Intro to Process

99-520 Summer 25

Software Development

?

=

Coding

Software Development

Coding

Today Was a Good Day: The Daily Life of Software Developers

André N. Meyer, Earl T. Barr, Christian Bird, *Senior Member, IEEE*, and Thomas Zimmermann, *Senior Member, IEEE*

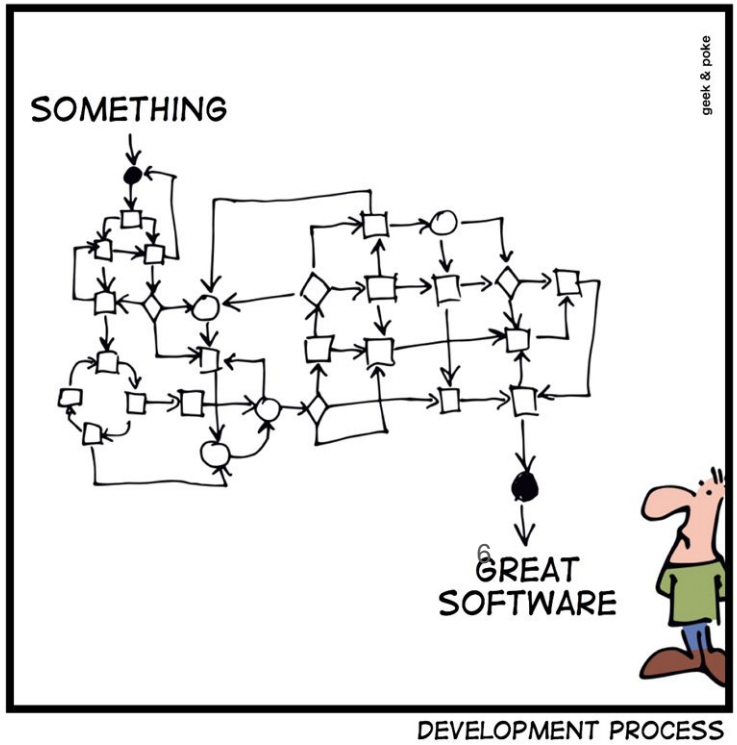
"developers spend surprisingly little time with coding, 9 to 61 percent depending on the study"

Outline

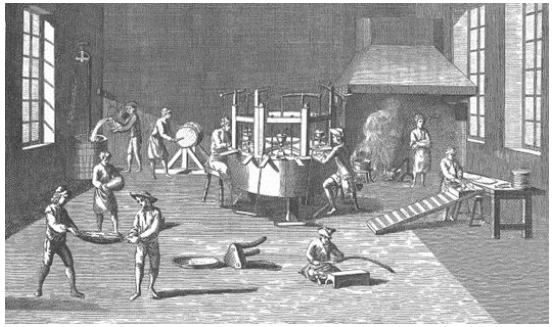
- **Software Processes and why we need them**
- **Software Process Models**
  - **Agile and Scrum**
- Challenges of distributed and remote teams

Software development methods

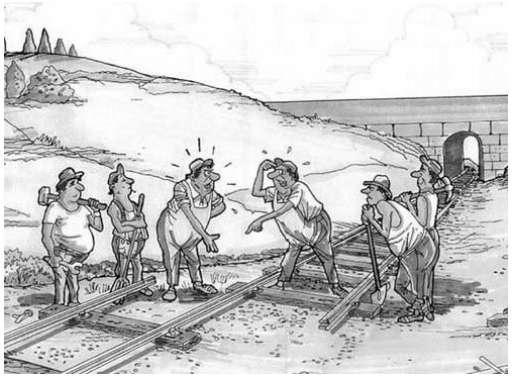
Software development methods are codified practices that support the work of a group and are based on an idea of how is best to work to achieve a particular business goal



Successful projects require that their members:



… divide the work …

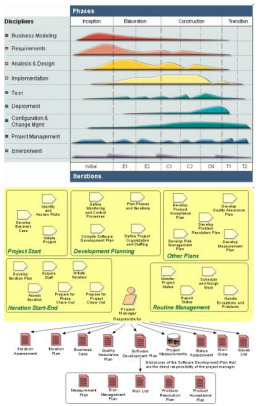


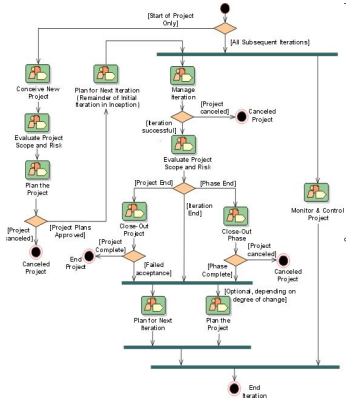
… coordinate their activities and resources …

