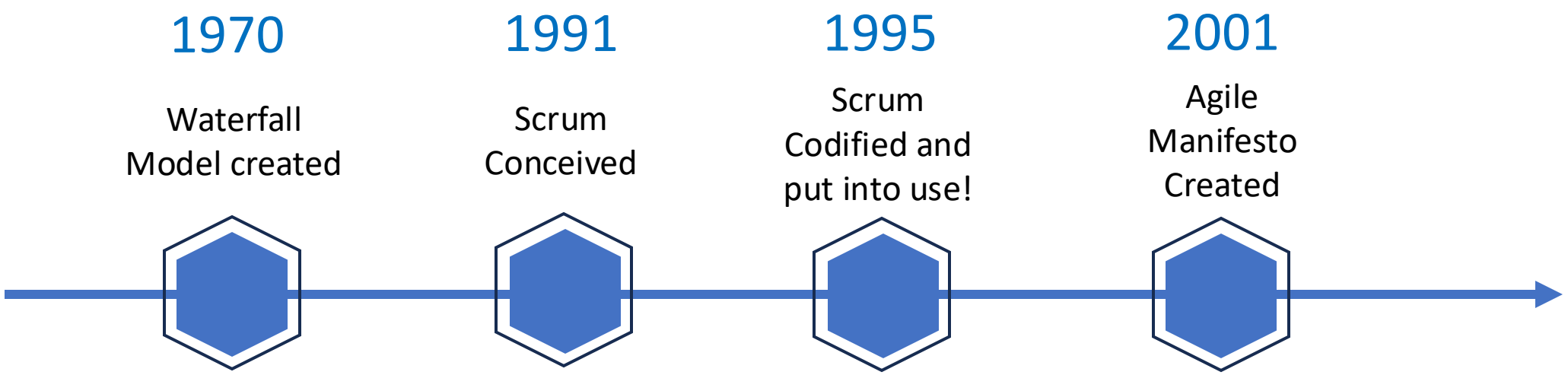
Software Engineering

From Waterfall to Agile

Waterfall to Agile



SOFTWARE ENGINEERING

Report on a conference sponsored by the

NATO SCIENCE COMMITTEE

Garmisch, Germany, 7th to 11th October 1968

