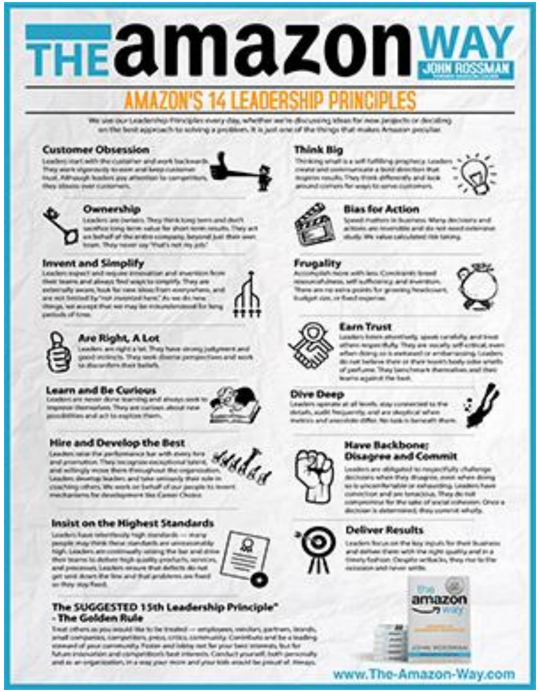
Scrum: A ridiculous name for a great idea

Northwestern | ENGINEERING

You have your press release



THE amazon WAY

JOHN ROSSMAN

AMAZON'S 14 LEADERSHIP PRINCIPLES

[illegible]

Customer Obsession

[illegible]

Ownership

[illegible]

Invent and Simplify

[illegible]

Are Right, A Lot

[illegible]

Learn and Be Curious

[illegible]

Hire and Develop the Best

[illegible]

Insist on the Highest Standards

[illegible]

Think Big

[illegible]

Bias for Action

[illegible]

Frugality

[illegible]

Earn Trust

[illegible]

Dive Deep

[illegible]

Have Backbone; Disagree and Commit

[illegible]

Deliver Results

[illegible]

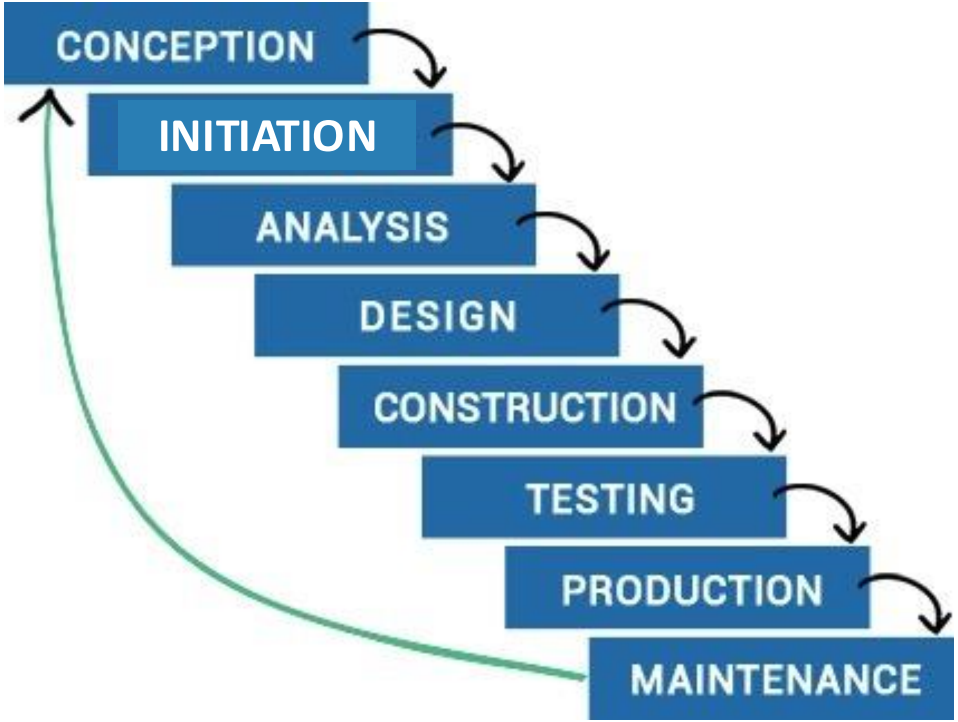
The SUGGESTED 15th Leadership Principle* - The Golden Rule

[illegible]

www.The-Amazon-Way.com

What does the team work on today?

