# Agile Software Development

## Best Practices Workshop, March 25-26 2019, Hampton VA

Carlos Cruz
Jules Kouatchou
**Brent Smith**

NASA GSFC Code 606/610 (ASTG/GMAO)
Greenbelt, Maryland 20771

A Case Study

Agile Methodology

Agile Frameworks

Errata

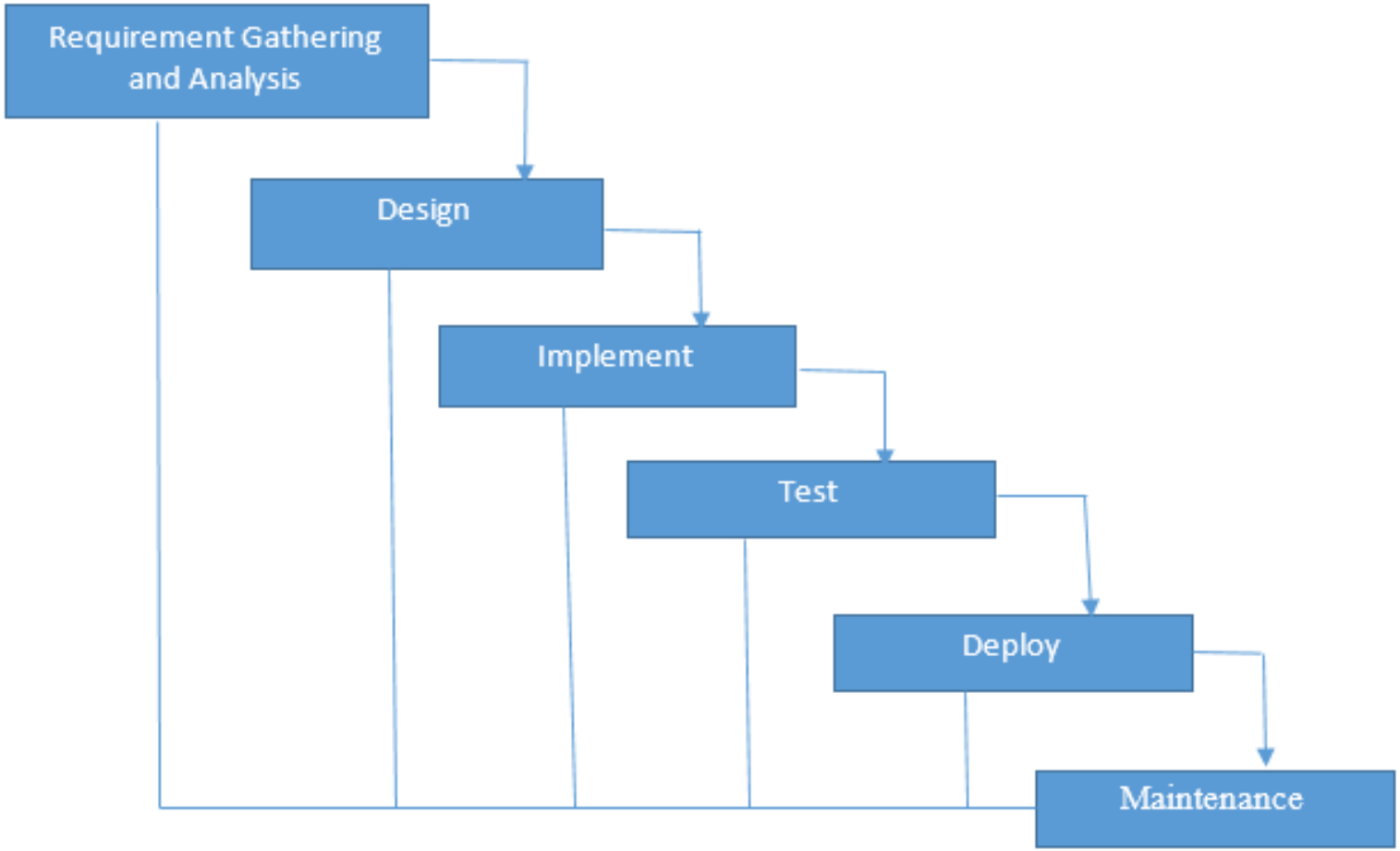

# What would you do?

Consider you or your team are given a new software development project with the expectation of a final product delivery in 1 year.