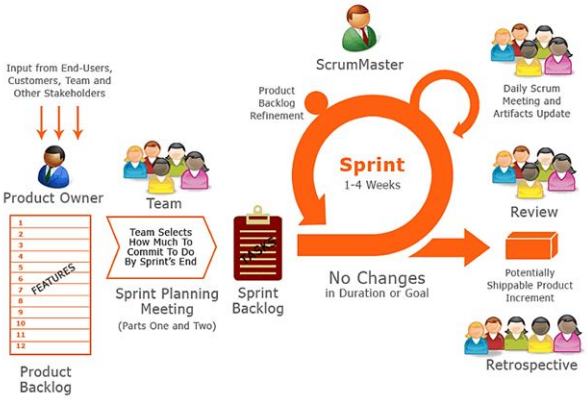
… achieve a shared understanding of the work to be done …

… and align their efforts

Development methods differ among many dimensions, a non-exhaustive list







	Rational Unified Process (RUP) – Plan oriented	SCRUM - Agile
What it values	The software will be delivered as promised, on time and on budget	We'll get you something working faster
Scope	Work management and technical practices. End-to-End	Mostly work management. Currently focused on construction
Knowledge management	Knowledge is made explicit through extensive documentation	Knowledge is mostly tacit
Human resource management	Specialized workforce Team members join as required	Cross-functional workforce Team members are assigned to the project at its outset, they are collocated and dedicated full time
Coordination	Reference to higher authority, preparation	Mutual adjustment
Work control	Project manager (bureaucratic control)	self-managed (clan control)

Breakout: Process Trade-Off Scenarios

- Work in groups
- Scenario
  1. A medical record system for a hospital.
  2. Social media app for students.
  3. A one-week hackathon project.

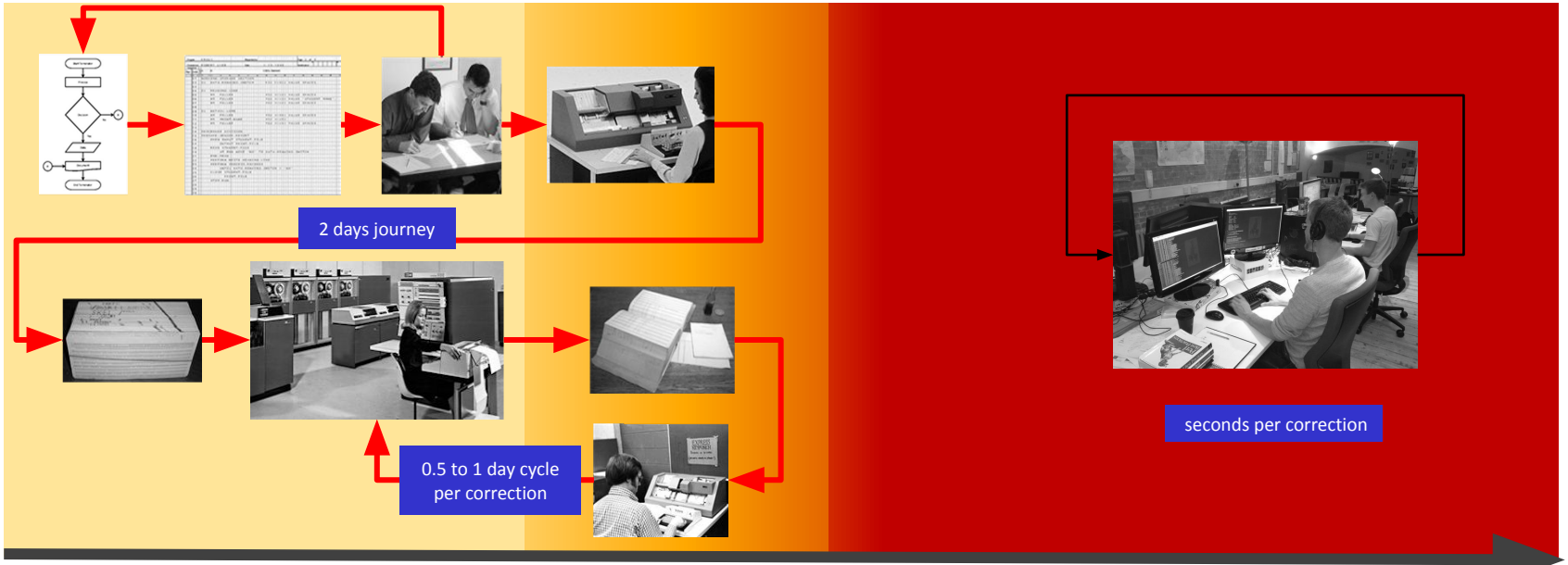
Answer on Slack (#lectures)

- Would you use a more plan-driven or agile (iterative) approach for this project? Why?
- What are some trade-offs or risks of the approach you picked?