Waterfall: A controlled approach



Like construction and manufacturing workflows, waterfall is a sequential design process. As each stage is completed, developers move on to the next.

Once a step has been completed, you don’t go back. There’s no room for change or error, so a project outcome and an extensive plan must be set in the beginning and then followed carefully.

Waterfall: Issues

1. Once a step has been completed, we can't go back to a previous stage and make changes.

2. It relies heavily on initial requirements and assumptions. If wrong, the project is doomed.

3. If a requirement error is found, or a change needs to be made, the project sometimes has to start from the beginning with all new code.

4. The product is only tested at the <u>end</u>. If bugs are written early, but discovered late, they impact other code.

5. The temptation to delay thorough testing is often very high, as these delays allow short-term wins of staying on-schedule.

6. The plan attend to client’s evolving needs or how products are changing in the marketplace.

Waterfall: Issues

1. It encourages sunk cost mistakes

2. It encourages ownership errors

3. It protects bad decisions

It was clear we needed something different

The world is changing

- Changing nature of work
- Jobs are not for a lifetime
- Satisfaction is more important than money
- The rise of gig economy
- People work on the tasks that appeal to them

The result

- Tasks, not jobs, are the focus
- Ephemeral, task focused teams are the norm
- Teams must be cross-functional
- Technologists/Non-Technologists must learn how to partner with each other
- Everyone must learn how to partner with machines

Tasks, Teams, and Partnership

- Tasks and teams come and go
- You need to move quickly between them
- You need to be responsive to changes in the world
- You need to be willing to try, fail, and move on
- You need to listen to the needs of your user/client
- You need to embrace existing solutions

Agile: A more human approach

Agile: Still controlled, but different

Individuals and interactions	over	Process and tools
Working software	over	Documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

