

# COMP1531

## 2.7 - PM & Teamwork - Intro

**What is agile?**

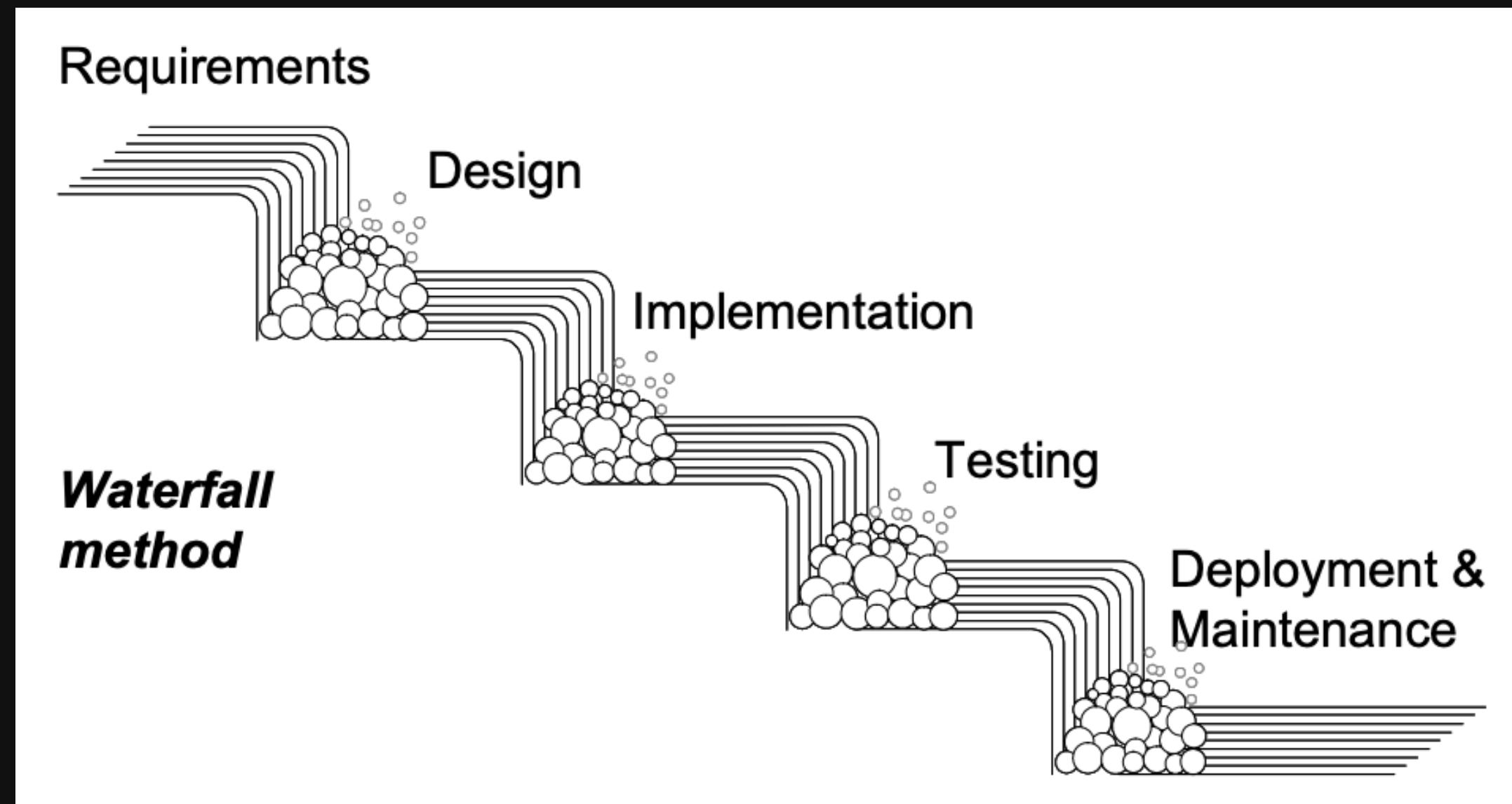
[agilemanifesto.org](https://agilemanifesto.org)

<https://agilemanifesto.org/>

# Yeah, but what is it really?

- Philosophy
- Practices
- Processes
- A cultural movement?

# A brief history lesson



# History is a lie

- "Waterfall" has never been proposed as a viable software methodology
- Reference:  
<http://www.idinews.com/waterfall.html>

# Defining features (that people usually agree on)

- Iterative and incremental
- Quick turnover
- Light on documentation

# So what is agile good for?

- Your resume?
- Changing requirements
- Delivering software on time
- Your project?