The evolution of software development methods: A hardware perspective

Focus on resource utilization

Focus on time to market



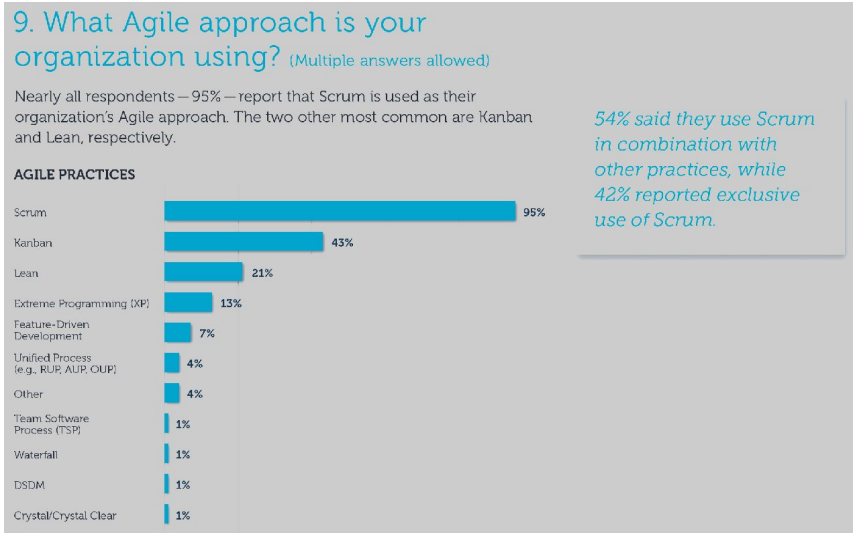
0.0005 | 0.0012 | 15 | 3,400 | 330,000 | 750,000

MIPS per 1,000$

1968	1970	1984	1986	1996	2001	2010+
NATO conference coins the term Software Engineering	Royce describes (& criticizes) the "waterfall" model of software development	SEI is instituted by DoD Maturity Models	Takeuchi & Nonaka. The New Product Development Game	Xp is born out of the C3 project at Chrysler	Agile Manifesto	DAD, Safe, DevOps

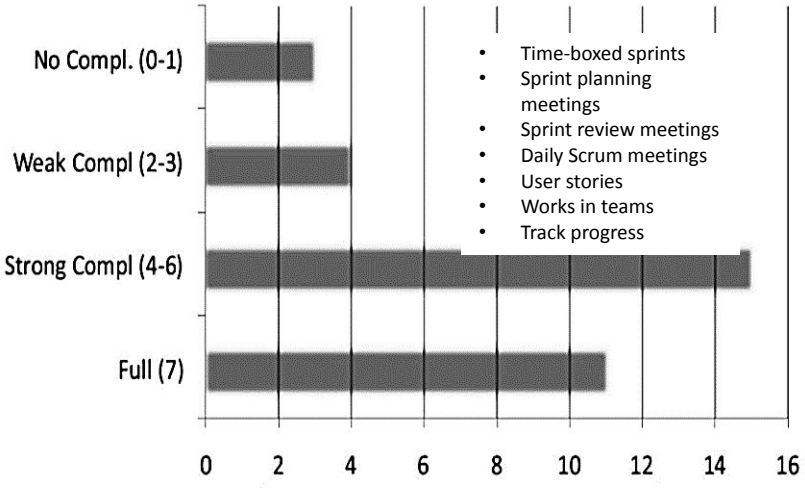
Methods, whether agile or plan based, are seldom used as prescribed

