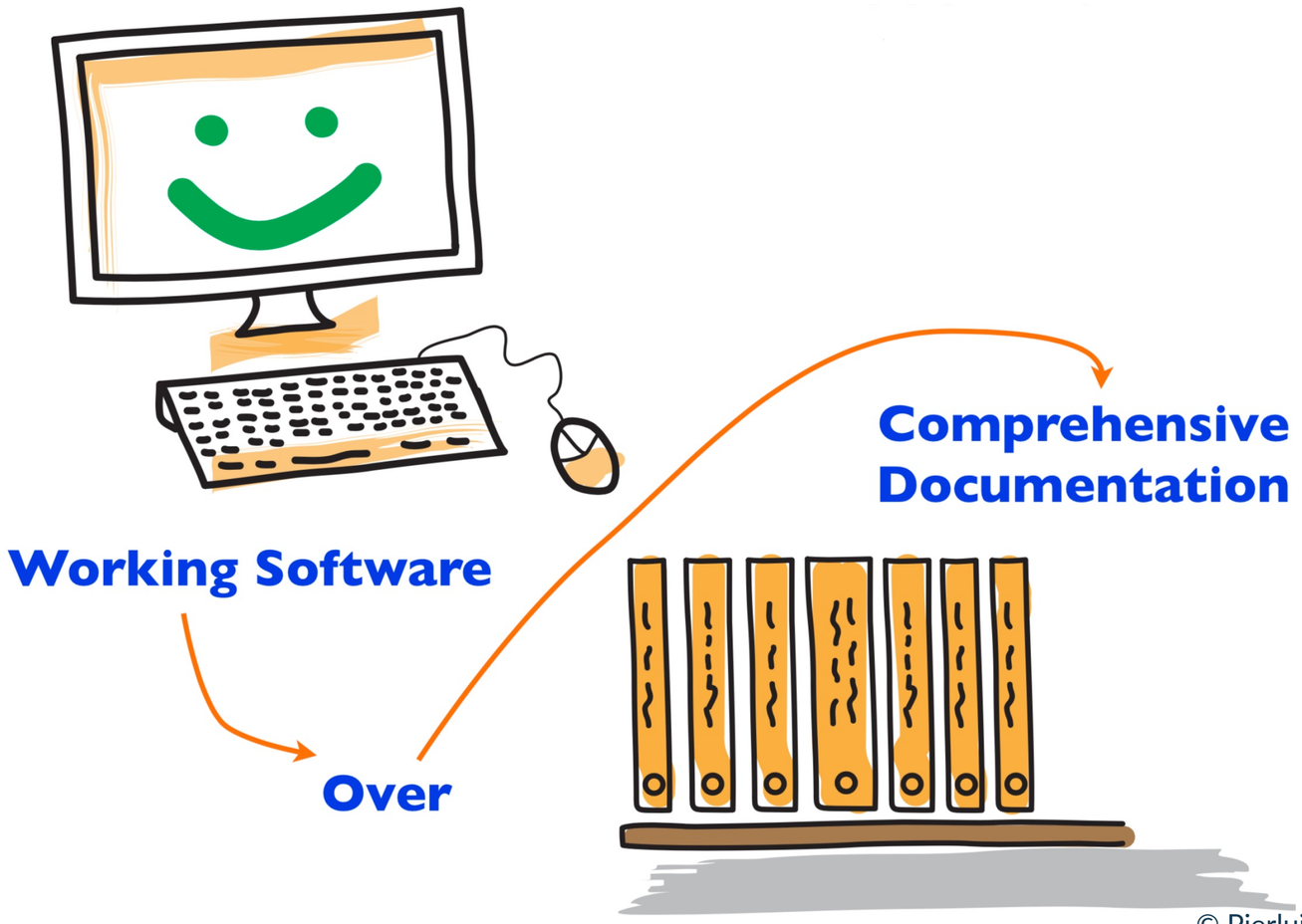


We value...