Chairman: Professor Dr. F. L. Bauer

Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

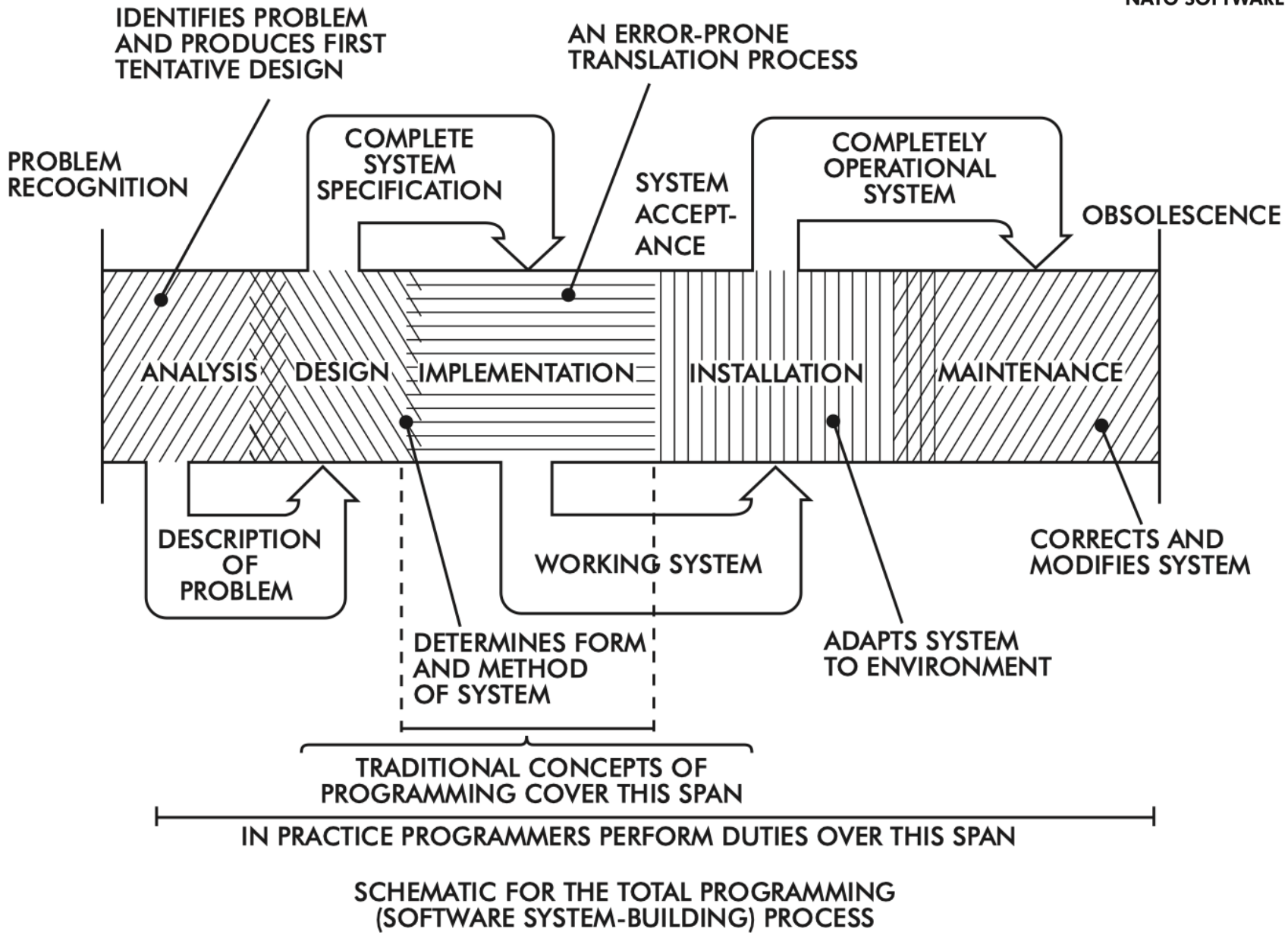
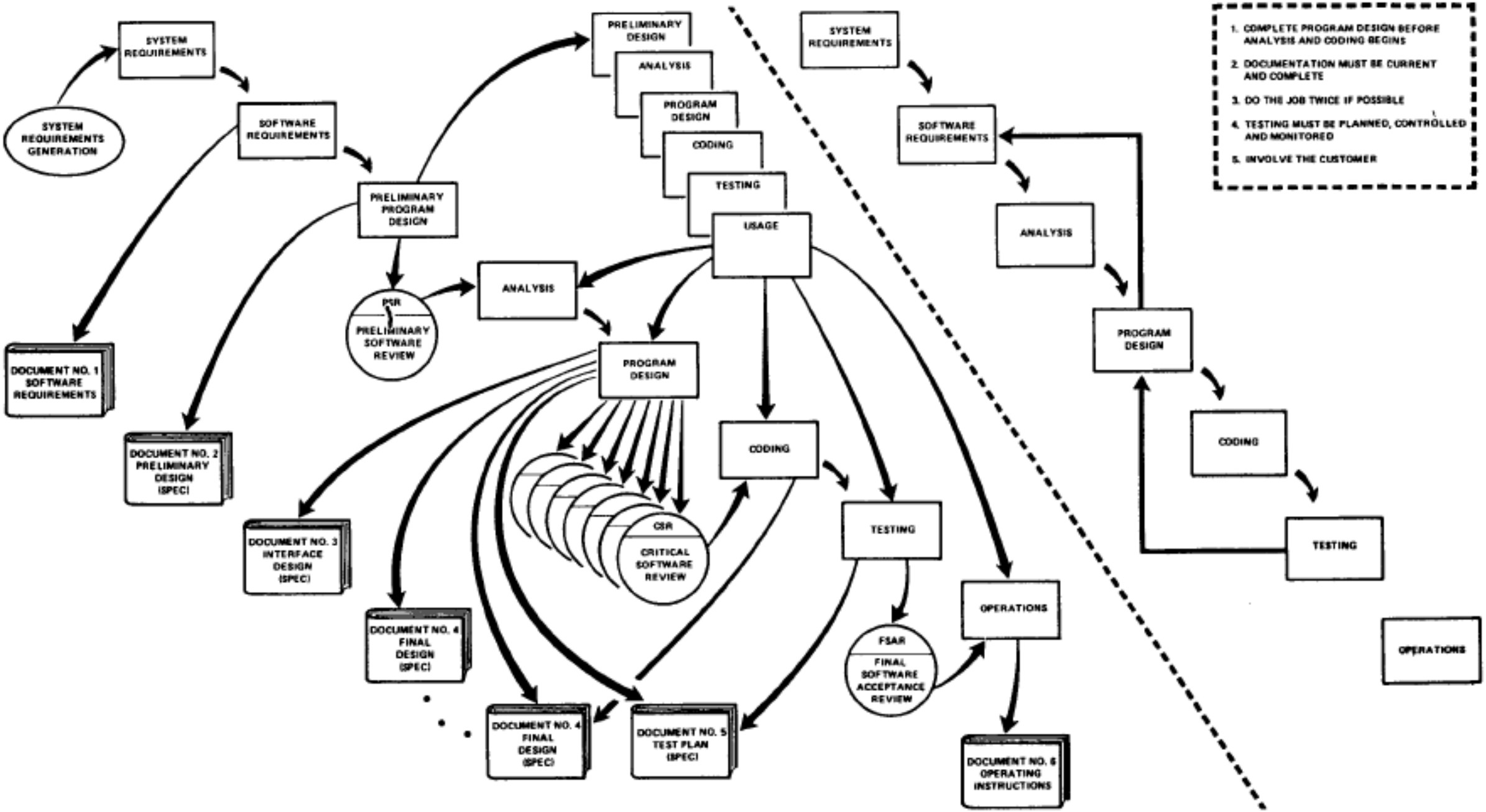
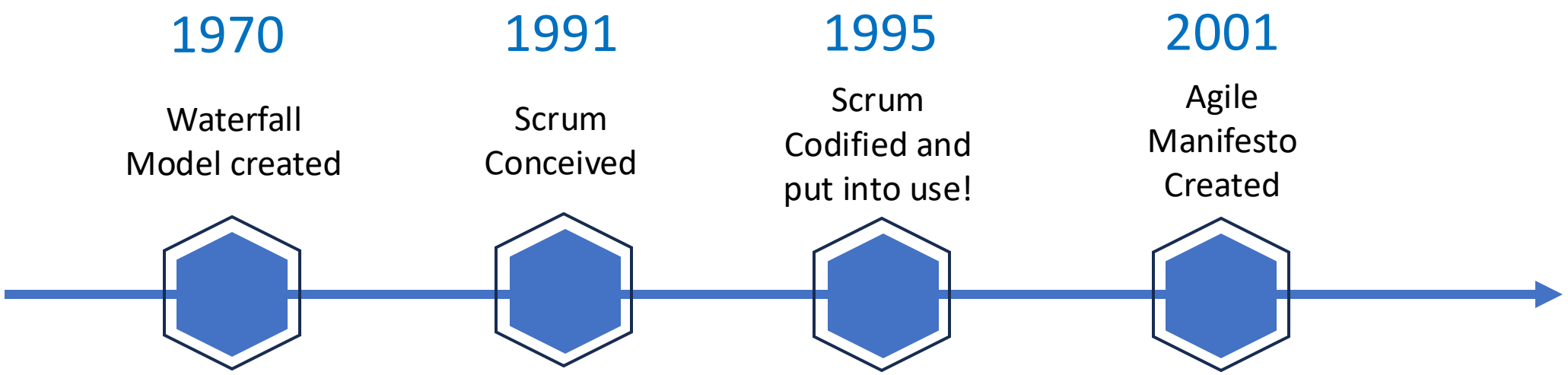


