
# Introduction to Agile

## What is Agile Project Management?

CE418: Systems Analysis and Design
Maryam Ramezani
Sharif University of Technology
maryam.ramezani@sharif.edu

# History

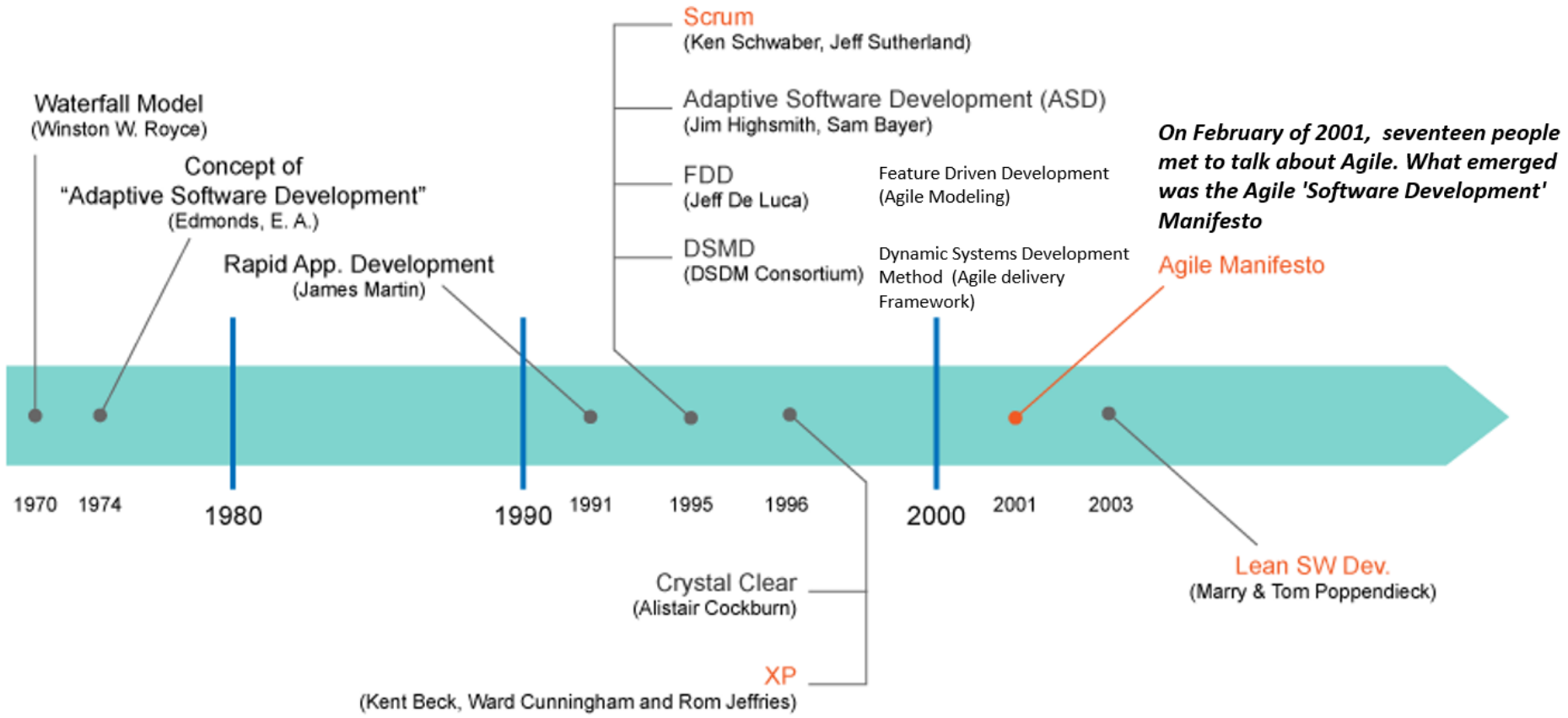
# Agile History

- In the early 1990s, software development faced a crisis. Industry experts estimated that the time between a validated business need and an actual application in production was about three years.
- Within the space of three years, requirements, systems, and even entire businesses were likely to change. That meant that many projects ended up being cancelled partway through, and many of those that were completed didn't meet all the business's current needs, even if the project's original objectives were met
- In certain industries, the lag was far greater than three years. In aerospace and defense, it could be 20 or more years before a complex system went into actual use. The Space Shuttle program, which operationally launched in 1982, used information and processing technologies from the 1960s
- In1990s, several technology leaders frustrated with these long lead times and decisions made early in a project that couldn't be changed late, began informal talks about ways to develop software more simply, without the process and documentation overhead of Waterfall and other popular software engineering techniques of the time.