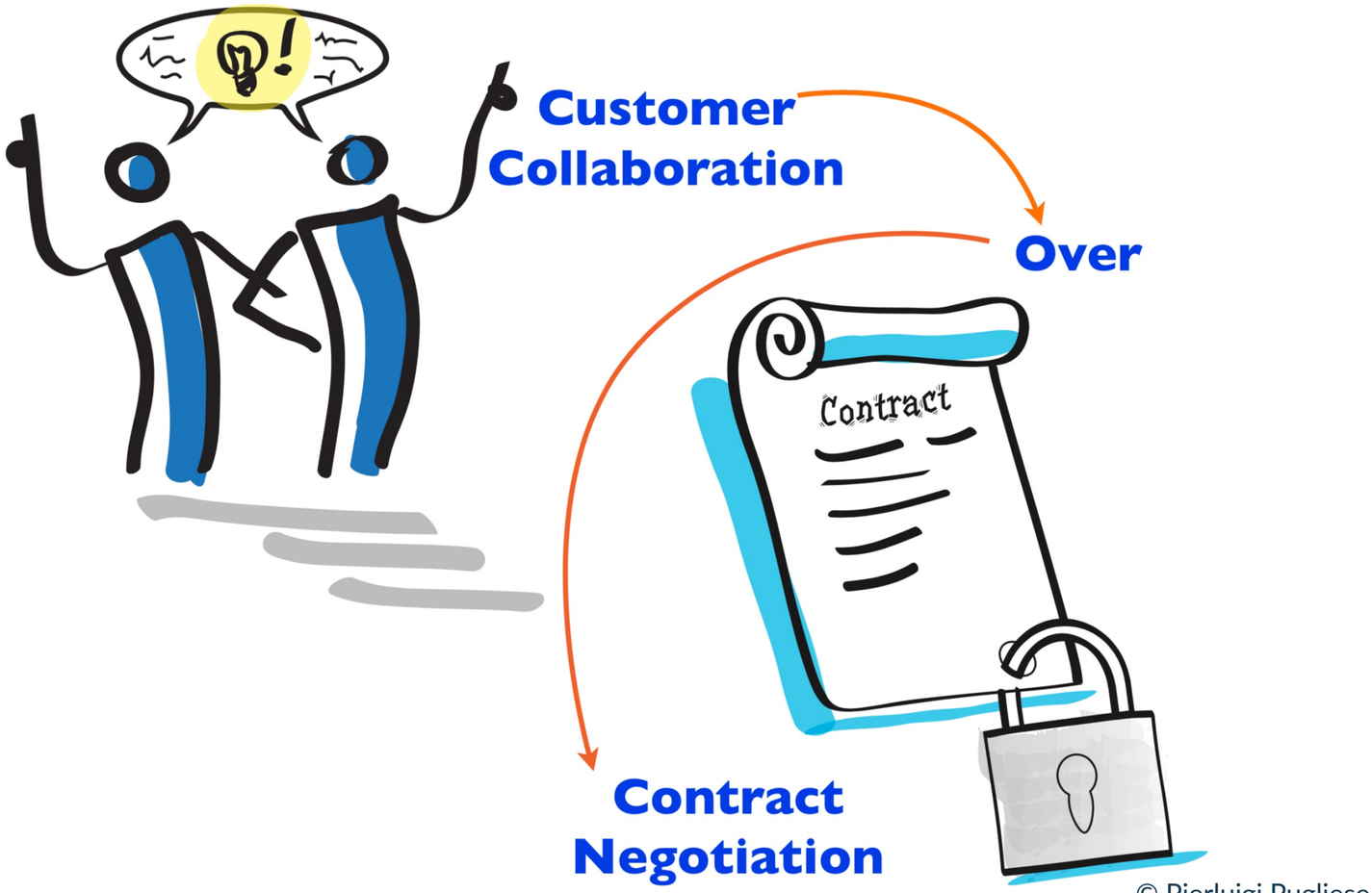
© Pierluigi Pugliese

We value…



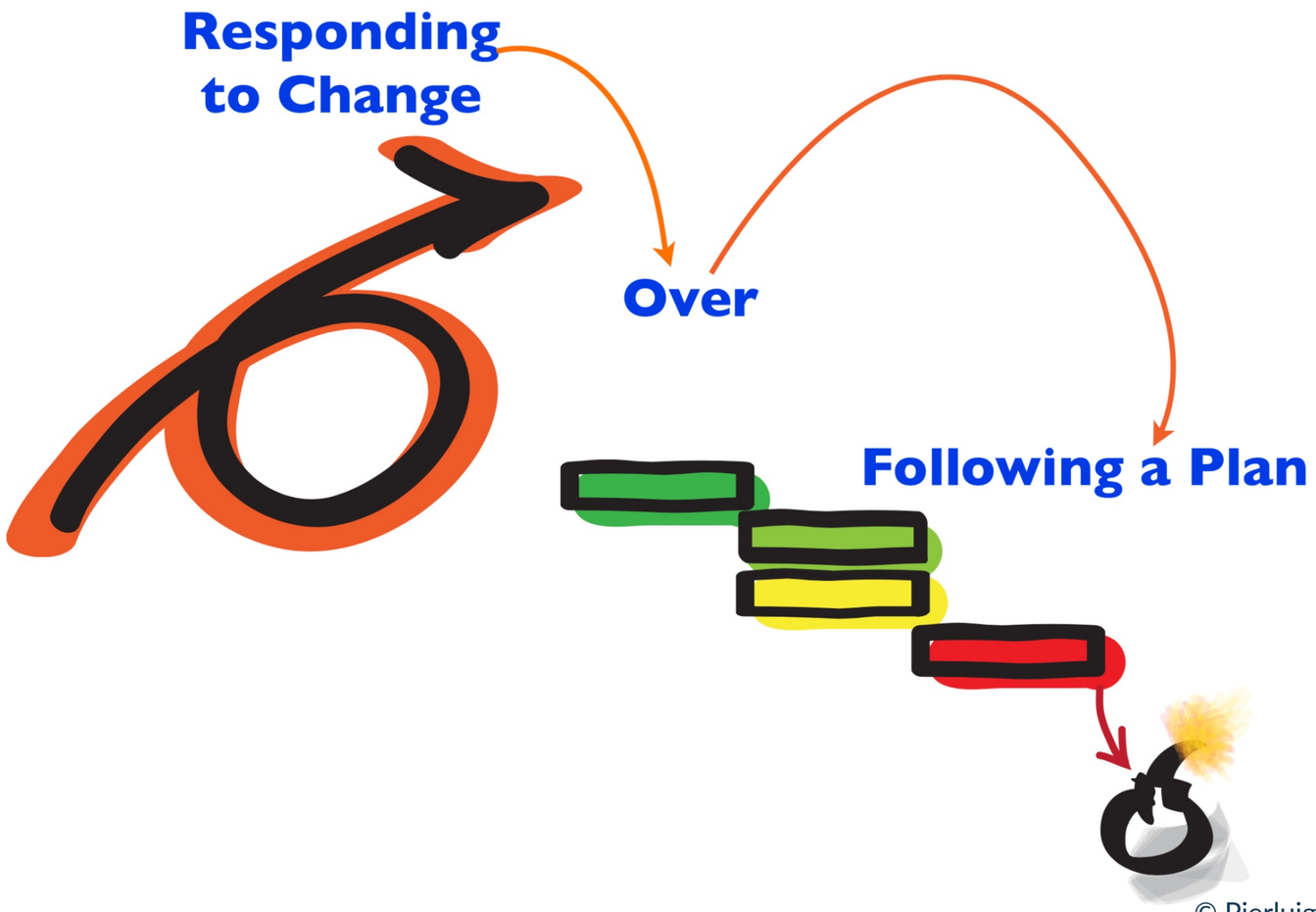
© Pierluigi Pugliese

We value…



© Pierluigi Pugliese

We value…



© Pierluigi Pugliese

Agile Manifesto Principles

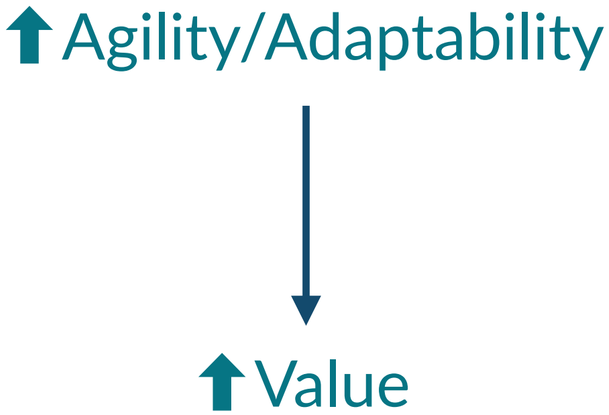
1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	4	Business people and developers must work together daily throughout the project.