Agile: Same steps but in parallel

Requirements

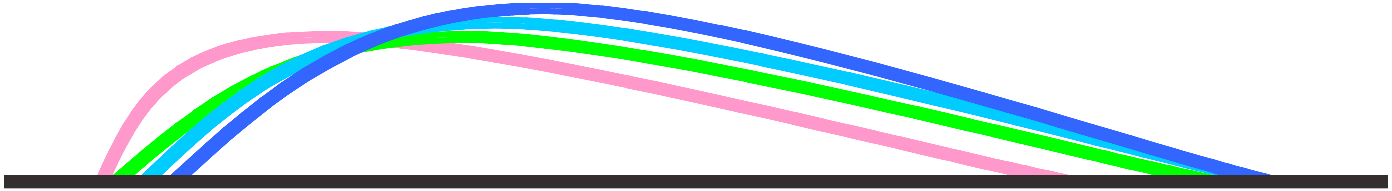
Design

Code

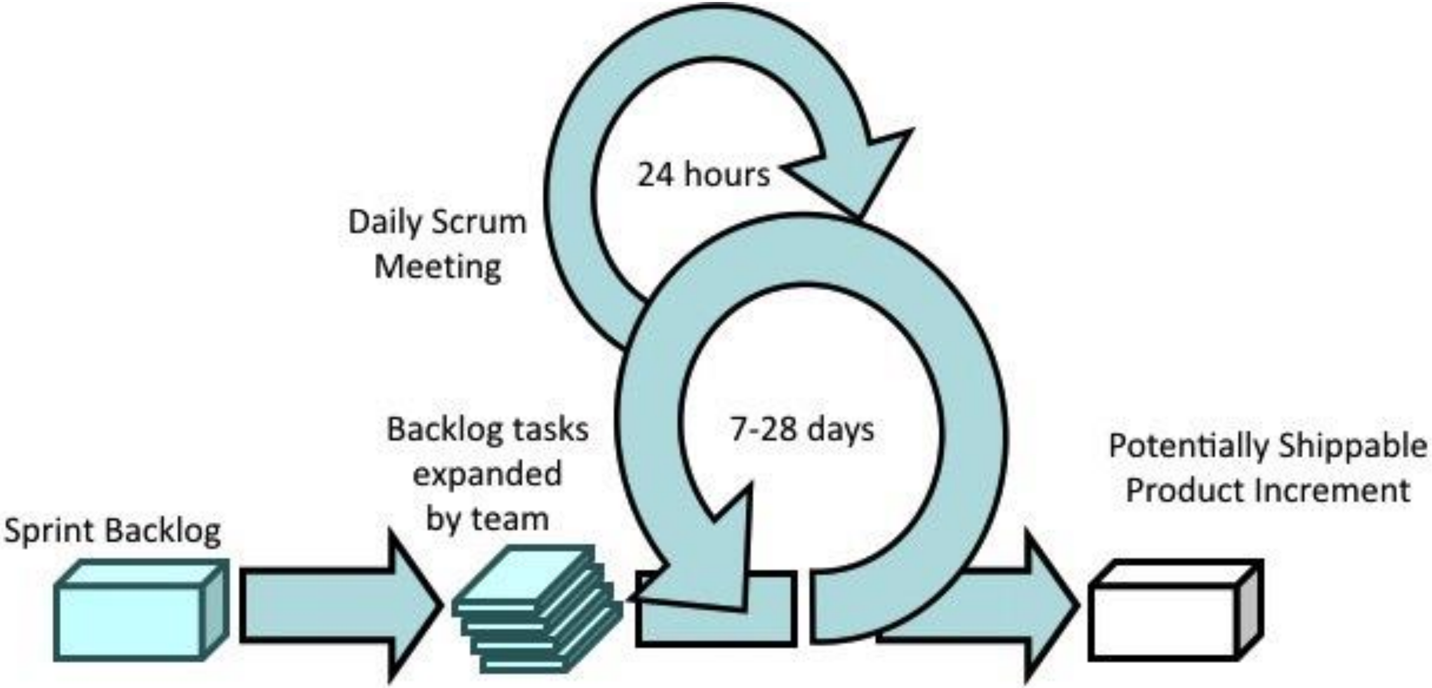
Test

Rather than doing all of one thing at a time…

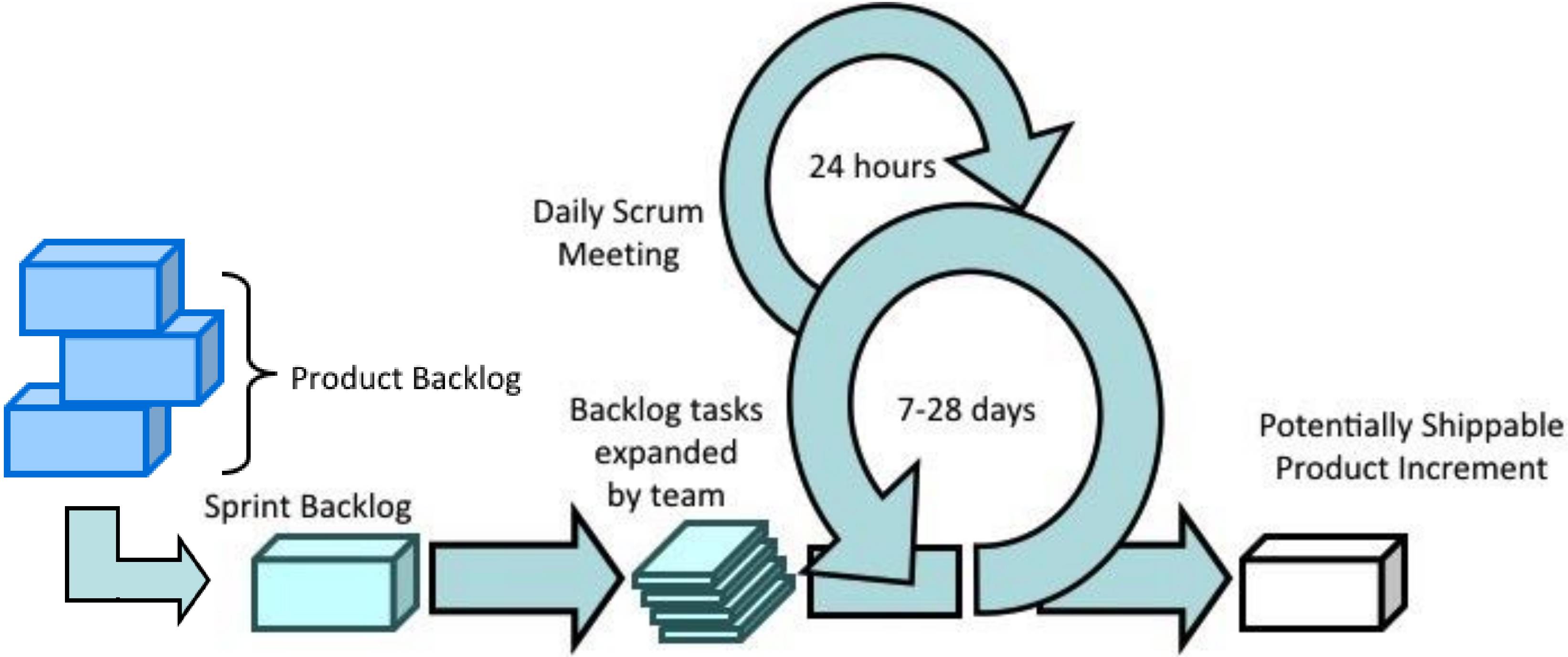
…Agile teams do a little of everything all the time