54% said they use Scrum in combination with other practices …



The 2015 State of Scrum Report, Scrum Alliance, 2015

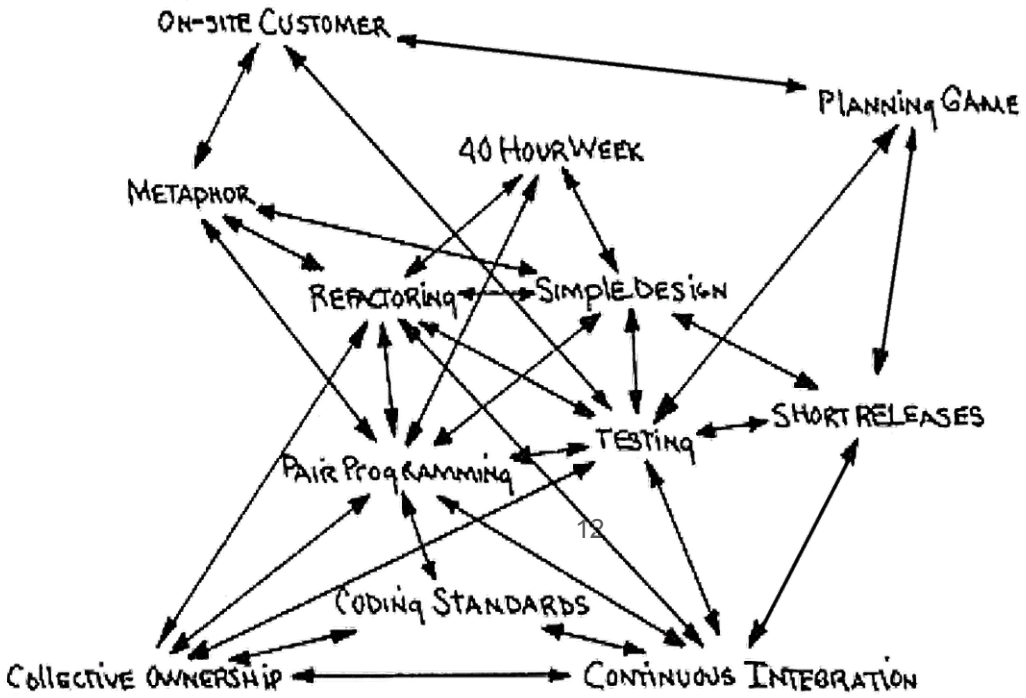
Only 33% of respondents said they used all 7 Scrum practices



Agile software development practice adoption survey, Kurapati, N. et al, 2012

If we are going to change any of a method's practices, we need to understand the role they play and the interactions between them

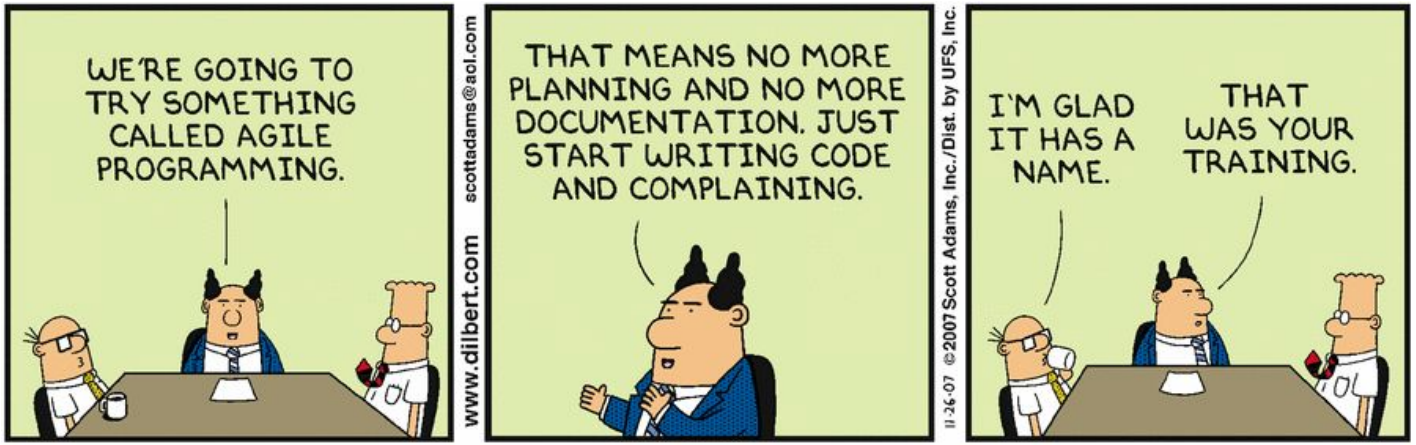
"Any one practice doesn't stand well on its own. They require the other practices to keep them in balance. Figure 4 is a diagram that summarizes the practices. A line between two practices means that the two practices reinforce each other. I didn't want to present this picture first, because it makes XP look complicated. The individual pieces are simple. The richness comes from the interactions of the parts"



Method's practices support each other

Extreme Programming Explained, K. Beck, 1999

Agile Methods



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

Twelve Principles of Agile Software

12 Agile Principles

@OlgaHeismann

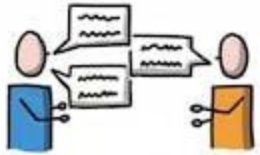
Satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development.

Deliver working software frequently.

Business people and developers must work together.

Build projects around motivated individuals. Give them the support they need. Trust them.



The most efficient and effective method of conveying information is face-to-face conversation.

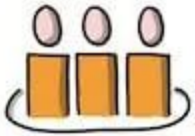
Working software is the primary measure of progress.

The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design.