2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	5	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Agile Manifesto Principles

7	Working software is the primary measure of progress.	10	Simplicity – the art of maximizing the amount of work not done – is essential.
8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	11	The best architectures, requirements, and designs emerge from self-organizing teams.
9	Continuous attention to technical excellence and good design enhances agility.	12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Characteristics of Agile

Adaptability



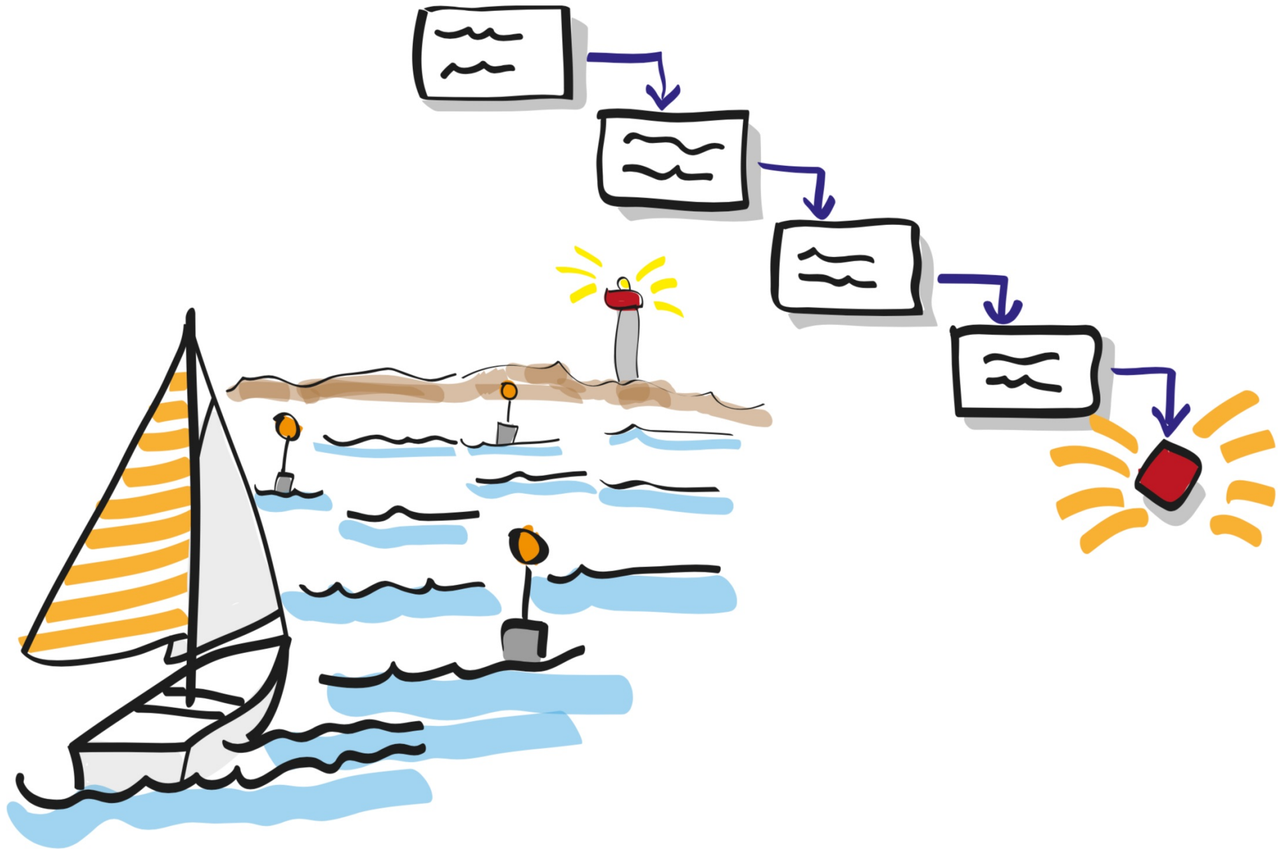
Adaptability as a driver for value.