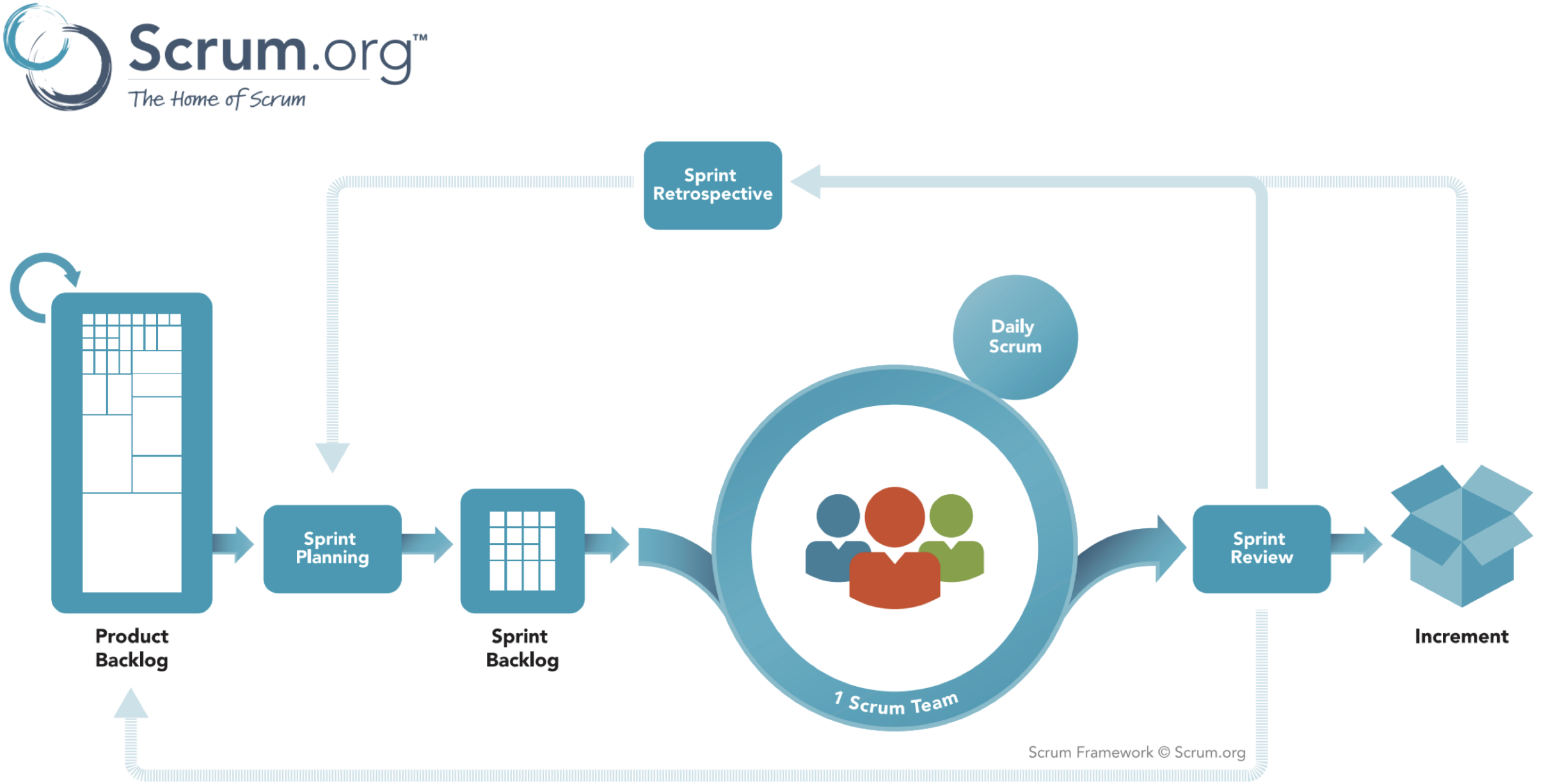
Figure 2. From Selig: Documentation for service and users. Originally due to Constantine.