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

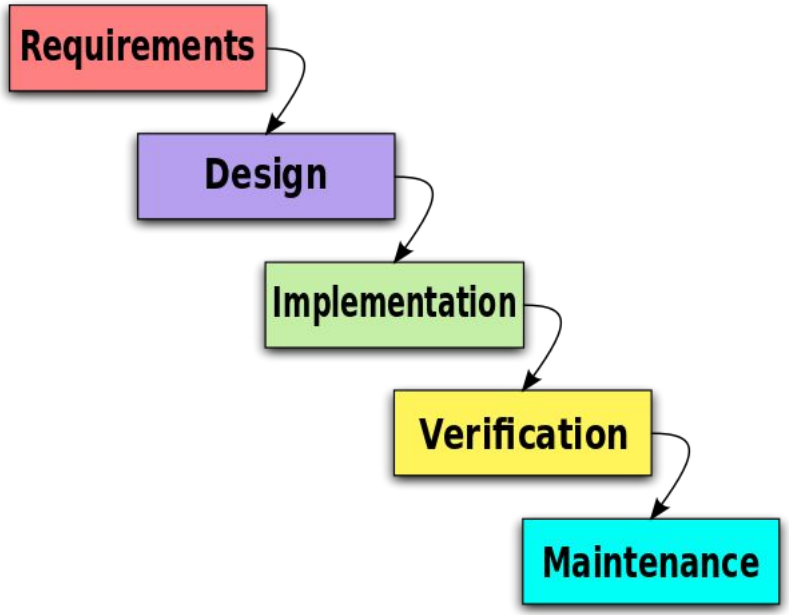
The best architectures, requirements, and designs emerge from self-organizing teams.



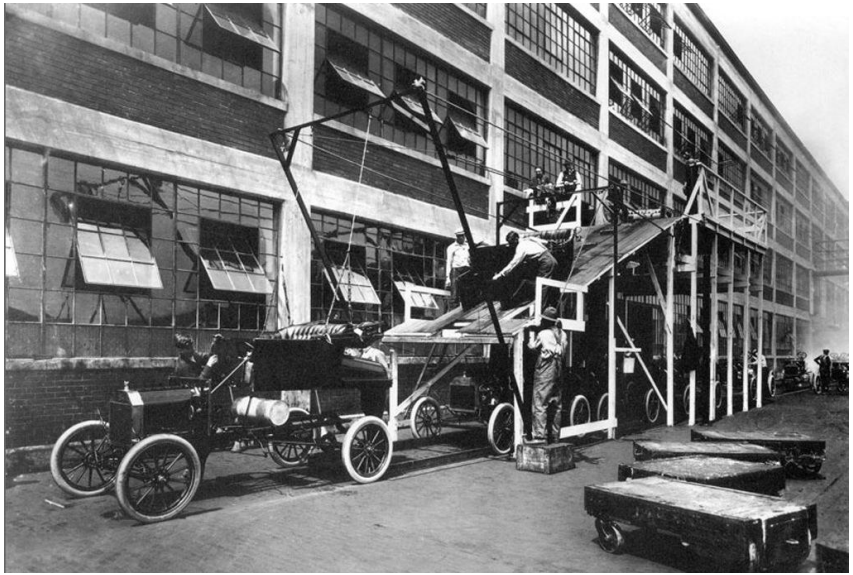
The team reflects on how to become more effective and adjusts its behavior accordingly.

Source: @OlgaHeismann

Waterfall model was the original software process



… akin to processes pioneered in mass manufacturing (e.g., by Ford)



Toyota Production System (aka lean production)

Taiichi Ohno

Concepts:

- Just-in-time – meaning *"Making only what is needed, only when it is needed, and only in the amount that is needed"*
- Jidoka – (Autonomation) meaning "Automation with a human touch"

Principles:

- Continuous Improvement
- Respect for People
- The right process will produce the right results
- Add value to the organization by developing your people and partners
- Continuously solving root problems drives organizational learning

