



AGILE, SCRUM, STAND-UP, USER STORIES, WATERFALL

ACTIVITY ~10 min

SMALL GROUPS

Write user stories

- Break into groups of three (based on who you're nearest).
- (5 mins.) Brainstorm 5 user stories for WebMD using this formula:

As a _____, I want to _____, so that _____.

• (5 mins.) Regroup, share, and discuss.

Designers and developers will be able to:

- Define a user story and write user stories.
- Describe the difference between a user story and formal requirements.
- Identify the elements of the product development cycle, and the two primary kinds.
- Discuss the different ways design is implemented into the dev cycle, and the pros and cons of each.
- Describe the Agile Manifesto and why it matters.
- Identify key roles in a Scrum team.

- User stories vs. user requirements
- → The product development cycle
- What is Scrum?
- Scrum roles
- Taking stock/Q&A

BUILDING BLOCKS

The design team; the larger work team

FUNDAMENTALS

Dev cycles and Scrum

DEEP DIVE

Other Agile approaches; backlogs, burn-down charts, and Trello; sizing



USER STORIES VS. USER REQUIREMENTS

USER STORIES VS. USER REQUIREMENTS

Remember user requirements?

- The "what" in "who, where, when, what, why, and how"
- Typically classed in User Requirement Docs (URDs or BRDs) using MODE categories.
 - Mandatory (product fails without this)
 - Desirable (something nice for the user, to be made if time and money allow)
 - Optional (typically a generic feature that could work throughout a product, e.g., "print to pdf" button)
 - Enhancement (a better way of doing a feature; often deferred to future work)

USER STORIES VS. USER REQUIREMENTS

User requirements finish the conversation

Fully defined user requirements and URDs (user requirement documents) may or may not happen in a project, depending on its size, the company culture, and the dev cycle.

USER STORIES VS. USER REQUIREMENTS

User stories start the conversation

A user story is a narrative description of a product feature that solves a problem.

As a _____, I want to _____, so that _____

- ▶ How big is this story?
- When does this happen?
- What's the priority?
- → How does this happen?

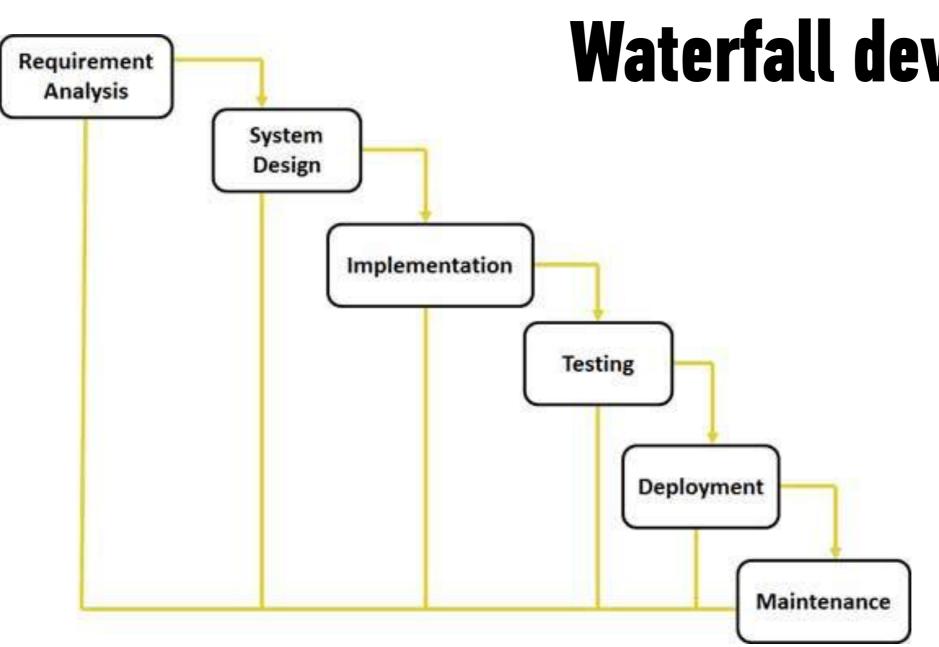
- ▶ What are the tasks involved?
- ▶ Who implements this?
- Does it need a special design? Or can we use the style guide?

WHAT IS A PRODUCT DEVELOPMENT CYCLE?

Creating a new product

Product development has four phases, which may overlap somewhat:

- 1. **Fuzzy front-end (FFE)**. Figuring out what you want to do. Includes research.
- 2. **Product design**. Creating use cases and defining requirements (this is where URDs are born); designing how to meet those needs.
- 3. **Product implementation**. This is when the product is developed.
- 4. **Commercialization**. Production and market launch. Some people call this fuzzy back-end.



Waterfall development cycle

WHAT IS SCRUM?

SO, WHAT IS AGILE?

"Agile" refers to any development approach which follows the Agile Manifesto.

The Agile Manifesto

AGILE MANIFESTO

The Agile Manifesto was written in 2001 by seventeen developers*, whom we may assume were tired of interminable projects with distant release dates, inability to change priorities, inconsistent team interaction, and long hours.