# Agile Development - History

- First appeared in 1990.
- The once-common perception that agile methodologies are nothing but controlled code-&-fix approaches, with little or no sign of a clear-cut process, is only true of a small – albeit influential – minority.
- Essentially based on practices of program design, coding and testing that are believed to enhance software development flexibility and productivity.
- Most agile methodologies incorporate explicit processes, although striving to keep them as lightweight as possible.

# Agile History Timeline

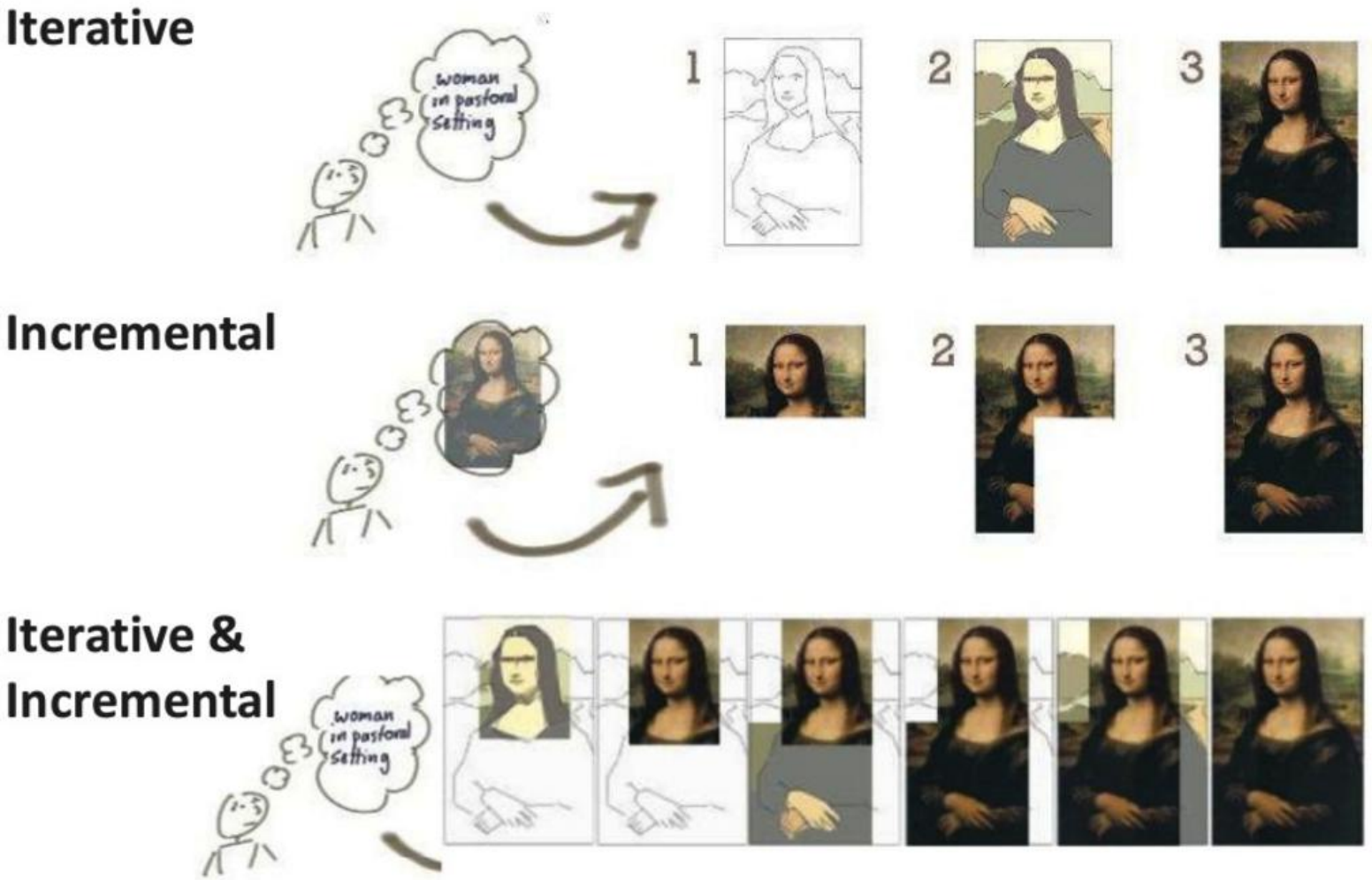



**Hirotaka Takeuchi 1986** ⟷ **1993 Jeff Sutherland, Jeff McKenna, John Scumniotales**

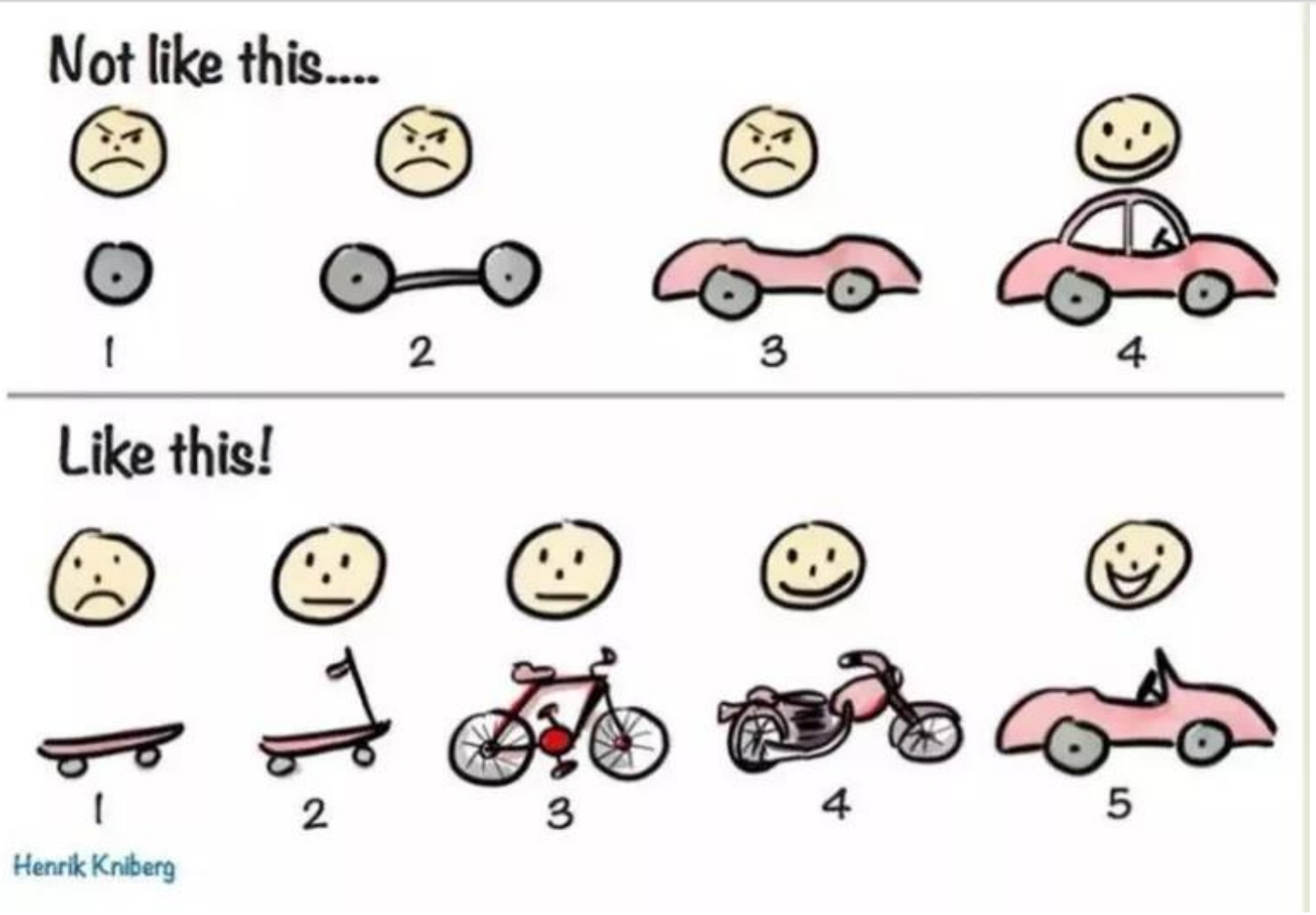
# Evolution of Project Management

| Simple Comparison | |
|---|---|
| Traditional Project Management | Modern Project Management (Agile) |
| **Elements impacting execution of a project**:<br>a) Planning<br>b) Control | **Elements impacting execution of a project**:<br>a) Competitive environment<br>b) Creativity and Innovation<br>c) Planning<br>d) Control |
| **Project Success is measured by having**:<br>a) Well defined requirements<br>b) Approved budget<br>c) On-Time Deliavery | **Project Success is measured by**:<br>a) Delivery of Business Value (what does customer want)<br>**Less important are:**<br>**a) Having well defined requirements**<br>**b) Budget constraints**<br>**c) Set Schedule**<br>Today's world has a much higher level of uncertainty which makes it difficult to always start a project with well-defined requirements |
| **Requires a Project Manager** | **Project Manager Role is distributed among**:<br>a) Product Owner<br>b) Scrum Master<br>c) Project Team |