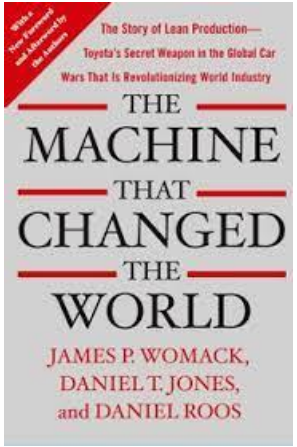
From TPS to Agile

The new new product development game

Stop running the relay race and take up rugby

Hirotaka Takeuchi and Ikujiro Nonaka

1986



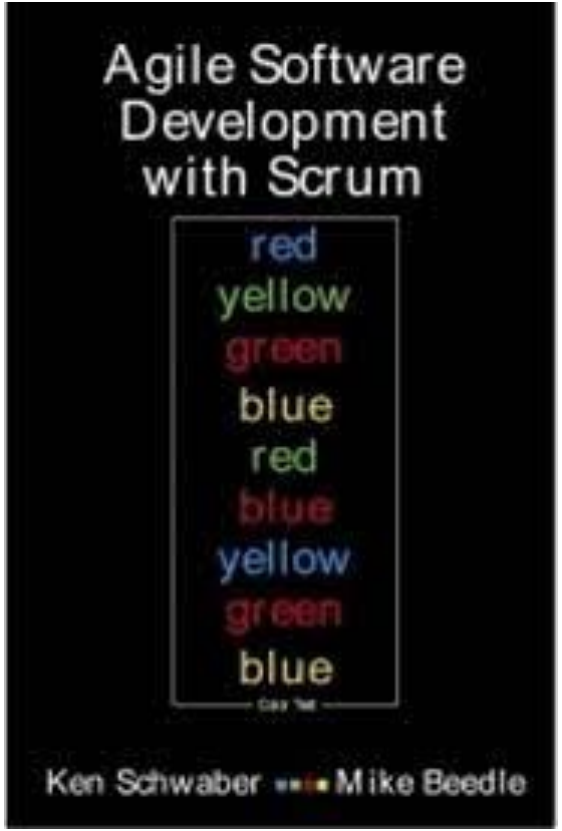
1990

...

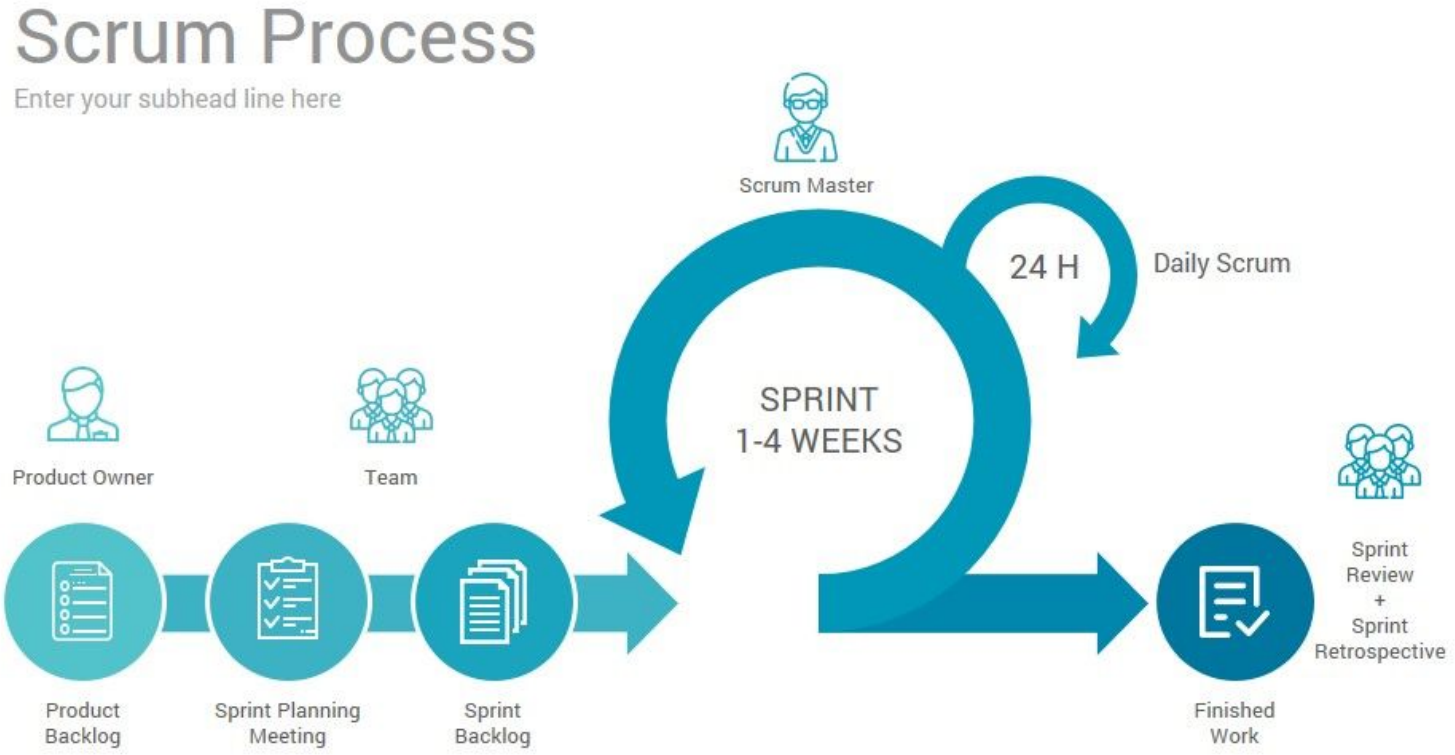
2001

Scrum

(Only a brief intro)



Elements of Scrum



Backlogs

The **product backlog** is all the features for the product

The **sprint backlog** is all the features that will be worked on for that sprint.