**How might you approach planning the tasks to complete this project?**


# Waterfall

The waterfall method is the traditional approach to software development.



# Waterfall

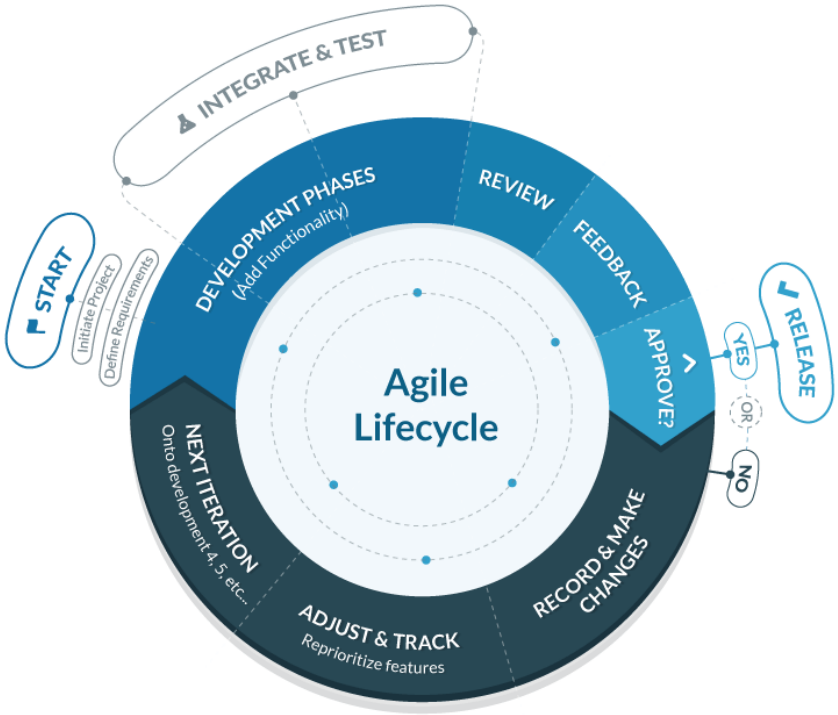
The process of the waterfall method involves the following steps/stages:

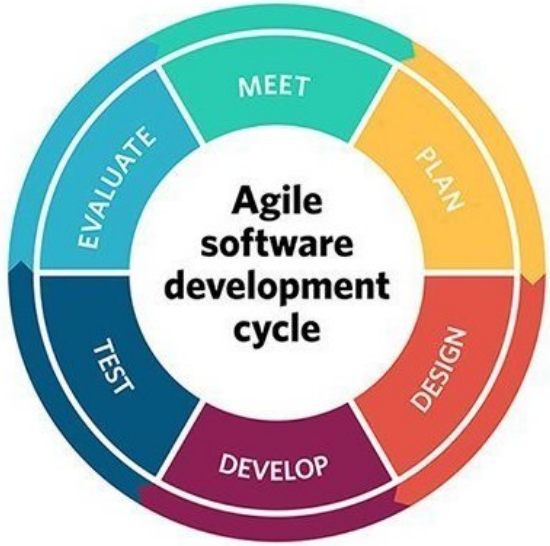
1. **Requirements Gathering and Analysis**
2. **Design**
3. **Implement**
4. **Test**
5. **Deploy**
6. **Matinenance**

Each step must be completed in sequence before proceeding to the next stage of development. This rigidity is an advantage for staying focused to complete the project, but is a disadvantage for unexpected changes.


# Agile

As a contrast, the Agile software development method breaks a project up into smaller sections (sprints) intended for a continuous cycle of development.



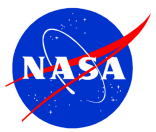



# Agile Background

Formalized in 2001 from 17 technologists, Agile came from a desire for better ways to develop software.