# Agile Overview

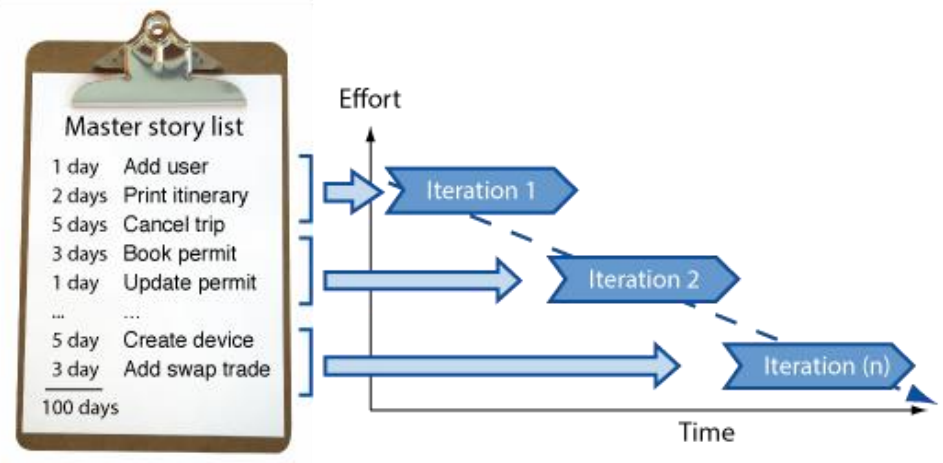


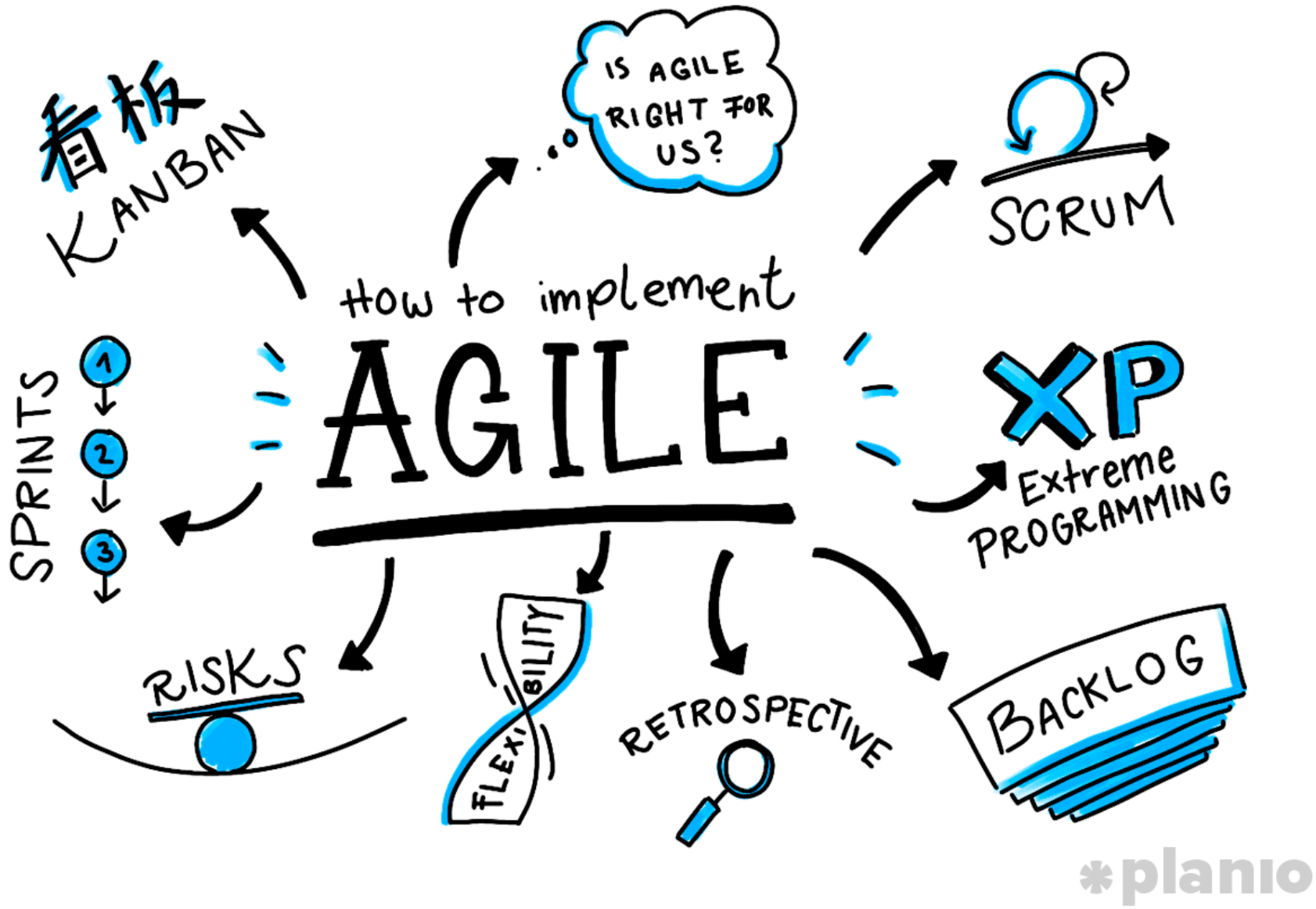
# Agile Overview



# Understanding Agile

- Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end. It works by breaking projects down into little bits of user functionality called User Stories, prioritizing them, and then continuously delivering them in short two-week cycles called Iteration.




看板
KANBAN
IS AGILE RIGHT FOR US?
SCRUM
how to implement
AGILE
SPRINTS
1
2
3
XP
Extreme PROGRAMMING
RISKS
FLEXIBILITY
RETROSPECTIVE
BACKLOG
planio