1970 Winston W. Royce's Final model, published in 'Managing the Development of Large Software Systems'

Waterfall to Agile





Ken Schwaber and Jeff Sutherland 1995 - https://www.scrum.org/resources/what-scrum-module

Scrum.org

COURAGE
Scrum Team members have courage to do the right thing and work on tough problems

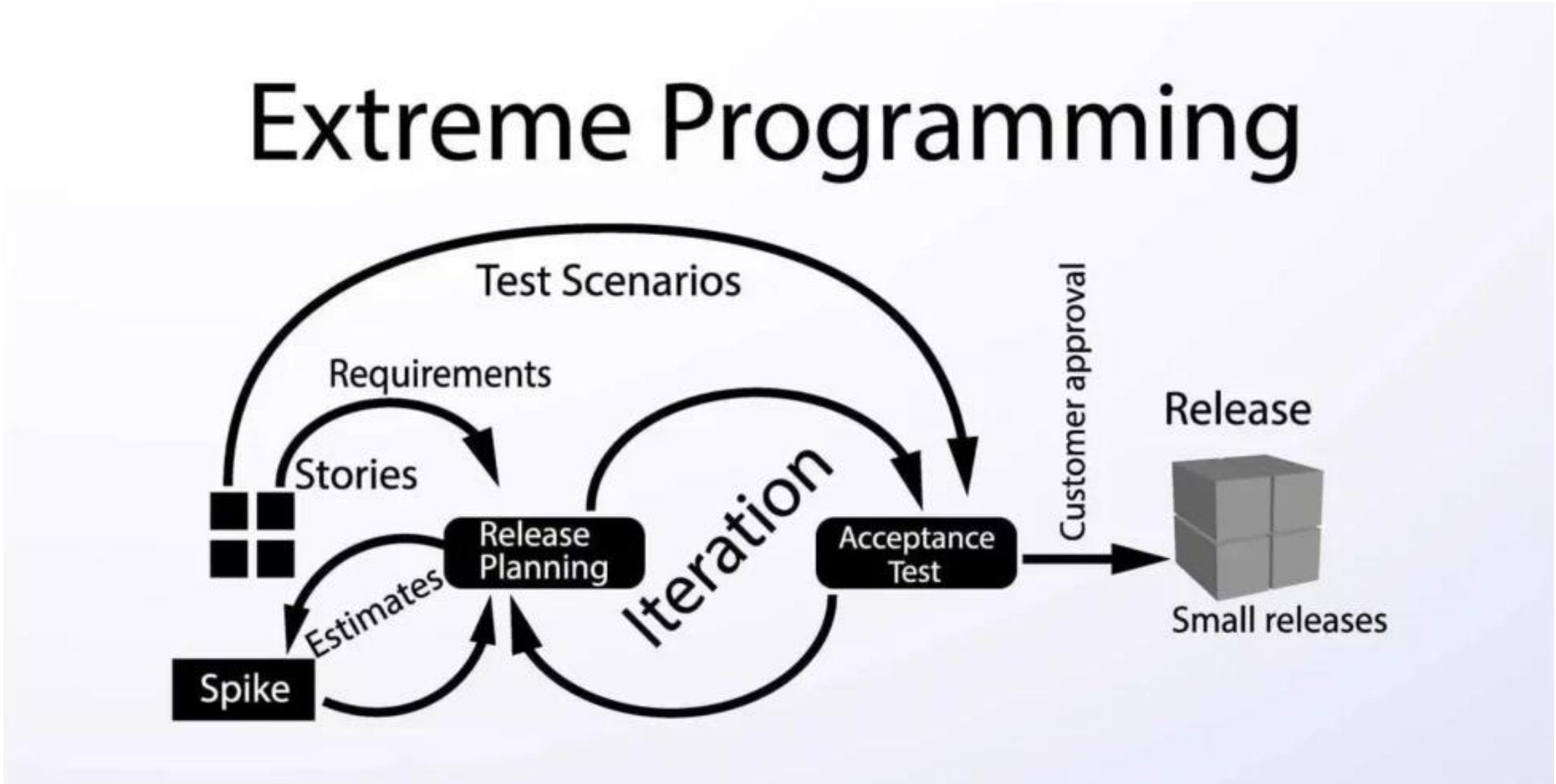
FOCUS
Everyone focuses on the work of the Sprint and the goals of the Scrum Team

COMMITMENT
People personally commit to achieving the goals of the Scrum Team

RESPECT
Scrum Team members respect each other to be capable, independent people

OPENNESS
The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work

Extreme Programming



Kent Beck 1999 - https://www.oreilly.com/library/view/extreme-programming-explained/0201616416/