Agility ≠ Fast

Agility ≠ Cheap

Fast and cheap are not goals for Agile.

Empirical Vs Defined Process

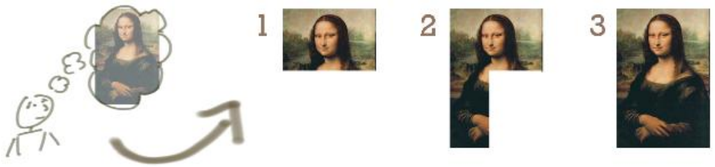


© Pierluigi Pugliese

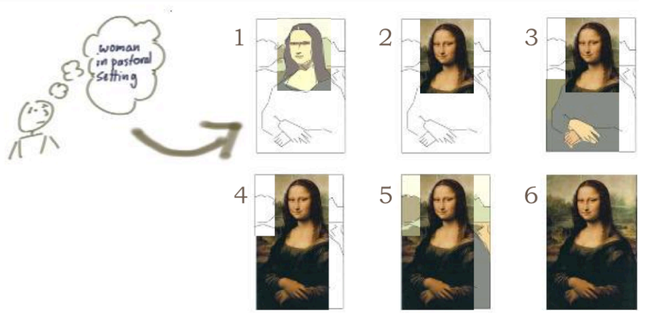
Iterative and Incremental Development



Iterate to evaluate and make changes to what you've already made

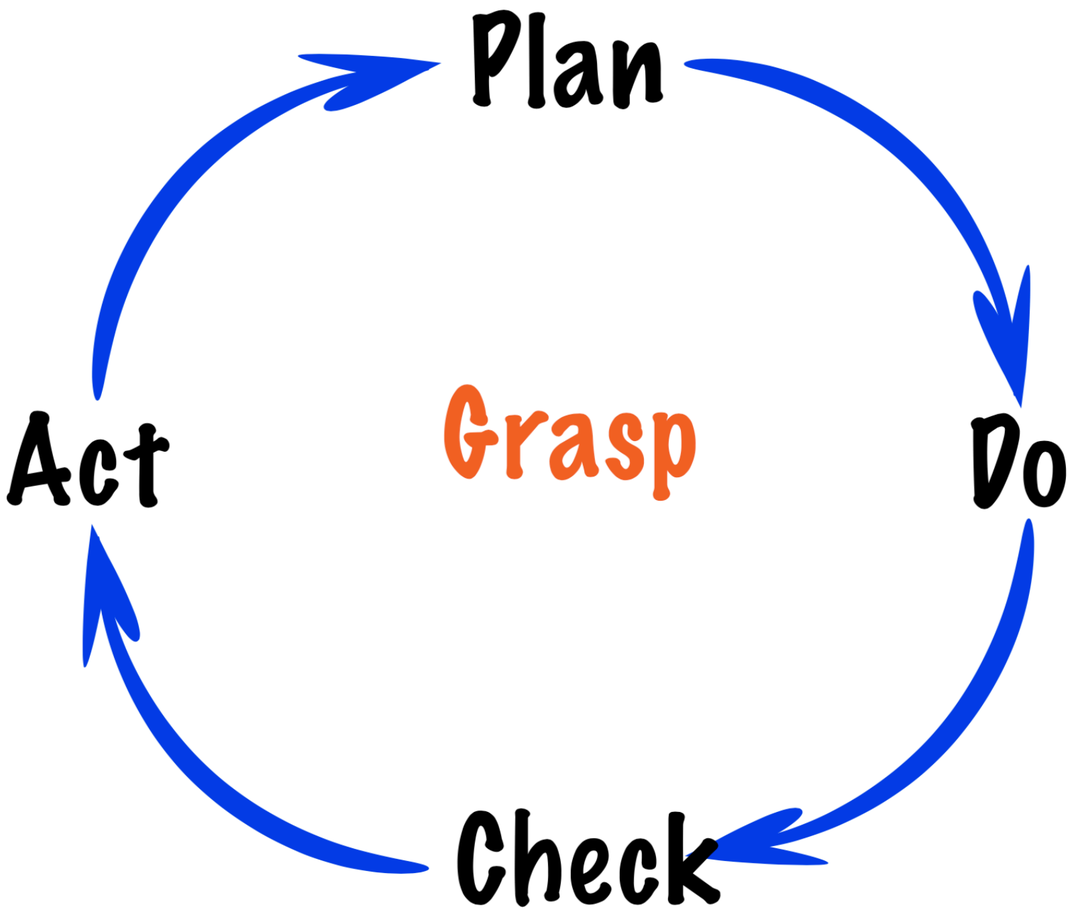


Increment to make additions