Scrum: Same steps, but in cycles


24 hours
Daily Scrum Meeting
Backlog tasks expanded by team
7-28 days
Potentially Shippable Product Increment
Sprint Backlog

Scrum: Same steps, but in cycles


24 hours
Daily Scrum Meeting
Product Backlog
Backlog tasks expanded by team
7-28 days
Potentially Shippable Product Increment
Sprint Backlog

Scrum: Roles, Ceremonies and Artifacts

Roles

- Product owner
- Scrum Master
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

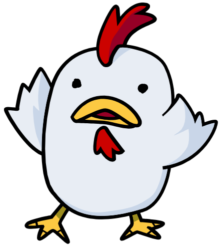
Scrum: Roles, Ceremonies and Artifacts

- Product Owner
  - Possibly a Product Manager or Project Sponsor
  - Decides features, release date, prioritization, $$$

- Scrum Master
  - Typically a Project Manager or Team Leader
  - Responsible for enacting Scrum values and practices
  - Remove impediments / politics, keeps everyone productive

- Project Team
  - 5-10 members; Teams are self-organizing
  - Cross-functional: QA, Programmers, UI Designers, etc.
  - Membership should change only between sprints

Scrum: Different levels of commitment



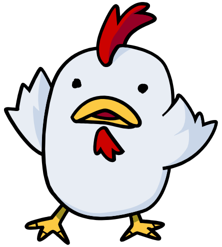
A pig and a chicken are walking down a road. The chicken looks at the pig and says, "Hey, why don't we open a restaurant?" The pig looks back at the chicken and says, "Good idea, what do you want to call it?" The chicken thinks about it and says, "Why don't we call it 'Ham and Eggs'?" "I don't think so," says the pig, "I'd be committed but you'd only be involved."

Scrum: Different levels of commitment

- **Pig**: Team member who has full commitment to the work.

