Northeastern University

Software QA & Testing
Fundamental of Testing

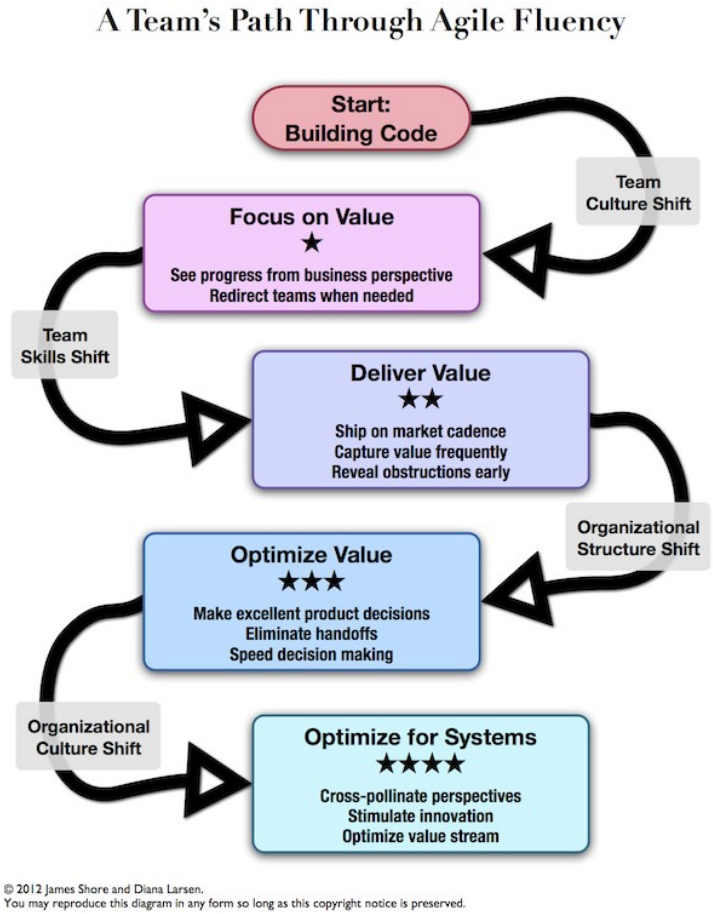
Medi Servat

INFO6255

m.servattalab@northeastern.edu

Agile Development

1. Adaptive rather than predictive
2. People oriented than process oriented
3. Project is divided into milestones
4. Minimum viable product (MVP)
5. Scrum teams
6. Milestones have sprints
7. Each sprint has multiple stories
8. Stories are the deliverables



Agile Versus Traditional Waterfall

Metric	Waterfall	Agile
Planning scale	Long-term	Short-term
Distance between customer and developer	Long	Short
Time between specification and implementation	Long	Short
Time to discover problems	Long	Short
Project schedule risk	High	Low
Ability to respond quickly to change	Low	High

https://www.venveo.com/blog/agile-vs-waterfall-project-management

Automated Testing in Agile World

Each Sprint adds **new features** and refactoring as the design might evolve

Previous code should continue to work

Automated testing framework provides the firepower for this continuous development

Automated tests are **added incrementally** and developing a prefect testing framework is not the intention at the start

Meaningful tests **add more value** than automated everything

Principles behind the Agile Manifesto

12 Principles:

1. Our highest **priority is to satisfy the customer** through early and continuous delivery of valuable software.

2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.

3. **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. **Business people and developers** must work together daily throughout the project.

https://martinfowler.com/agile.html

Principles behind the Agile Manifesto (Continued)

5. Build projects **around motivated individuals**. Give them the environment and support they need, and trust them to get the job done.

6. The **most efficient and effective method** of conveying information to and within a development team is **face-to-face** conversation.

7. **Working software** is the primary measure of progress.

8. Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a **constant pace** indefinitely.

https://martinfowler.com/agile.html

Principles behind the Agile Manifesto (Continued)

9. Continuous attention to **technical excellence** and good design enhances agility.

10. **Simplicity**--the art of maximizing the amount of work not done--**is essential**.
    - Make the design as simple as possible and don't incorporate every potential future requirement in the initial design.
11. The best architectures, requirements, and designs emerge from **self-organizing teams**.