The two tactics can be **conjoined**

Continuous Improvement



© Pierluigi Pugliese



Quick introduction to Scrum

Takeuchi and Nonaka

“The New New Product Development Game”

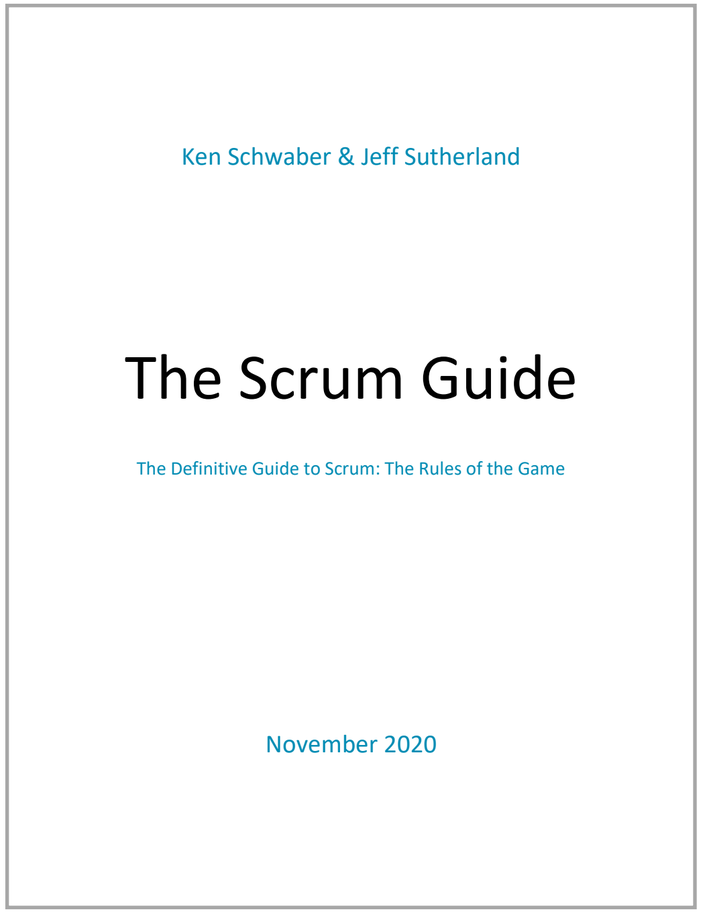
- Used the term Scrum
- Referred to the game of rugby to stress the importance of teams
- Their research showed that teams requires autonomy to achieve excellence