These should be broken down into discrete tasks:

- Fine-grained
- Estimated
- Assigned to individual team members
- Acceptance criteria should be defined

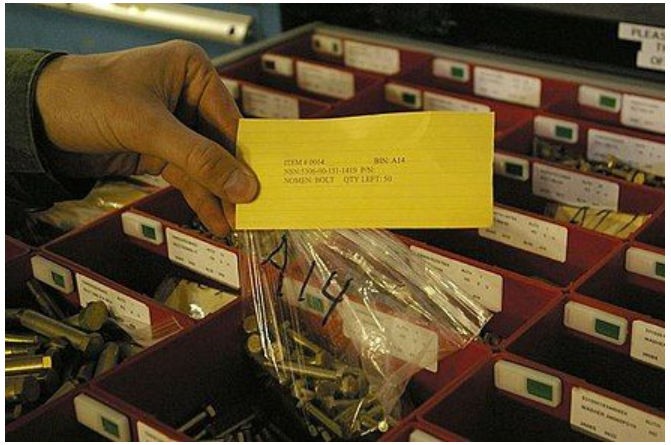
User Stories are often used

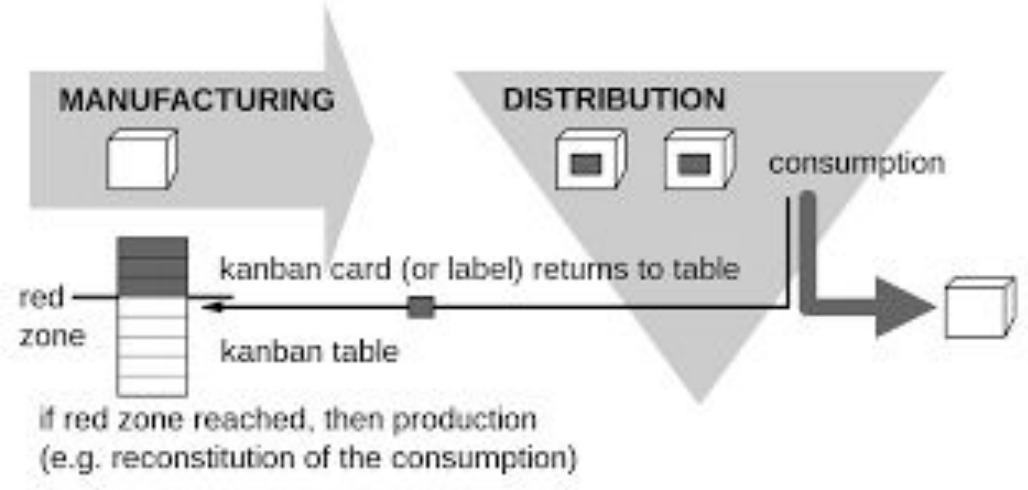
Kanban Board

Information Radiator

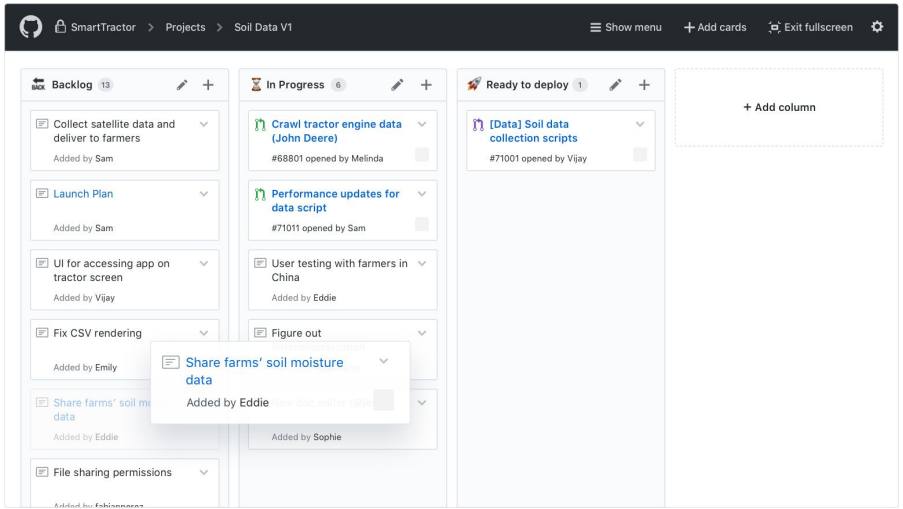
Team + Instructors

Just in time production: Only do the work that needs to be done, when it needs to be done