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

Manifesto for Agile Software Development

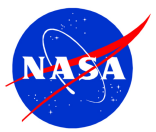



# Agile Principles

- Customer satisfaction by early and continuous delivery of valuable software.
- Welcome changing requirements, even in late development.
- Deliver working software frequently (weeks rather than months)
- Close, daily cooperation between business people and developers
- Projects are built around motivated individuals, who should be trusted
- Face-to-face conversation is the best form of communication (co-location)

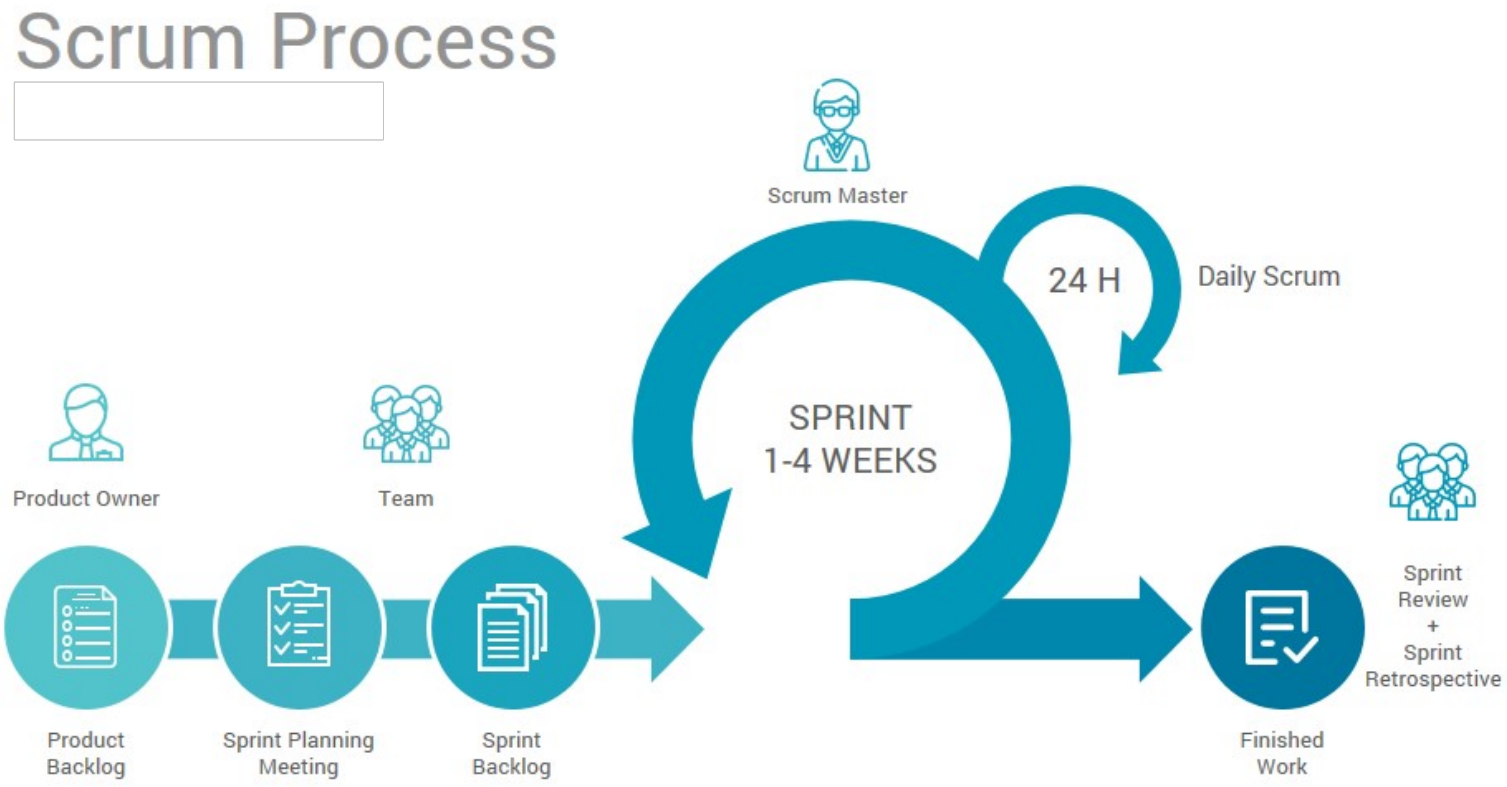

# Agile Principles (Continued)

- Working software is the primary measure of progress
- Sustainable development, able to maintain a constant pace
- Continuous attention to technical excellence and good design
- Simplicity—the art of maximizing the amount of work not done—is essential
- Best architectures, requirements, and designs emerge from self-organizing teams
- Regularly, the team reflects on how to become more effective, and adjusts accordingly