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:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

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

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Beck et al. 2001 - https://agilemanifesto.org

Principles behind the Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity--the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Beck et al. 2001 - https://agilemanifesto.org

Principles behind the Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

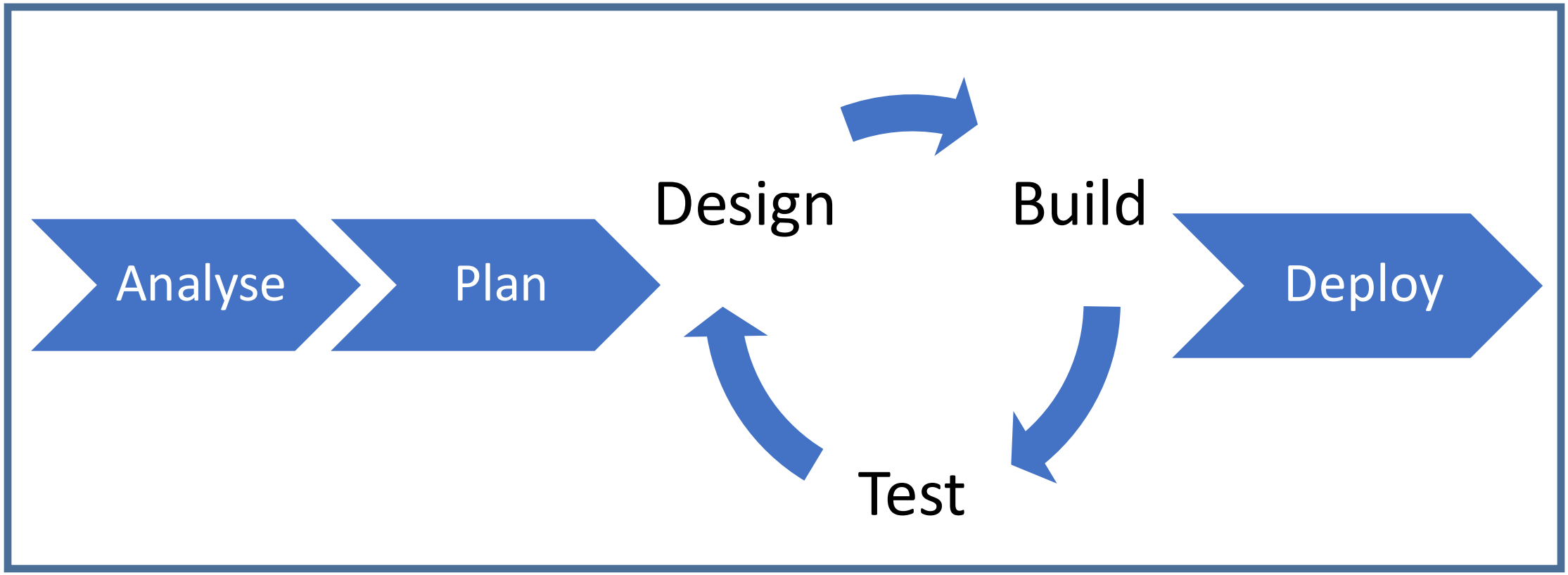
10. Simplicity--the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

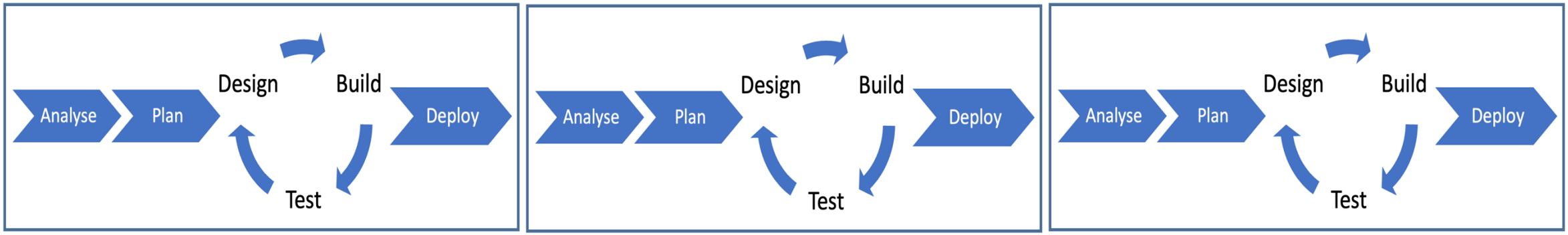
Beck et al. 2001 - https://agilemanifesto.org

Implementation: a typical ‘sprint’