In practice, Agile development consists of short iterations by cross-functional teams creating releasable code each iteration. Overall goals can change from iteration to iteration, allowing dev goals to change in light of customer feedback, testing, and market changes. It also allows the formation of strong teams, with individuals taking ownership of specific tasks, and the team taking ownership of each iteration's work.

*Two of those developers had already invented Scrum.

OVERVIEW, PRODUCT DEVELOPMENT CYCLE

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

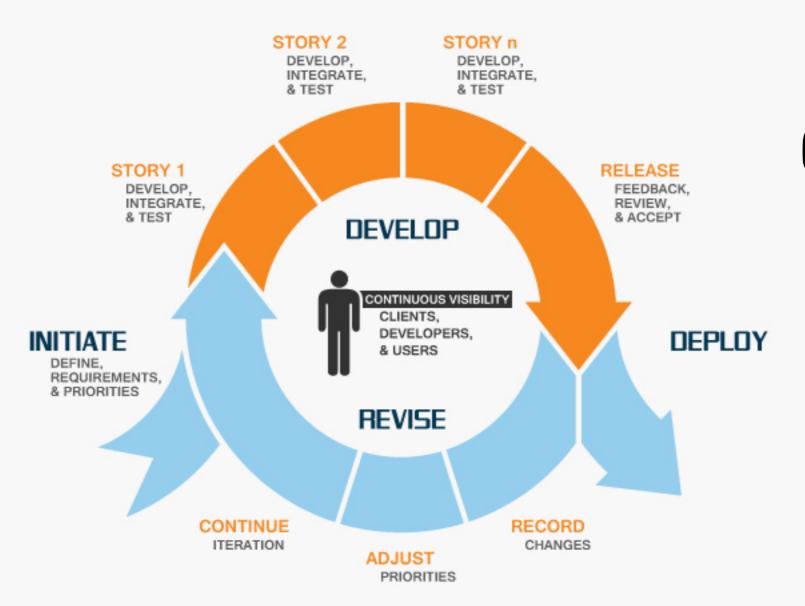
Individuals & interactions over processes and tools

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.



Agile development cycle

ACTIVITY ~5-10 min

How does this affect you?



As a class, let's capture ways Agile vs. waterfall affect design and development.

MHAT IS SCRUM?

WHAT IS SCRUM?

OK, SO THAT'S AGILE. WHAT'S SCRUM?

Scrum is an approach developed in the 1980s which strongly overlaps the requirements of the Agile Manifesto.

1980S: "WE'RE LOSING THE RELAY RACE"

"The... 'relay race' approach to product development...may conflict with the goals of maximum speed and flexibility. Instead a holistic or 'rugby' approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today's competitive requirements."

Hirotaka Takeuchi and Ikujiro Nonaka, "The New New Product Development Game", *Harvard Business Review*, January 1986.

Scrum in 100 words

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.

SCRUM HAS BEEN USED BY:

Microsoft Intuit

Yahoo Nielsen Media

Google First American Real Estate

Electronic Arts BMC Software

High Moon Studios Ipswitch

Lockheed Martin John Deere

Philips Lexis Nexis

Siemens Sabre

Nokia Salesforce.com

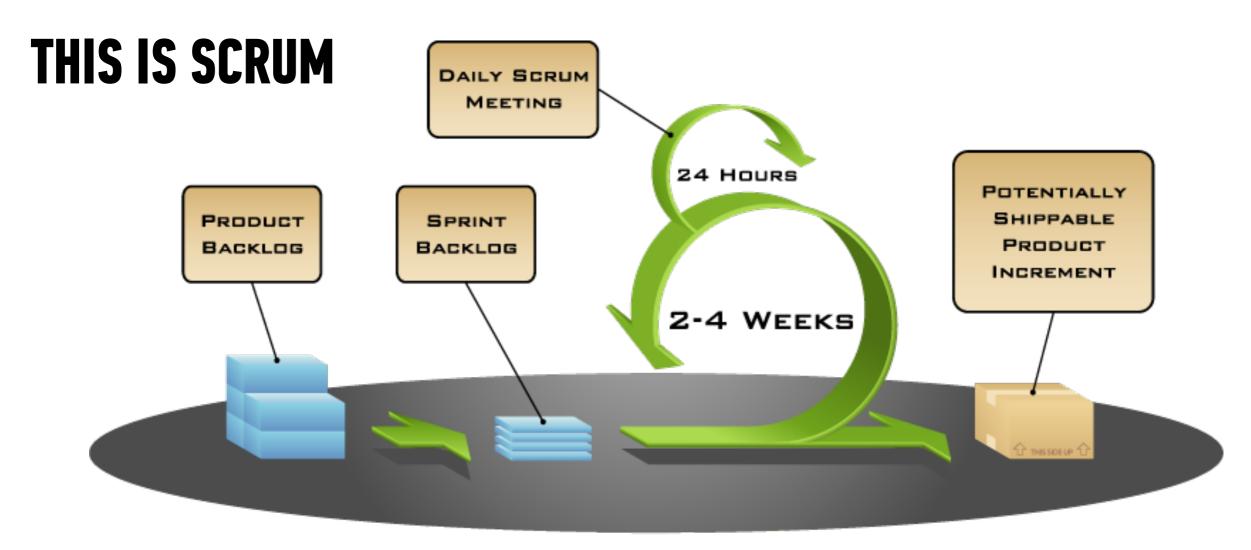
Capital One Time Warner

BBC Turner Broadcasting

Intuit Dell

CHARACTERISTICS

- Self-organizing teams
- Product progresses in a series of "sprints"
- Requirements are captured as items in a list of "product backlog"
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- Agile