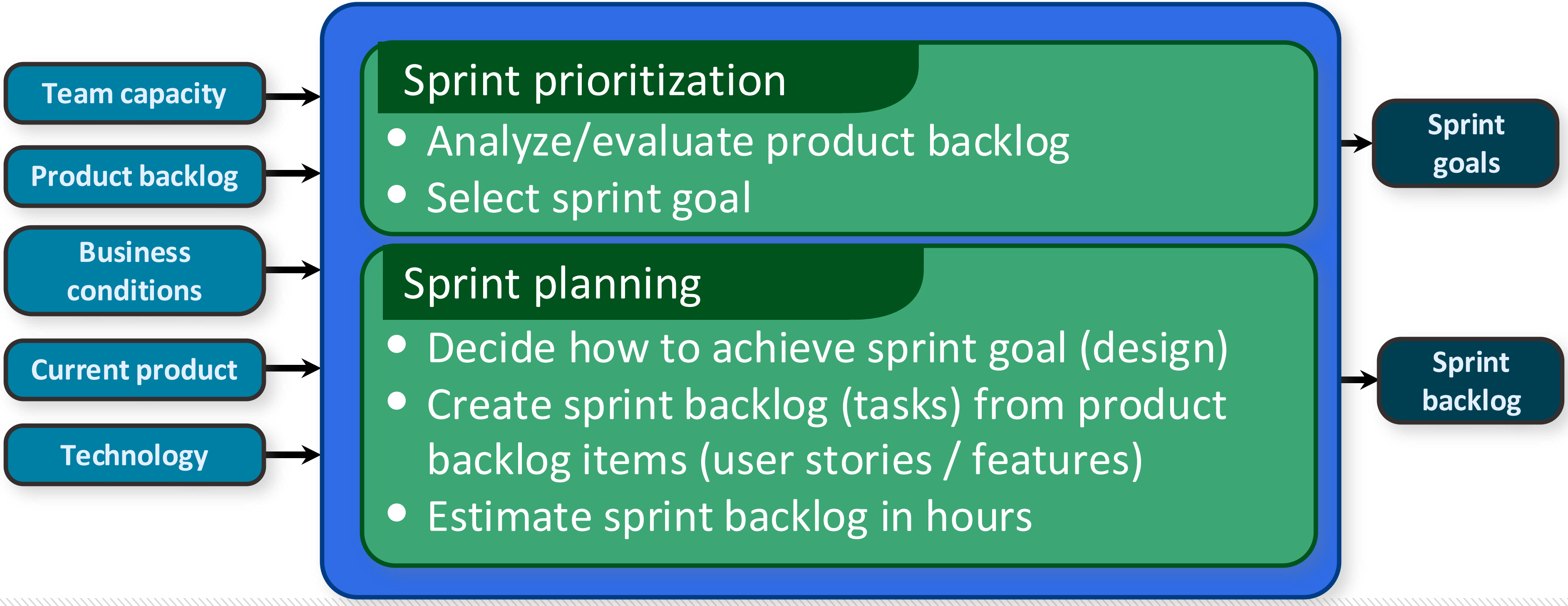
- **You**

- **Chicken**: Individual who is interested in the product but has less skin in the game.



- **Faculty**

Scrum: Planning the Sprint



Team capacity

Product backlog

Business conditions

Current product

Technology

Sprint prioritization

- Analyze/evaluate product backlog
- Select sprint goal

Sprint planning

- Decide how to achieve sprint goal (design)
- Create sprint backlog (tasks) from product backlog items (user stories / features)
- Estimate sprint backlog in hours