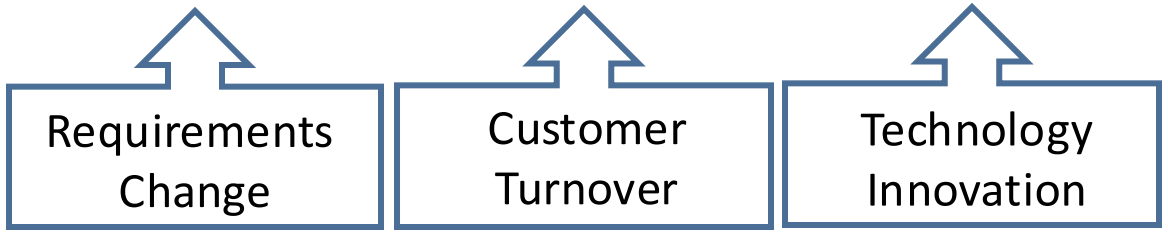


Waterfall

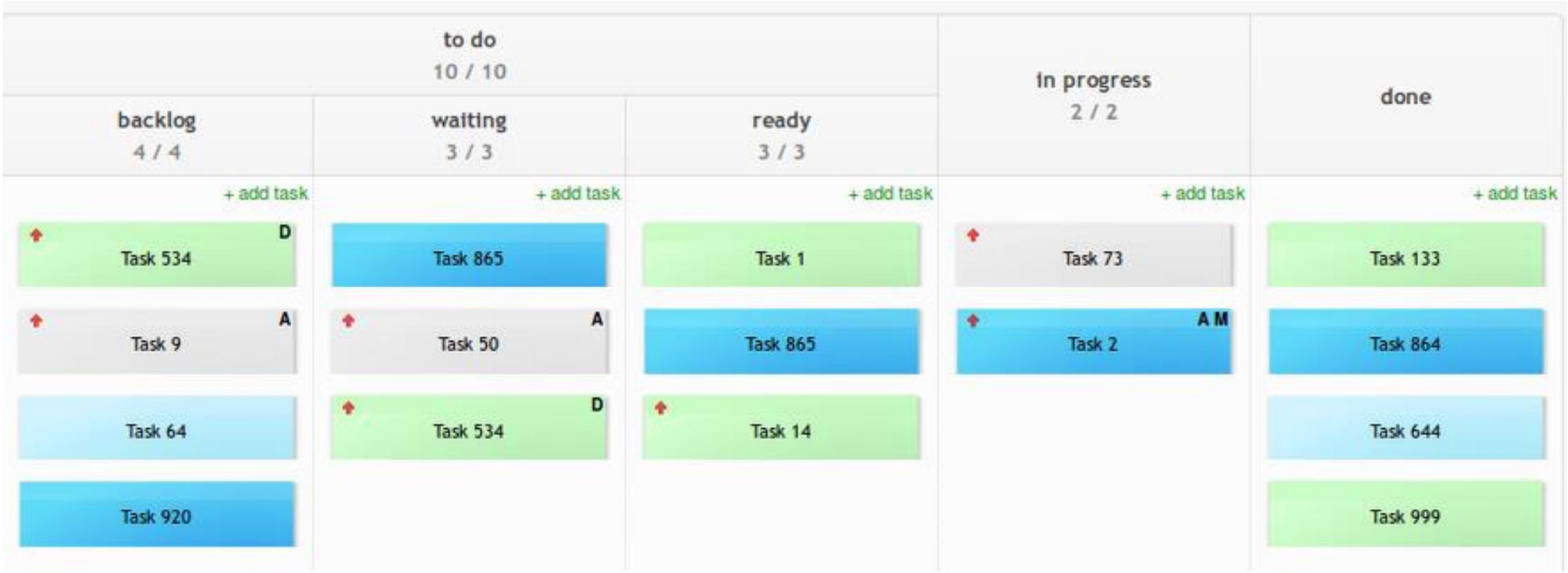
Agile



Project Timeline



Kanban Board (also Agile)



Kanban boards originated at Toyota Motor Company in 1963.
Software developers began using the Kanban method around 2006.

To Do (3)

Task 1

Default Task List

Mar 16

Task 2

Default Task List

Task 3

Default Task List

Add a description of your choice

Apr 20

+ Add Task

In progress (3)

Task 4

Default Task List

1

Task 5

Default Task List

Task 6

Default Task List

+ Add Task

Complete (2)