Scrum is…

A lightweight framework that helps people, teams and organization generate value through adaptive solutions to complex problems.

- Simple and purposefully incomplete
- Founded on empiricism and lean thinking
- Iterative and incremental approach
- Cross-functional and self-managing team



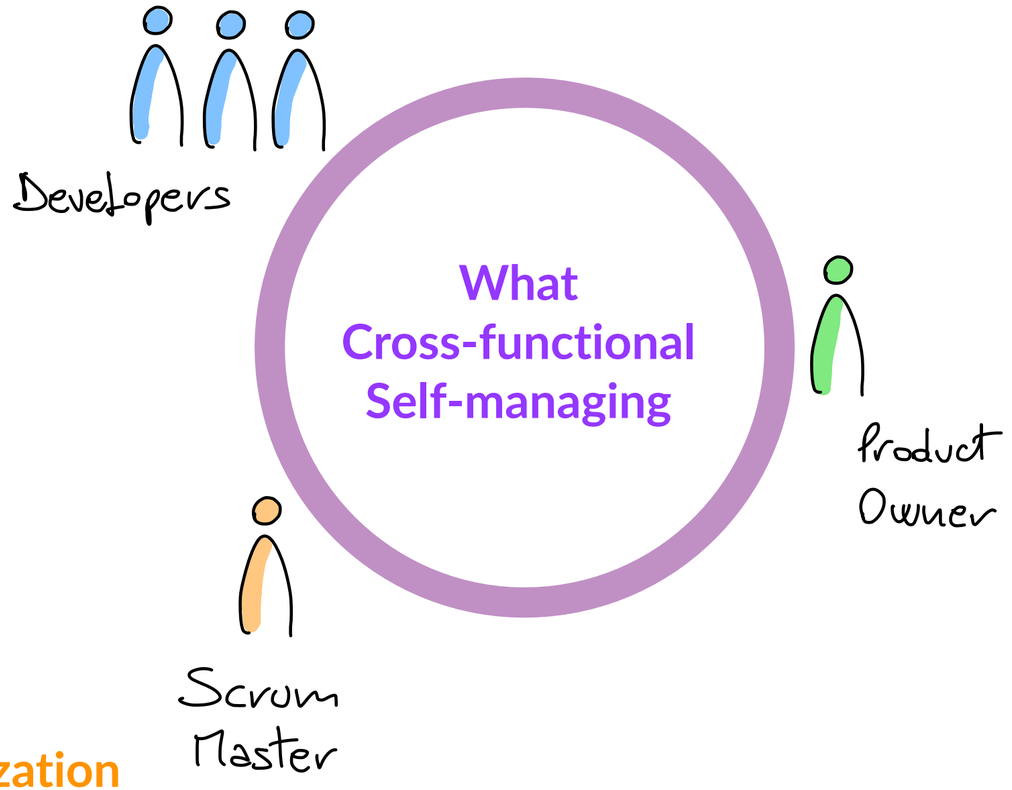
Scrum makes visible the relative efficacy of your current management, environment and work techniques, so that improvements can be made.