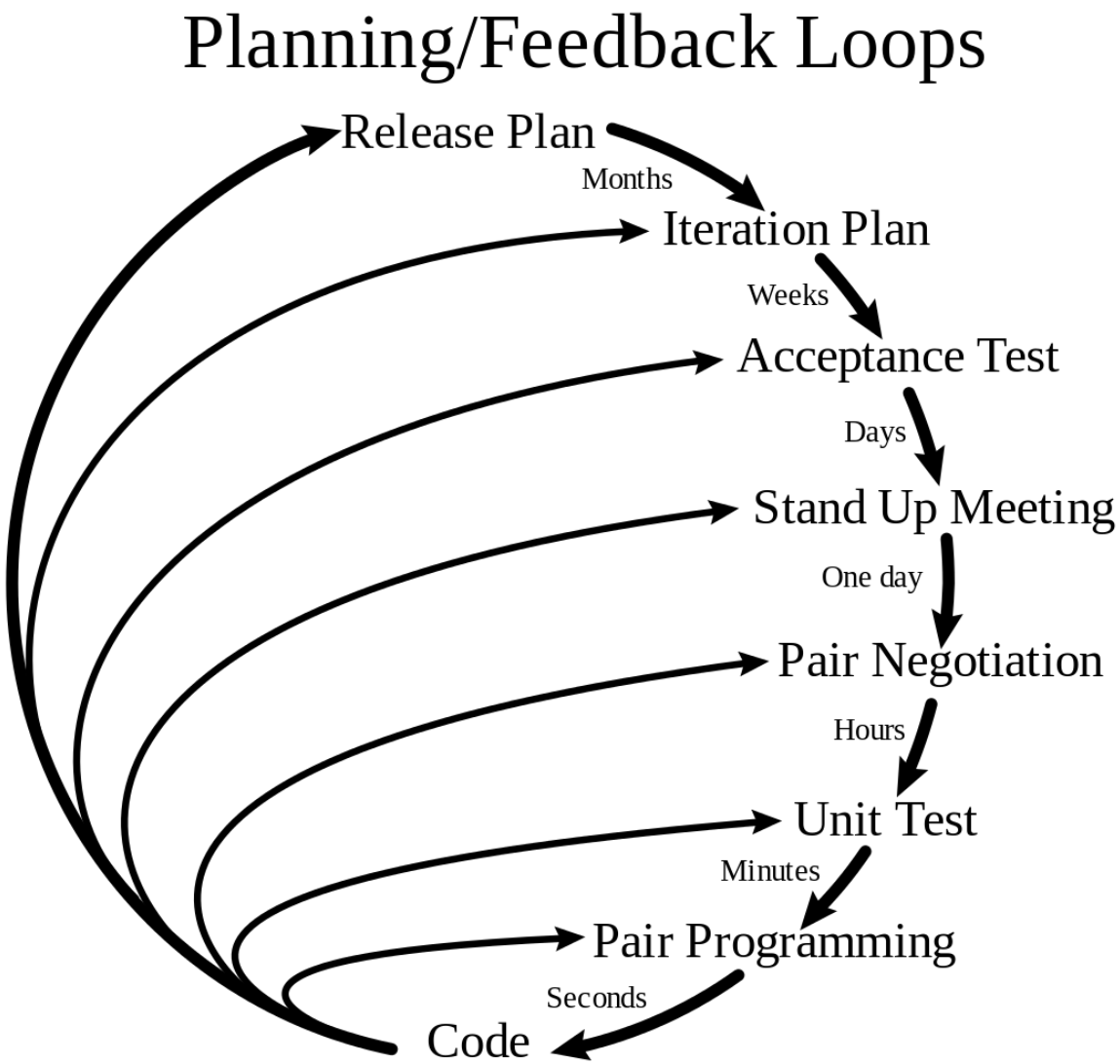



# Scrum

Scrum is based upon fixed-interval of 2-4 weekly delivery for a specific capability or feature. It is the most popular process used and involves specific roles for project management.