Sprint goals

Sprint backlog

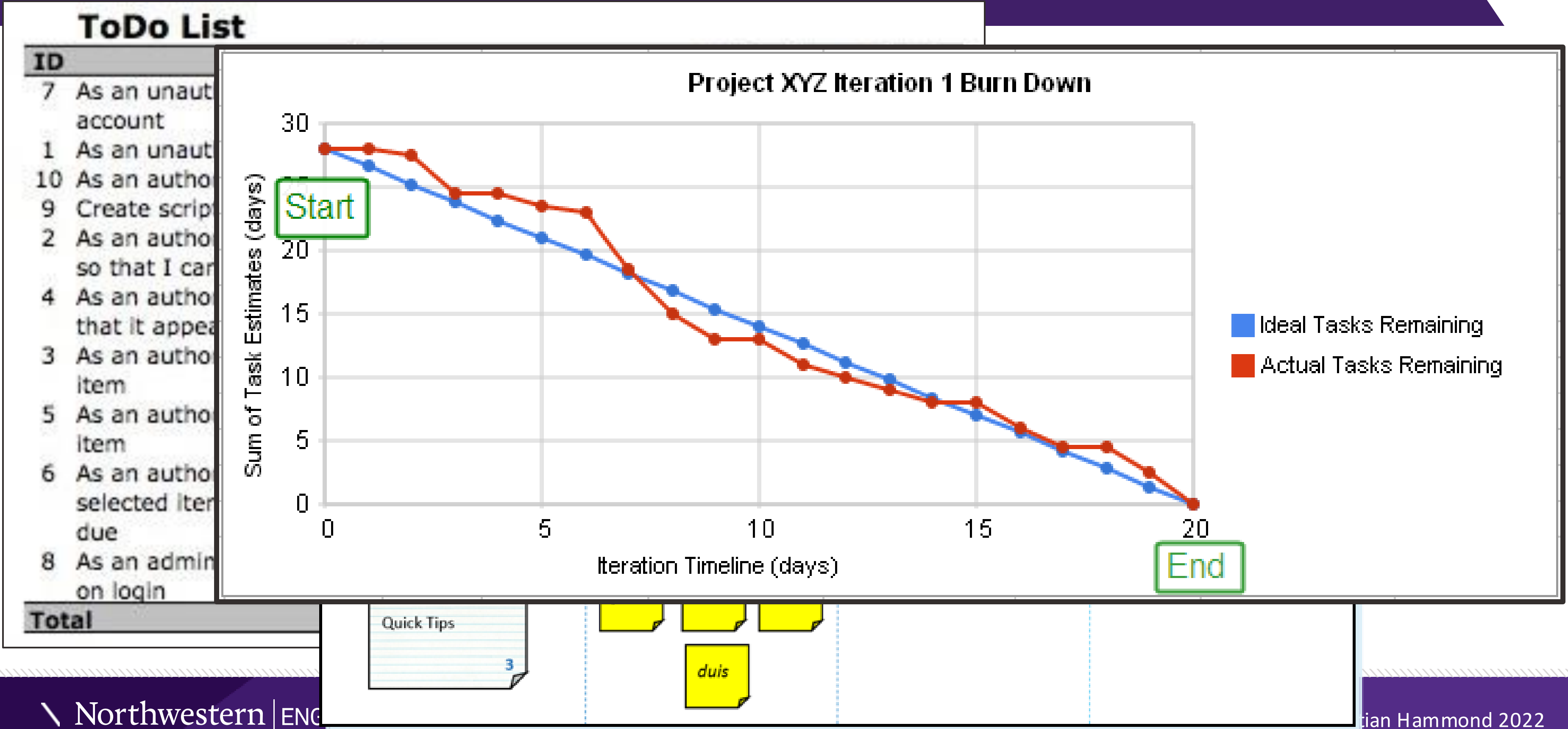
Scrum: Planning the day with a stand up

- Logistics
  - Daily, ~15 minutes, Stand-up
- Not for problem solving
  - Whole world is invited…
  - Only team members, Scrum Master, product owner, can talk
- Three questions answered by each team member:
  - What did you do yesterday?
  - What will you do today?
  - What obstacles are in your way?

Scrum: Documents

- Scrum has remarkably few artifacts
  - Product Backlog (features to be developed/scheduled)
  - Sprint Backlog (features to be developed this sprint)
  - Burndown Charts

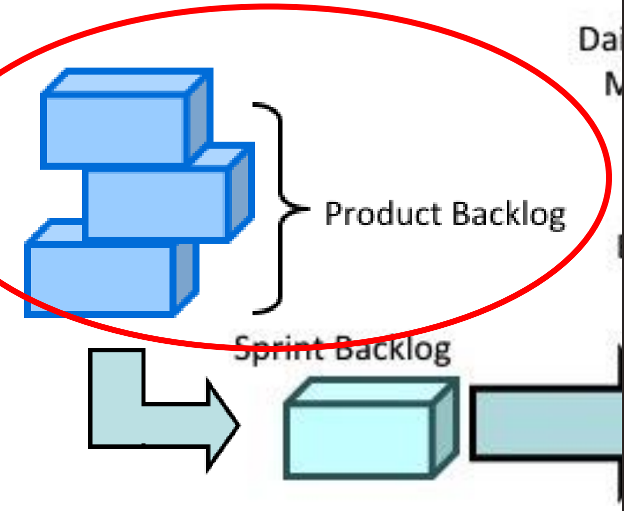
Scrum: Documents



ToDo List

ID	
7	As an unaut account
1	As an unaut
10	As an autho
9	Create scrip
2	As an autho so that I car
4	As an autho that it appea
3	As an autho item
5	As an autho item
6	As an autho selected iter due
8	As an admin on login
Total	