Ken Schwaber and Jeff Sutherland - The Scrum Guide™



Scrum Team

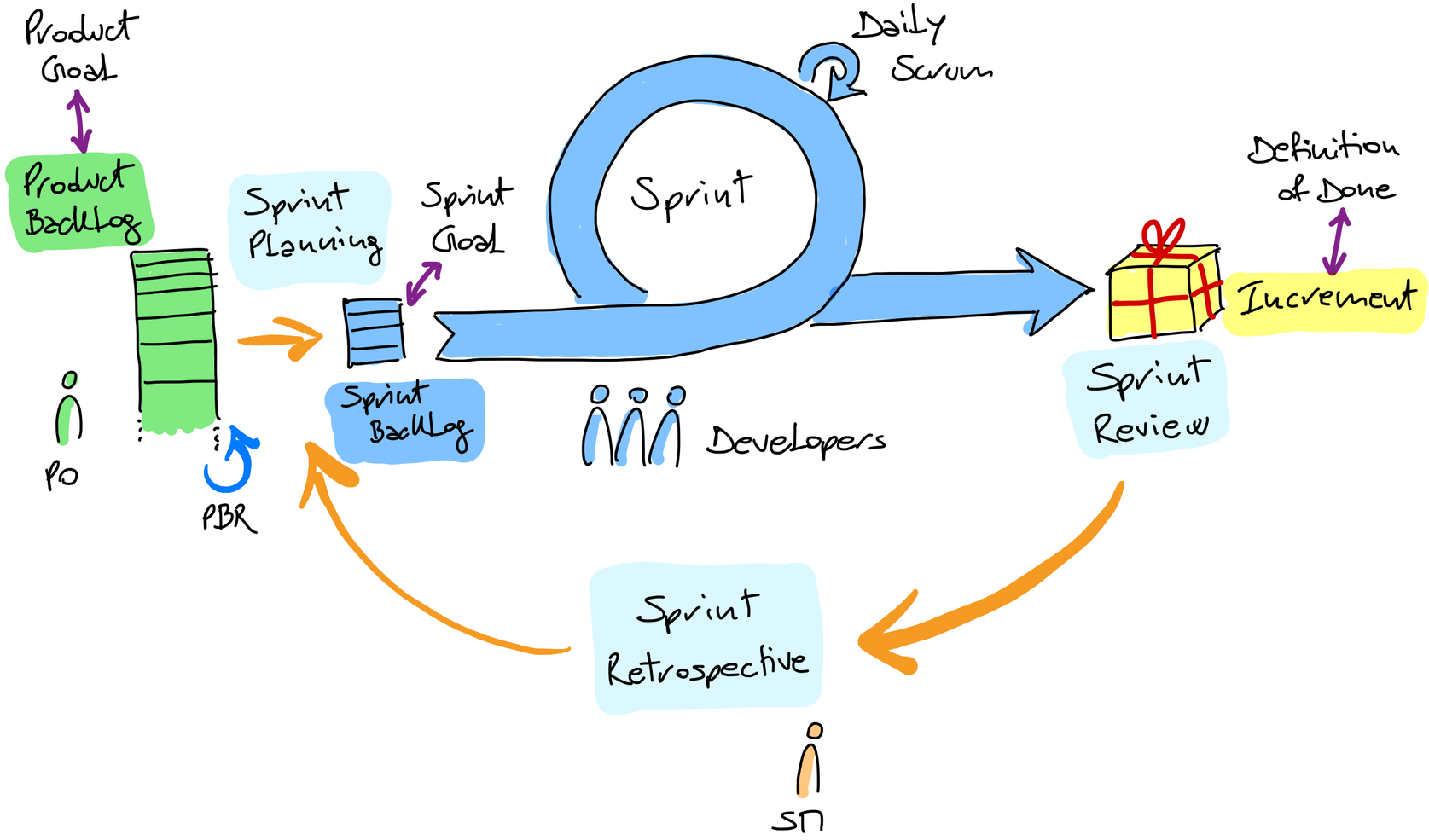
How
Whole product focus
Quality



Why
Return on investment
Business decisions

Scrum
Team
Organization

Events and Artifacts



References



Agile Manifesto
https://agilemanifesto.org

Scrum Guide
Jeff Sutherland, Ken Schwaber
https://scrumguides.org