~~Task 8~~

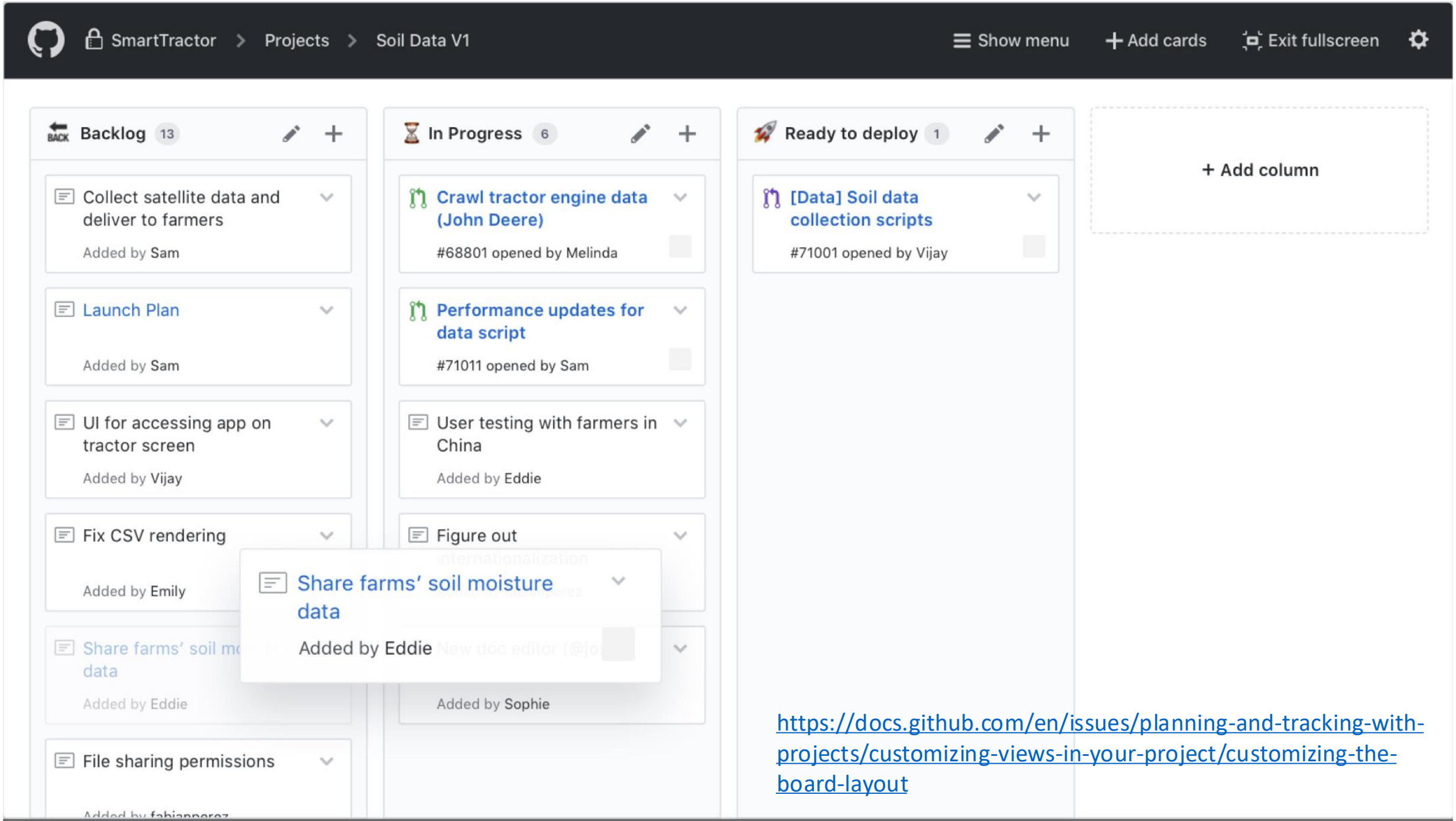
Default Task List

Task Description

~~Task 7~~

Default Task List

Feb 16 - Feb 28



https://docs.github.com/en/issues/planning-and-tracking-with-projects/customizing-views-in-your-project/customizing-the-board-layout

Assignment task

Your task is to form a group (recommended between two and four members) to:

- Collaborate and develop a game in an object-oriented language of your choice, using version control **(40%).**
- Document the development stages of your game by making regular commits to your combined GitHub repository **(40%).**
- Provide a group explanation of your game concept and demonstrate how your game can be played (this could be live or pre-recorded) **(20%).**

Below are the minimum requirements which your game must fulfil. Given that this an open assignment, the general requirements below should describe most game ideas out there.

- Create a 2D game with a minimum of one level.
- The playable character should have some form of attribute (health, energy, magic, strength, etc.) level which can increase or decrease.
- The playable character should be able to collect and store items and then use items.
- There should be some form of 'opposition' (e.g. enemies, time limit, reduction in energy as player moves around the level, etc.) that makes the objective of the game more challenging to achieve.

Example Game Requirements (Space Invaders)

- The Defender should move left and right and be able to fire bullets
- Invaders must alternate direction as they move down the screen
- Invaders speed of travel should increase as fewer remain and levels progress
- Shields should crumble as they shot by both the Invader and the Defender

Space Invaders

Nicholas Day edited this page now · 1 revision

Requirements

- The Defender should move left and right and be able to fire bullets
- Invaders must alternate direction as they move down the screen
- Invaders speed of travel should increase as fewer remain and levels progress
- Shields should crumble as they shot by both the Invader and the Defender