12. At regular intervals, the **team reflects on how to become more effective**, then tunes and adjusts its behavior accordingly

https://martinfowler.com/agile.html

Agile – Scrum Framework

•There are several Agile Frameworks: **Scrum, Kanban, XP and Others**

- **Scrum:** A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

  - Scrum is **lightweight** and simple to understand.
  - Scrum emphasizes **self-organizing**, cross functional teams.
  - Scrum does **not prescribe any technical practices**, rather it **emphasizes** management practice such as the role of Scrum master.

Kanban

- **Kanban** is a **popular framework** used to implement agile software development. It requires **real-time communication** of capacity and full transparency of work.

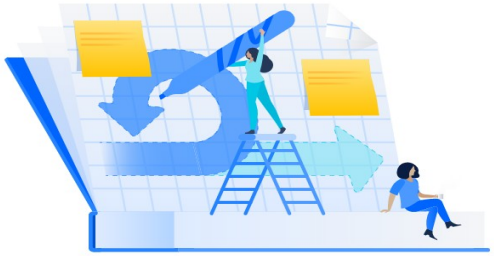
- Work items are represented visually on a **kanban board**, allowing team members to see the state of every piece of **work at any time**. A **Kanban Board is also used to visualize work** and optimize the flow of the work among the team.

- While physical boards are popular among some teams, **virtual boards** are a crucial feature in any agile software development tool for their **traceability, easier collaboration, and accessibility** from multiple locations.

https://www.atlassian.com/agile/kanban

Scrum vs. Kanban

- **Kanban** is primarily concerned with **process improvements,** while **Scrum** is concerned with getting more **work done faster**.



	Scrum	Kanban
Cadence	Regular fixed length sprints (ie, 2 weeks)	Continuous flow
Release methodology	At the end of each sprint	Continuous delivery
Roles	Product owner, scrum master, development team	No required roles
Key metrics	Velocity	Lead time, cycle time, WIP
Change philosophy	Teams should not make changes during the sprint.	Change can happen at any time

https://www.atlassian.com/agile/kanban/kanban-vs-scrum

Scrum Factors and Core Team values

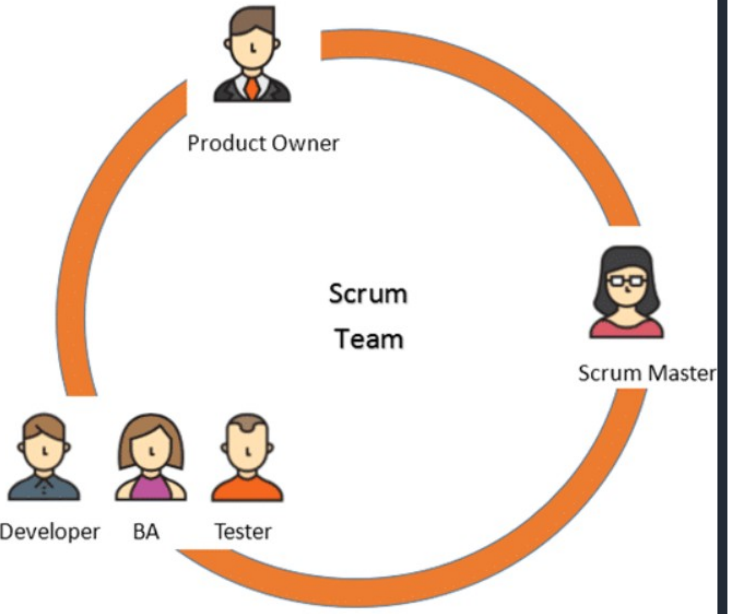
3 Important Factors of Scrum:

1. Transparency of the Process.
2. Frequent Inspection of the Artifacts.
3. Adaptation, or making adjustments after inspection.