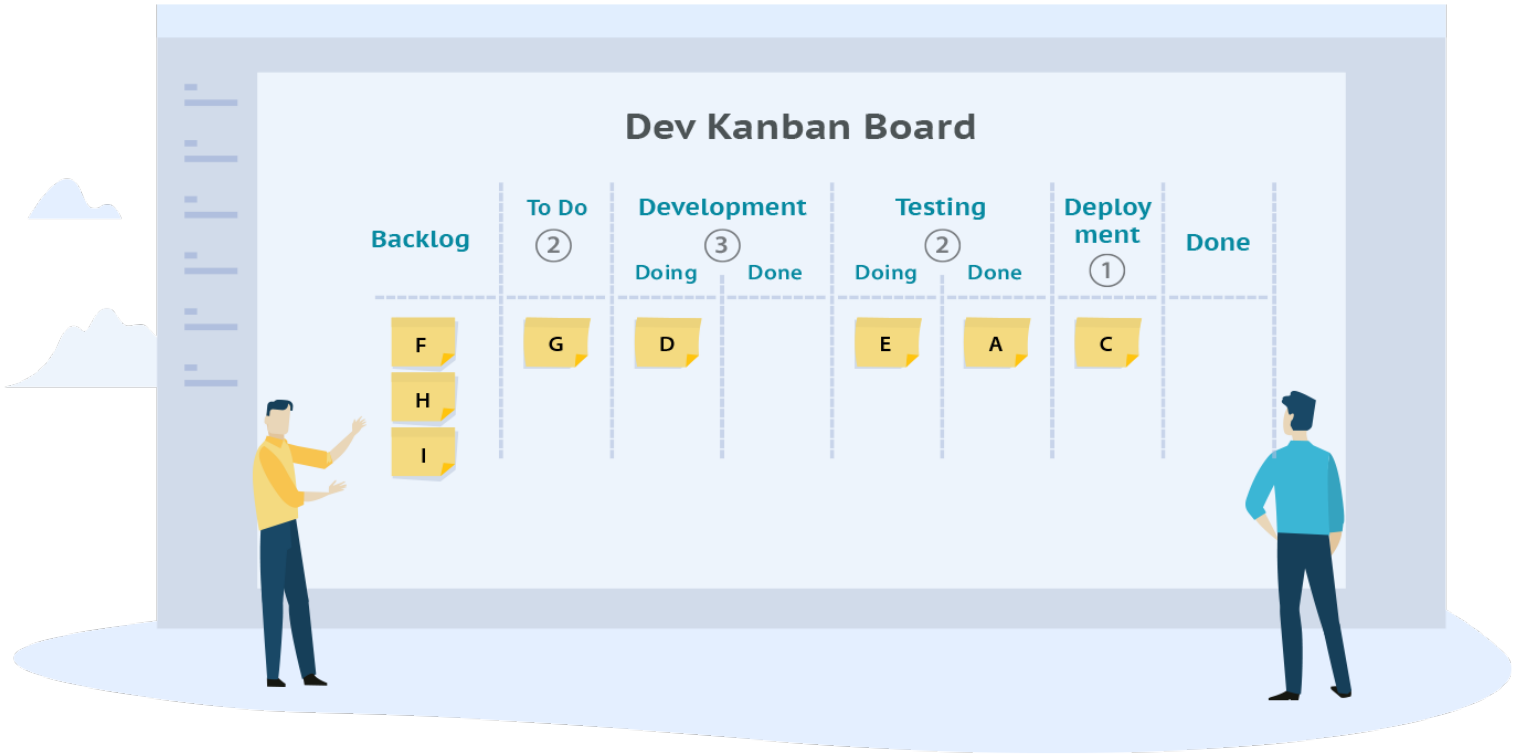


# Extreme Programming (XP)



# Kanban

Kanban is similar to Scrum, except that it focuses on continuous collaboration by using a Kanban board:



# Resources

- Manifesto for Agile Software Development
- `scrum.org`
- `scrumalliance.org`
- `extremeprogramming.org`
- `atlassian.com/agile`
  - JIRA Agile Software
- `planningpoker.com`: Uses card games to teach how about planning sprints


# Resources (Continues)

Tutorials

- https://resources.collab.net/agile-101
- https://linchpinseo.com/the-agile-method/
- https://luis-goncalves.com/what-is-agile-methodology/
- https://acodez.in/12-best-software-development-methodologies-pros-cons/


# Questions?