# Definition of the term “manifesto”

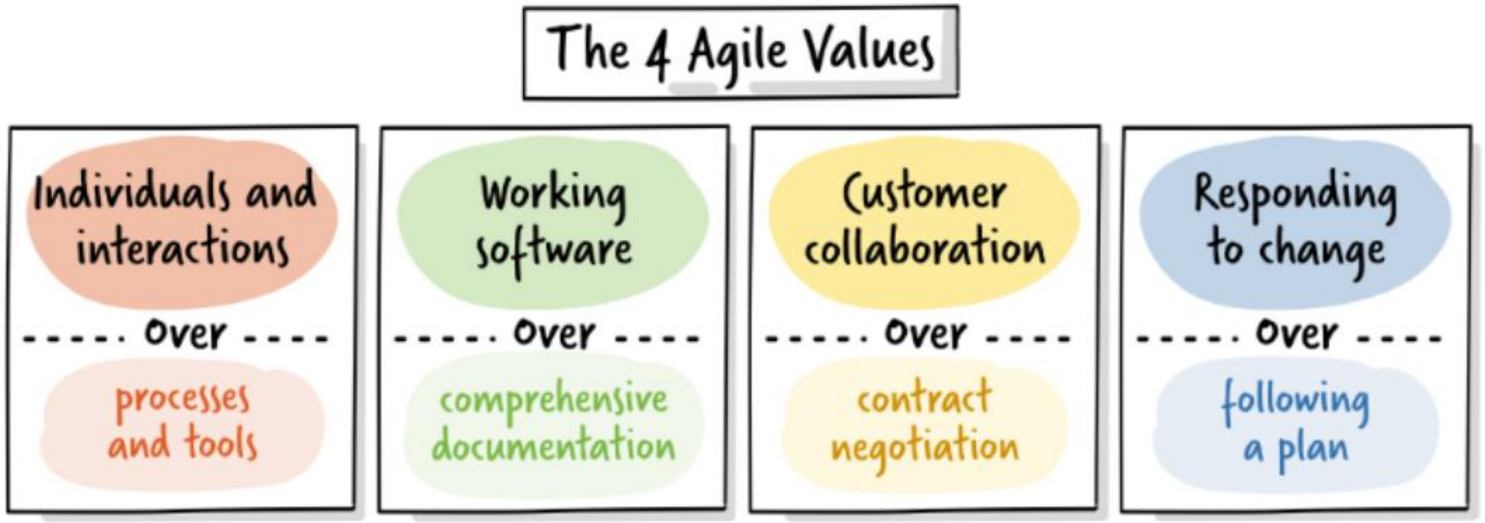
- A public declaration of principles, policies, or intentions, especially of a political nature.

# 2001:
# The Agile Manifesto is Created

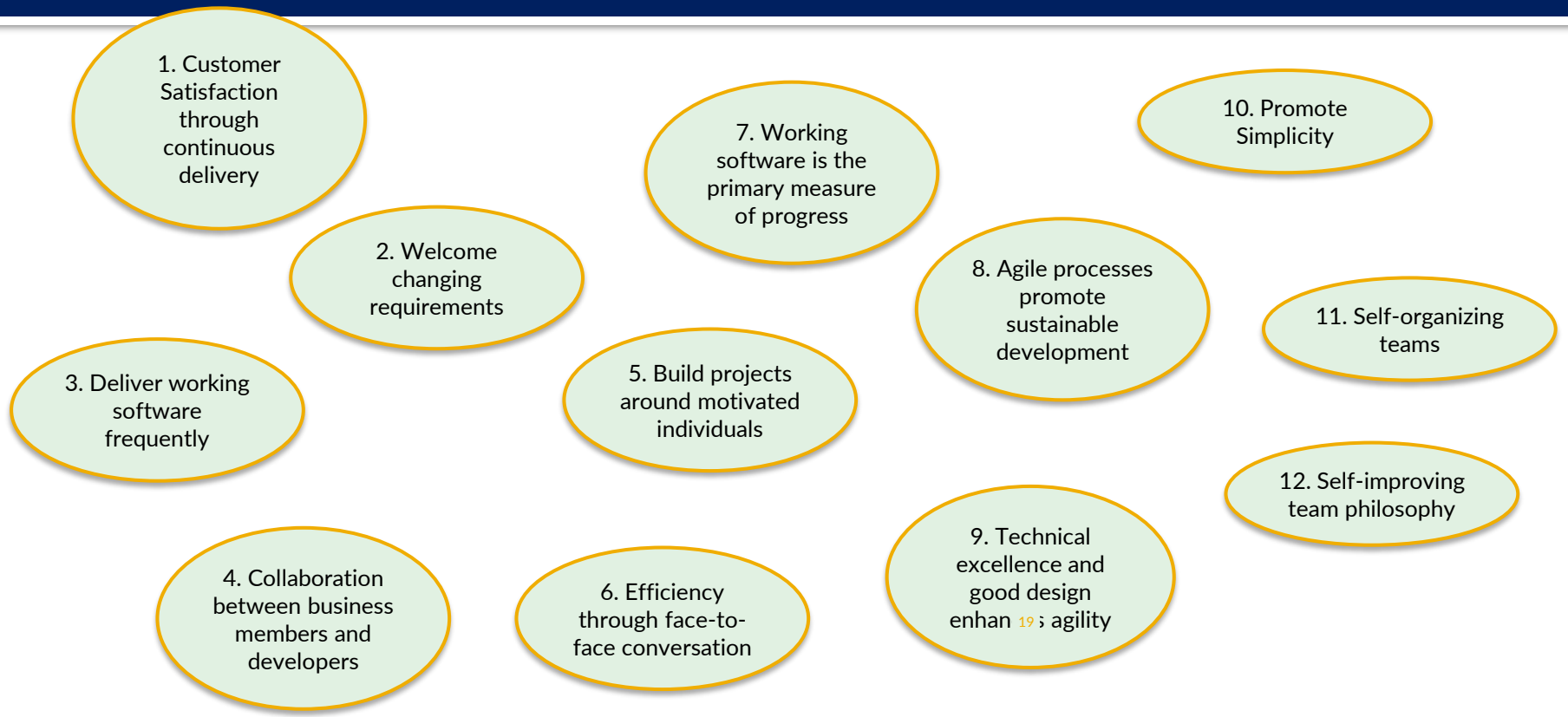
The Agile Manifesto was created and signed by 17 evangelists that observed the increasing need for an alternative to documentation-driven and heavyweight software development processes. It consists of 4 values and 12 principles and continues to be the basis for all agile software development until today.