Core Team Values of Scrum:

1. Team Commitment to meet the goals.
2. Value Focus to deliver better quality, faster.
3. Openness with the team members to adapt better.
4. Respect the team to allow them voicing concerns.
5. Value Courage to allow greater challenge.

Scrum Roles



- **Here are the Scrum roles**
  - Product Owner
  - Scrum Master
  - “The Team”
    - QA
    - Developers
    - BSA

The Complete Agile Team

- The Whole-Team Approach – The team works together on all projects:
  - Project Manager
  - Tester
  - Business Analyst
  - Business Representative
  - Database Analyst
  - Networking/Infrastructure
  - User Experience
  - Developer

Scrum Master

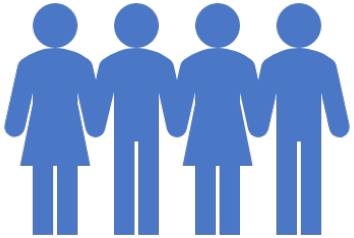
The Scrum Master is responsible for the project's success

- Role of the Scrum Master
  - **Represents** management and the project
  - **Responsible** for enacting Scrum values and practices
  - **Removes** impediments
  - **Ensures** that the team is productive
  - **Enables** close cooperation across all roles and functions
  - **Shields** the team for external influences
  - Is a **Servant Leader**

Scrum Master’s “Don’ts”

Scrum Master:

- **Does not** own the product decisions on product owner’s behalf
- **Does not m**ake estimates on team’s behalf
- **Does not** make technology decisions on team’s behalf
- **Does not** assign the tasks to the team members
- **Does not** manage the team



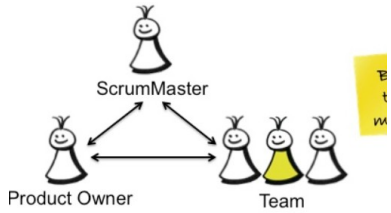
Servant Leadership

- **What is a Servant Leadership?**
  - Change from a **traditional leadership** (workers need to be monitored closely)
  - It is based on **trust** (the team members to be self-motivated)
  - **The Servant Leader:**
    - **Protects** the team from outside **distractions**
    - **Removes** impediments
    - **Facilitates** the team to address the tasks and resolve problems
    - **Removes obstacles** that would prevent the team providing business values

Daily Scrums

- The scrum meetings criteria:
  - **Daily**
  - **Same Time**
  - **Same Place**
  - **Stand up**
- Everyone is invited, however only the team members and the scrum master can talk.
- 3 Questions are asked:
  - **What did you do yesterday?**
  - **What will you do today?**
  - **Is anything blocking you??**

Product Owner



- **The Product owner is responsible for the product success**
  - The Product Owner role is **filled by one person** who has a vested interest in the product:
    - **Is the Subject Matter Expert (SME):** Understands the domain well enough
    - **End User and Customer advocate:** Describes the product with understanding of the users and the functionality of the product
    - **Business Advocate:** Understands the needs of the organization for the product
    - **Communicator:** Capable of communicating vision and intent
    - **Decision Maker:** Be the final decision maker for hard product decisions

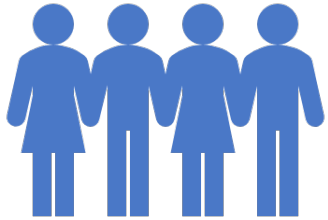
Product Owner Roles

- **Defines** the features of the product
- **Decides** on Release dates and contents
- Is **responsible** for the **Total Cost of Ownership** and the **Profitability** of the Product
- **Orders** features according to Market value
- **Adjusts** features and priority every sprint as needed
- **Available** daily to answer questions
- **Inspects** product progress

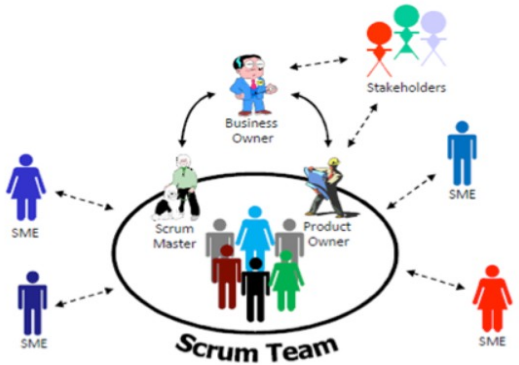
Product Owner Roles…

- The Product Owner Does Not
  - **Choose** how much work will be accomplished in the Sprint.
  - **Change** anything within the Sprint once it has started.