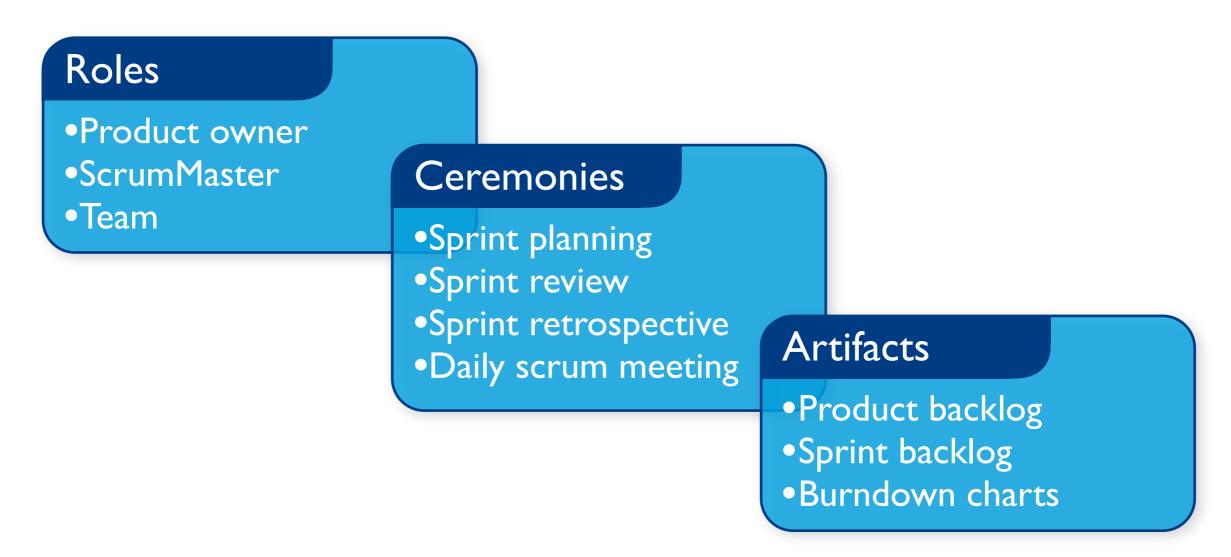
COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

SEQUENTIAL VS. OVERLAPPING DEVELOPMENT

Requirements Design Code Test Rather than doing all of one thing at a time... ...Scrum teams do a little of everything all the time

Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.

SCRUM FRAMEWORK





WHO DOES WHAT?

Roles

- Product owner
- ScrumMaster
- Team

PRODUCT OWNER

- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accepts or rejects work results

The product owner typically writes the user stories (see following slides). Good product owners write brief, highly focused stories that are easily evaluated and sized.

THE SCRUM MASTER

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

Actually, IRL the PO and ScrumMaster may overlap roles or be the same person. The PO may be better placed to removed impediments and may work closely with the ScrumMaster to protect the team.

THE TEAM

- Typically 5-9 people
- Cross-functional:
 - Programmers, testers, user experience designers, etc.
- Members should be full-time team members
 - May be exceptions (e.g., database administrator)

THE SINGLE, WRINGABLE NECK

The product owner takes responsibility for the entire outcome of the sprint. They are the "single, wringable neck" from management's perspective.

The Scrum team takes collective responsibility for finishing

their accepted stories.

But wait, that seems to conflict. How does that work?

- The Scrum team accepts work and gets to decide *how* the work is done.
- The Product Owner defines *what* work is done (goals and stories) and gets to decide *if* the work is done. They accept stories and mark them as done.

LESSON TITLE

TAKING STOCK

TAKING STOCK

WE LEARNED

- How to recognize and write user stories.
- The difference between a user story and formal requirements.
- Identify the elements of the product development cycle.
- Understand waterfall vs. Agile/Scrum.
- Describe the different roles of product owner, scrum master, and team.

YOU DID

- Wrote Scrum user stories!
- Discussed the relevance of Agile and waterfall IRL.

TAKING STOCK

COMING UP

- We're about to dive into user story best practices, backlogs, burn-down charts, and tracking the process.
- We'll discuss alternatives to Scrum in the Agile world.
- You'll apply the processes of Scrum in a collaborative project across two classes.
- In the real world, you'll see how Scrum and similar approaches can significantly improve the quality not only of work product, but your work life overall.



Mountain Goat Software, User stories

- https://www.mountaingoatsoftware.com/agile/user-stories

Scrum history

- http://www.scrumguides.org/history.html

The Agile Manifesto

- http://agilemanifesto.org