Scrum: Product Backlog



- The requirements
- A list of all desired work on project
- Ideally expressed as a list of user stories along with "story points", such that each item has value to users or customers of the product
- Prioritized by the product owner
- Reprioritized at start of each sprint

Scrum: User Stories

- Instead of Use Cases, Agile project owners use "user stories"
  - **Who** (user role) – Is this a customer, employee, admin, etc.?
  - **What** (goal) – What functionality must be achieved/developed?
  - **Why** (reason) – Why does user want to accomplish this goal?

    As a [user role], I want to [goal], so I can [reason].
- Example:
  - "As a user, I want to log in, so I can access subscriber content."
- **story points**: Rating of effort needed to implement this story
  - common scales: 1-10, shirt sizes (XS, S, M, L, XL), etc.

Scrum: Review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
  - 2-hour prep time rule
  - No slides
- Whole team participates
- Invite the world

Scrum: Retrospective

- The Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and their Definition of Done.
- Assumptions that led them astray are identified and their origins explored. The Team discusses what went well, problems it encountered, and how they were (or were not) solved.
- The Scrum Team identifies the most helpful changes to improve its effectiveness. The most impactful improvements are addressed as soon as possible.
- The Sprint Retrospective concludes the Sprint. It is timeboxed to a maximum of three hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.
- During the Sprint Retrospective, the team discusses:
    - What went well in the Sprint
    - What could be improved
    - What will we commit to improve in the next Sprint

Scrum: Planning

Sprint planning is an event in scrum that kicks off the sprint.

The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved.

Sprint planning is done in collaboration with the whole scrum team.

Scrum: Planning

The What – Objectives or Goals of the Sprint

The How –The work necessary to deliver the Sprint Goals.

The Who – Product owner and the development team.

The Inputs – Product backlog and burndown.

The Outputs –The Sprint backlog.

Scrum: Planning

The What – The product owner describes the objective (or goal) of the sprint and what backlog items contribute to that goal. The scrum team decides what can be done in the coming sprint and what they will do during the sprint to make that happen.

Scrum: Planning

The How – The development team plans the work necessary to deliver the sprint goal. Ultimately, the resulting sprint plan is a negotiation between the development team and product owner based on value and effort.

Scrum: Planning

The Who – You cannot do sprint planning without the product owner or the development team. The product owner defines the goal based on the value that they seek. The development team needs to understand how they can or cannot deliver that goal. If either is missing from this event it makes planning the sprint almost impossible.

Scrum: Planning

The Inputs – A great starting point for the sprint plan is the product backlog as it provides a list of 'stuff' that could potentially be part of the current sprint. The team should also look at the existing work done in the increment and have a view to capacity.

Scrum: Planning

The Outputs – The most important outcome for the sprint planning meeting is that the team can describe the goal of the sprint and how it will start working toward that goal. This is made visible in the sprint backlog.

Scrum: But this is a class not a company

- Ideas such as daily stand ups make less sense for distributed students working on a variety of projects/classes
- What does this process mean for non-software projects
- How does this relate to projects where we are buying, not building

Scrum is an approach not a religion

- It is an agile, lightweight process
- Can manage and control software and product development
- Uses iterative, incremental practices
- Has a simple implementation
- Increases productivity

- Reduces time to benefits
- Embraces adaptive, empirical systems development
- Is not restricted to software development projects
- Embraces change, including with itself…

Scrum's Tenets

- Link user needs to technology capabilities
- Users may not be able to articulate their needs
- Vendors may not be able to articulate their capabilities
- Listen to understand not follow

Your Considerations

- **Potential areas of responsibilities**
  - User needs
  - Workflow documentation
  - Requirements identification
  - Bench research
  - Tool development and testing
  - Documentation
  - Work plan development
  - Weekly reporting
- **Considerations**
  - Skills/experience
  - Aspirations (short and longer-term)
  - Competencies