The “Team”



- **The team’s Dos and Don’ts**
  - **Self organizing** (Self empowered, Responsible with no individual egos)
  - Should include **developers, testers, UI designers**, etc.
  - Full Time (FTEs)
  - **Should not change** during Sprints
  - **Should not skip** meetings
  - **Should not stop** working when they hit roadblock



Agile Project Process (Events)

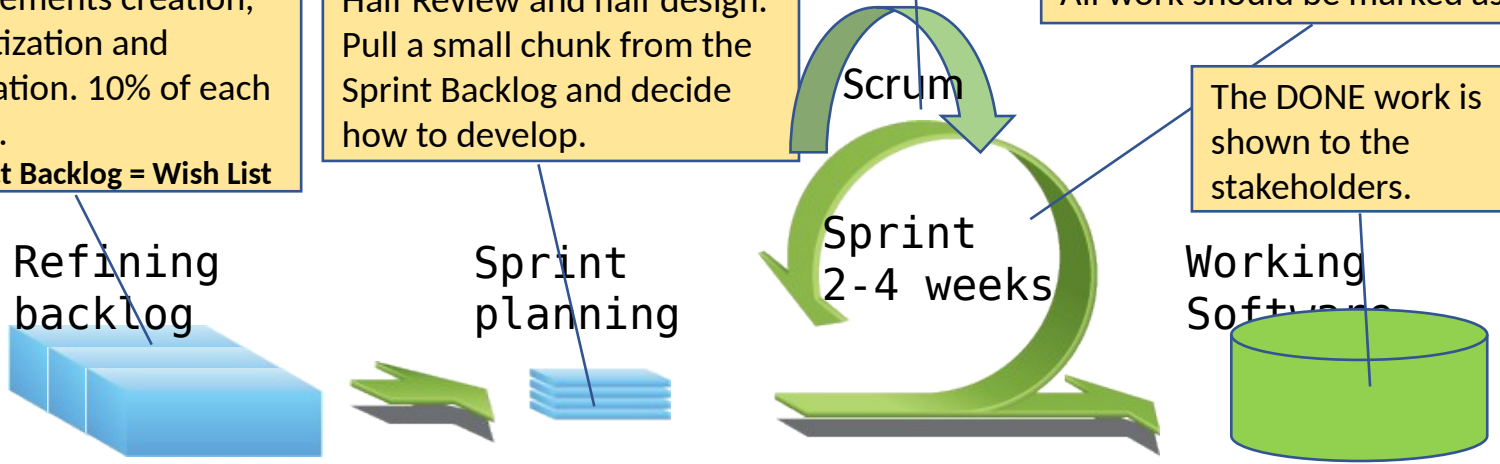
Daily 15 minute scrums, same time and same place

Involves creating requirements creation, prioritization and estimation. 10% of each Sprint.
Product Backlog = Wish List

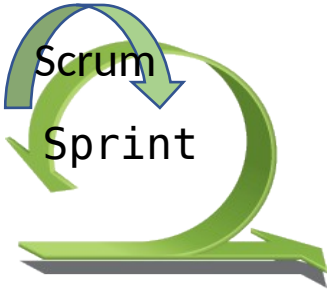
Occurs every Sprint.
Half Review and half design.
Pull a small chunk from the Sprint Backlog and decide how to develop.

Sprints are a max length of 1 month.
All work should be marked as DONE.

The DONE work is shown to the stakeholders.