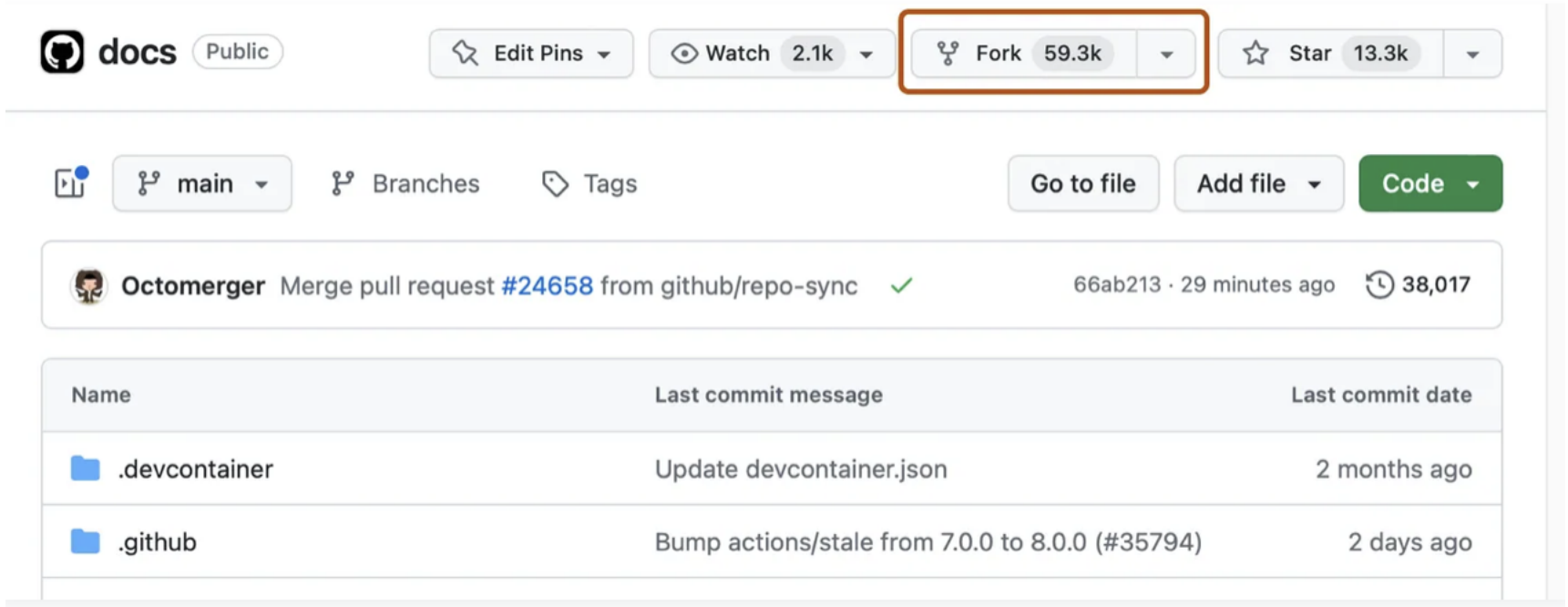

GitHub


What is
GitHub?

Forking or branching?

Forking

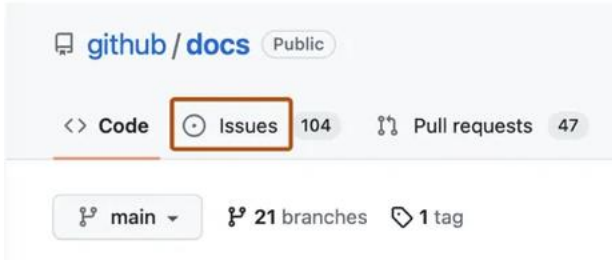
- Creates a local copy of the entire repo (by another author)
- Maintains the commit history from the original author

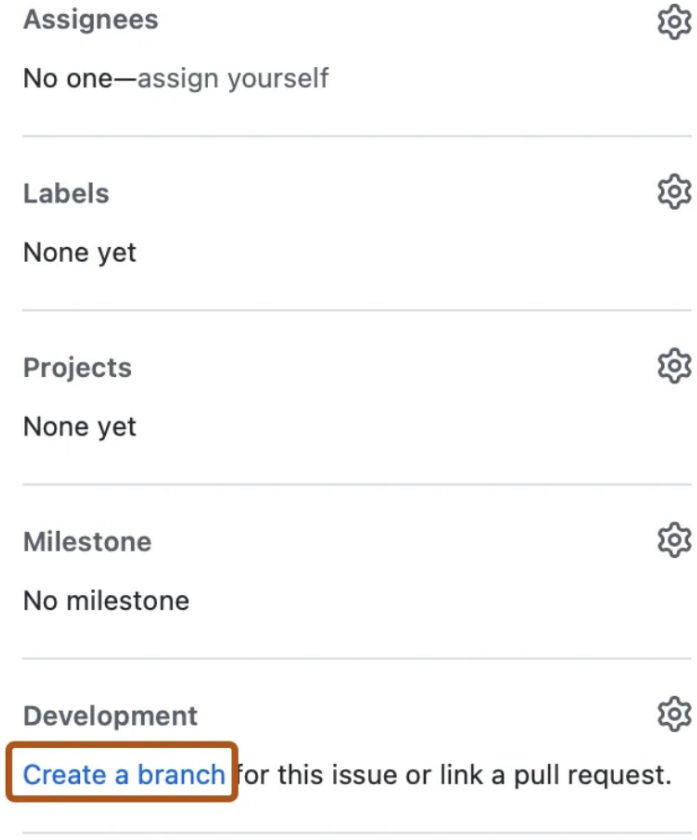


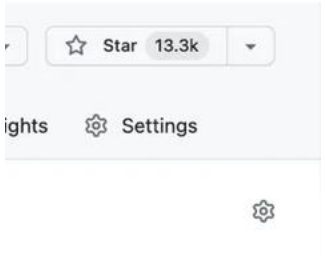
https://docs.github.com/en/get-started/quickstart/fork-a-repo

Branching

- Portion of the repo to isolate work on an issue
- Have to ‘merge’ branches later







https://docs.github.com/en/issues/tracking-your-work-with-issues/creating-a-branch-for-an-issue