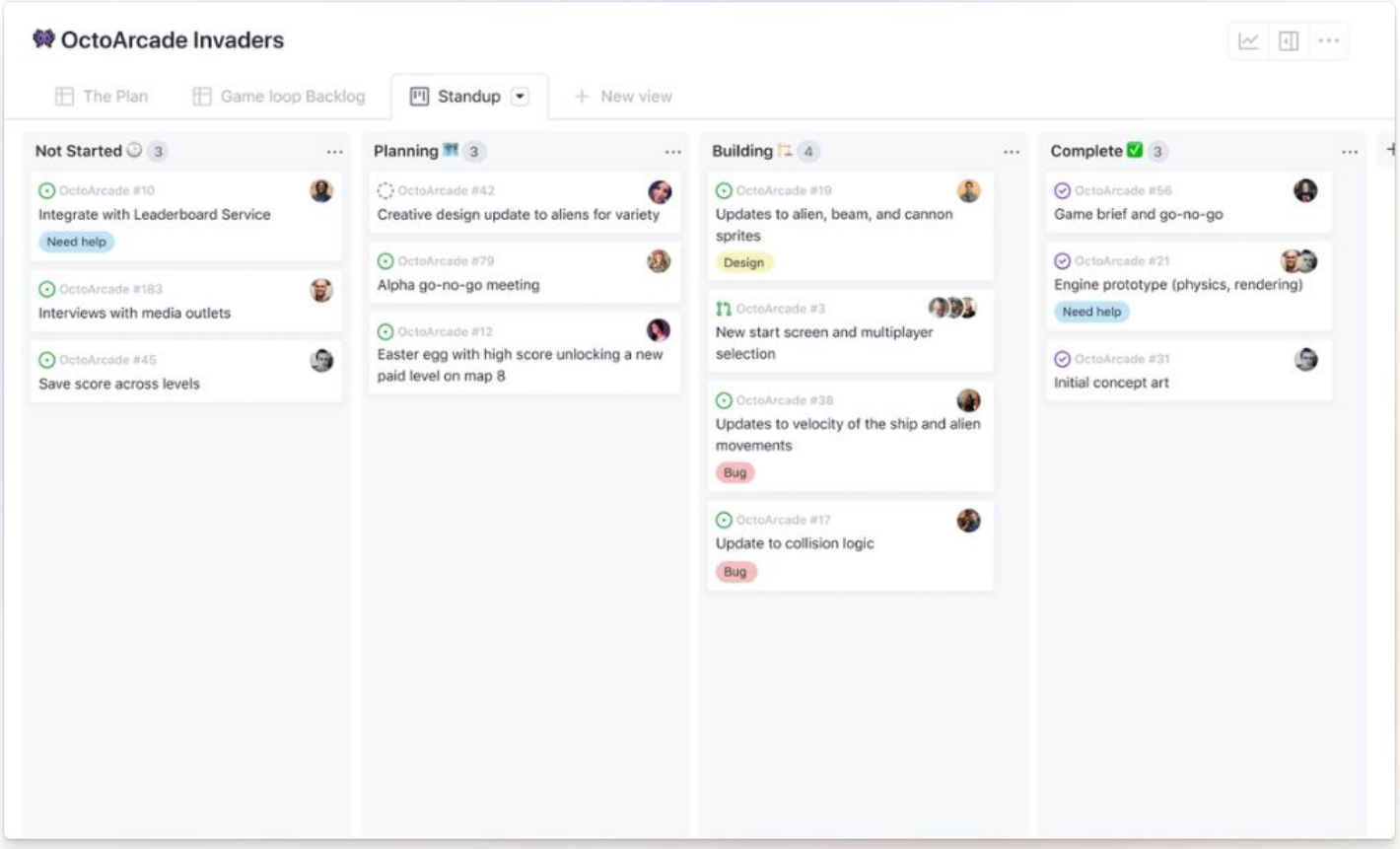




Kanban boards



Kanban Board



Scrum Meetings

- Sprint Planning Meeting
  - Entire Team decides together what to tackle for that sprint
- Daily Scrum Meeting
  - Quick Meeting to touch base on :
    - What have I done? What am I doing next? What am I stuck on/need help?
- Sprint Retrospective
  - Review sprint process
- Sprint Review Meeting
  - Review Product

In this course:

Continuous Improvement

- Retrospectives, Reflective Presentations

Respect for People

- Team Contract, Feedback guidelines, organized meetings, code review

The right process will produce the right results

- Kanban, JIT, CI/CD, QA (testing and beyond)

Add value to the organization by developing your people and partners

- Team members are multi-skilled and understand the process, role rotations

Continuously solving root problems drives organizational learning

- Standup meetings with Faculty and mentors.

Process we recommend (with Client approval)

Brainstorm user stories

Create product backlog

Select sprint backlog

Prioritize and assign users stories for sprint

Work for sprint

Implement user story in branch

Open PR for team to review

Code Review open PRs

SCRUM ARTIFACTS

https://www.atlassian.com/agile/scrum/artifacts

Next: Agile in Remote and Distributed Teams