# Four Values of The Agile Manifesto

- Individuals and Interactions Over Processes and Tools
- Working Software Over Comprehensive Documentation
- Customer Collaboration Over Contract Negotiation
- Responding to change over following a Plan



# Principles of Agile Manifesto



# Principles of 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.

# Principles of Agile Manifesto

4. 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.

# Principles of Agile Manifesto

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.

# Principles of Agile Manifesto

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.

*Source: http://agilemanifesto.org*

# Project Schedule / Cost Estimation Techniques

- Waterfall approach:

  **Bottom-Up**: Detail out all requirements and estimate each task to complete those requirements in hours/days, then use this data to develop the project schedule. In the software industry, the use of the bottom-up method has severe drawbacks due to today's speed of change. *Speed of change* means that the speed of new development tools and the speed of access to new knowledge is so great that any delay in delivery leaves one open to competitive alternatives and in danger of delivering an obsolete product
- Agile approach:

  **Top-Down**: The top-down method addresses this key issue, by using the information currently available to provide gross-level estimates. Rolling-wave planning is then used to incorporate new information as it's learned, further refining estimates and iteratively elaborating with more detail as the project progresses. This method of learning just enough to get started, with a plan to incorporate more knowledge as work outputs evolve, allows the project team to react quickly to adversity and changing market demand

# How to estimate project schedule in Agile?

- Determine **Point Value** for each item to be worked on. The most popular technique of gross level estimation is **Planning Poker**, using Fibonacci sequence to assign a point value to a feature or item
- Determine **Team Velocity**. A team's average velocity is used in forecasting a long-term schedule. Average velocity is calculated by summing and averaging the velocity measurements from the team's last three iterations
- The team's average velocity number is used to calculate the most likely scenario, while velocity numbers from the team's worst-performing iterations are used to calculate the most **pessimistic forecast** completion date. Using velocity from iterations where the team was able to complete more than expected provides the most **optimistic forecast**

# How to estimate project cost in Ag

- A simple formula is used to determine the cost per point:

  ***Σ (loaded team salaries for period n) / points completed in period n***

  A team whose total loaded salaries are $240,000 over six weeks, and completed 60 points of work in those three iterations, would have a cost per point of $4,000
- Use the following formula to determine budget:

  ***(Cost per point x total point value of items to be completed) + other expenses = forecast budget***

  ***Budget estimates are based on what we know today***

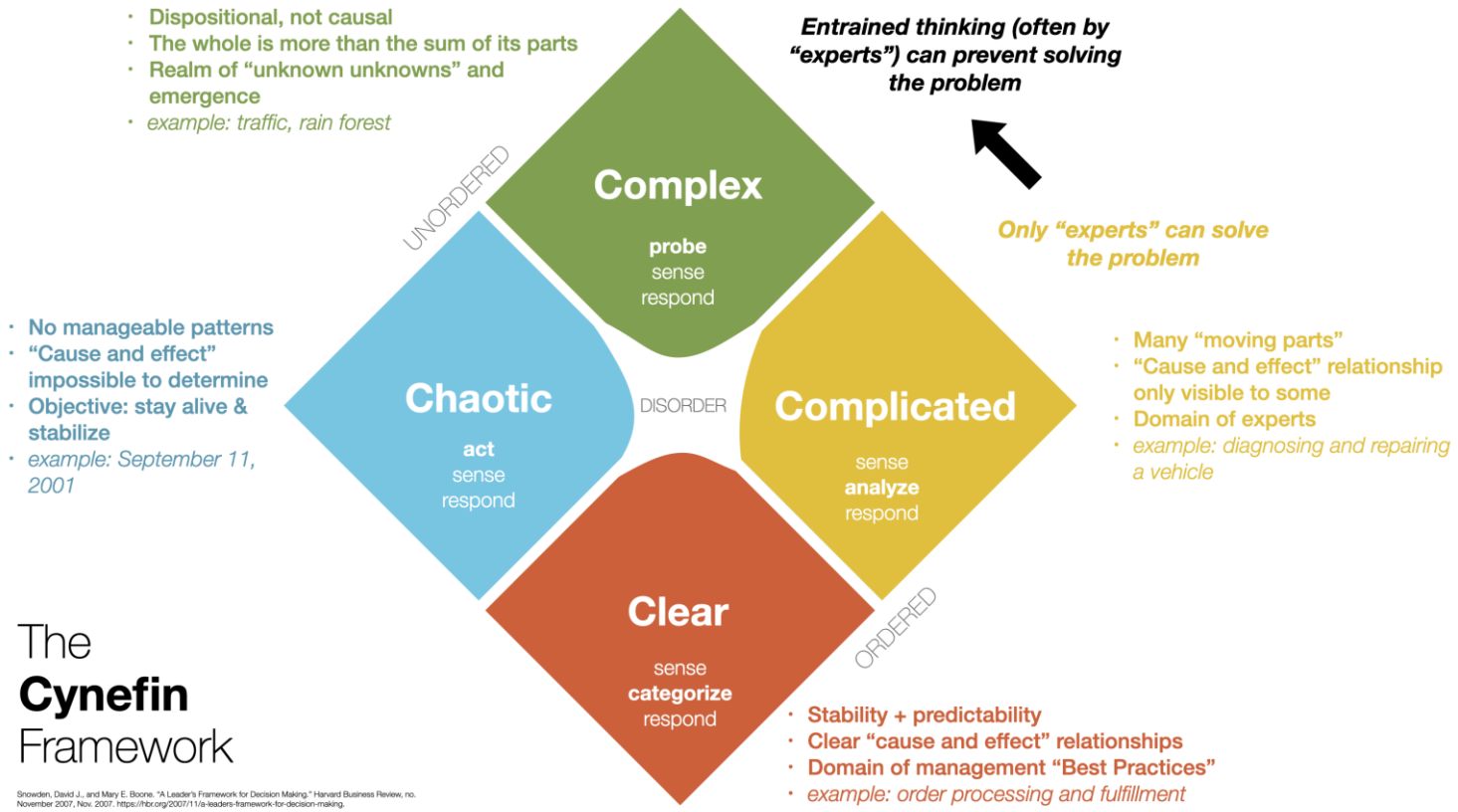
  In the example used, budget estimate is for the first release and not the entire project. The team could apply an additional 20% for the second release and an additional 5% for the last release, based on past experience

# When Agile Fails?

- System Criticality
  - Comfort (C)
  - Discretionary Money (D)
  - Essential Money (E)
  - Life (L)

# Cynefin Framework

- A decision framework



# Any Question?