# Agile Practices

- Practices today, processes later on
- We will focus on the ones you will find most helpful in your project

# Standups

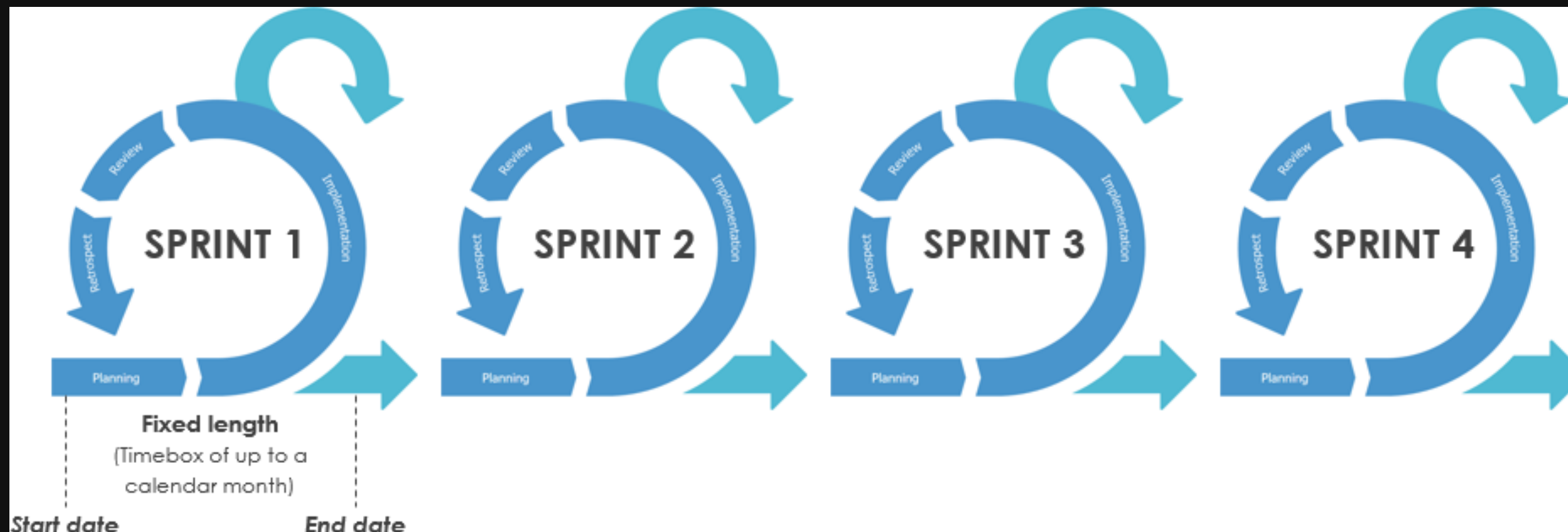
- Frequent (often daily) **short** progress update meetings
- Traditionally, everyone stands up
- Answer 3 key questions
  - What did I do?
  - What problems did I face?
  - What am I going to do?

# Asynchronous Stand-ups

- A somewhat controversial topic
- Advantages
  - No need to find a suitable time for everyone
  - May work better for big teams
- Disadvantages
  - "Blockers" take longer to be addressed
  - Easy to forget to give an update
  - Less personal
  - Updates from others can be missed

# Sprints/iterations

- Time is fixed, scope is flexible
- Plan only for the next sprint
- Typically have a release at the end of each sprint



# Task Boards



# Taskboards

- Available in GitLab
- Use them to store and track your progress on user stories
- You don't need many columns. E.g.
  - Backlog
  - Todo
  - Doing
  - Testing?
  - Closed/Done

# Pair programming

- Two programmers, one computer, one keyboard
- Take it in turns to write code, but discuss it as they go
- Can result in better code quality
- Good for helping less experienced programmers learn *micro-techniques* from more experienced programmers

# Test-Driven-Development (TDD)

- Writing tests *before* the implementation
- Write only enough code to make the next test pass
- Takes some practice
- We'll come back to this next week



# Meeting Minutes

- For more formal meetings (e.g. weekly meetings) it's quite common to take *minutes* of the meeting.
- Meeting minutes will typically consist of documenting:
  - Attendees
  - (Optional) Agenda
  - Discussion Points
  - Actions

Date	Attendees	Agenda	Notes, decisions and action items
Jan 11, 2021	@Ana @Anne @Hayden Smith	<ul style="list-style-type: none"> <li>✓ @Ana K. to have joint/trust data from Kurt so that @Hayden Smith @Anne can make decision &amp; start the plan</li> <li>✓ Update on Q2 initiative(s)</li> <li>✓ @Anne to quickly share Invest drafts</li> </ul>	<ul style="list-style-type: none"> <li>• @Hayden Smith To do a 30 minute run through the mobile site for feedback</li> <li>• @Hayden Smith Post in team-leads about multi-brokerage</li> <li>• @Hayden Smith Make portfolio/profile tickets</li> <li>• @Anne to work on profile/portfolio integration</li> <li>• @Hayden Smith to make confluence page for "template portfolios"</li> <li>• @Hayden Smith to push to prod every day</li> <li>• @Hayden Smith Reach out to DW for brokerage</li> </ul>