Sprints



Scrum projects progress via a series of Sprints

- During a **Sprint** the Scrum team takes a small set of features from

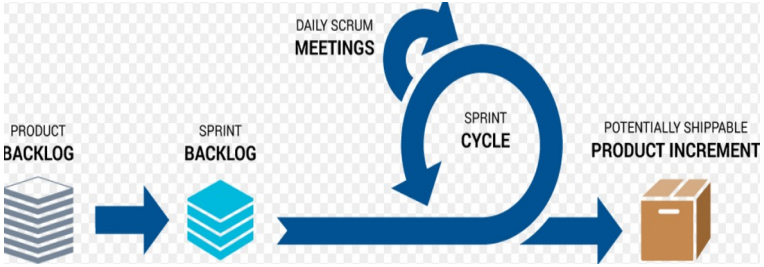
  Idea -> Coded -> Tested

- At the end of the **Sprint** the team has gone through:

  Idea -> Coded -> Tested -> Integrated -> Delivered

- **There should be No Changes during a Sprint**

Sprint Durations, Pros and Cons



- **Four-Week Sprint:**
  - **Pros:** easy to roadmap, low process load
  - **Cons**: Long turnaround and mini Waterfall
- **One Week Sprint:**
  - **Pros:** Fast Turnaround, High Energy
  - **Cons:** Minimal customer feedback, Lack of a roadmap, Relatively heavy or no process.

The Scrum Events

- **Sprint Planning:** 2 hours per week on Requirements & Design
- **Sprints:** One Month Max
- **Daily Scrums:** 15 minutes Daily
- **Backlog Refinement (grooming):** 2 hours/week
- **Sprint Review:** a couple of hours a week
- **Sprint Retrospectives:** a few hours for a one month Sprint

Agile Terminology

- **User Story –** A defined feature that the user desires.
- **Theme –** User Stories that have something in common**.**
- **Epic –** A large User Story that may be able to be broken down into smaller stories as they get ready for development.
- **Spike –** A Special user story used for investigation activity.
- **Product Backlog** – A list of desired work on the project
- **The Sprint Backlog** – The Product Owner works with the Development team to select stories based on a Sprint goal
- **Burndown Charts** – Tracks work remaining. Story Points vs. Days.
  - **Sprint Burndown Chart** – Show work remaining
  - **Release Burndown Chart** – Show work remaining for the entire release
- **Velocity** – Rate of Progress.

Agile Planning

- **Product Vision** – is key to success
- **Product Roadmap** – shows progress toward strategy
- **Release Plan** – Occurs once the vision and roadmap are complete
- **Sprint Plan** – Sprint Planning
- **Daily Plan** – Scrums

- **Minimum Viable Product:**
  - The highest return on Investment.
  - To test an idea by exposing an early version to target users

References

1. https://martinfowler.com/agile.html
2. http://info.thoughtworks.com/rs/thoughtworks2/images/ebook-agile-software-testing.pdf?aliId=26951538
3. Https://Scrum.org
4. http://agilemanifesto.org
5. https://www.atlassian.com/agile/kanban/kanban-vs-scrum

Problems with the Agile Implementation

- It is **difficult to change** the culture of an organization.
- **Waterfall** is in place and is hard to change.
- Not many people like **“Change”.**
- **The Leadership** has not bought into it.
- Agile can be implemented in any organization by:
    - **Education**
    - **Execution**
    - **Continuous Improvement**

Your First Group Assignment

- **Do a research Online and find**
  - **One Success Story about how the Agile process changed the mindset and delivered a good quality product in an organization**

-- OR --

  - **One Failed Story about how the Agile process did not work in the organization and why it failed**

Assignment – Agile Implementation – Failure or Success

- Each team will do a research Online to find a company that went through an Agile transformation from Waterfall. Describe a success or a failure. The presentation will be just a few pages long. It should contain the following:
  - The company name.
  - When it happened (the date).
  - Why it happened.
  - Explain the reason for the success or the failure in bullet points.

- It should be clear and concise, and **ONLY one of the members of the team** will be presenting it to the class.
- The Presentation will take between 5 to 7 minutes only.
- There will be one submission per group on Canvas due at any time before the next class.
- Make sure your group name is included in the file name that is submitted.