SSW-555: Agile Methods for Software Development

Scrum

Instructor: Prof. Zhongyuan Yu

School of Systems and Enterprises

Stevens Institute of Technology

This Lecture is based on material from Prof. Jim Rowland

Acknowledge

Material for this lecture comes from a variety of sources including:

- Agile Software Development with Scrum, Schwaber and Beedle, 2002
- Introduction to Agile Methods, Ashmore and Runyan, 2015
- http://www.innolution.com/essential-scrum
- https://www.mountaingoatsoftware.com/articles/toward-a-catalog-of-scrum-smells

Today's Topics

- Origin of Scrum
  - Defined vs Empirical processes
- Compare Plan Driven to Chaos to Scrum
- Scrum practices
  - Process
  - Roles
  - Meetings
  - Artifacts

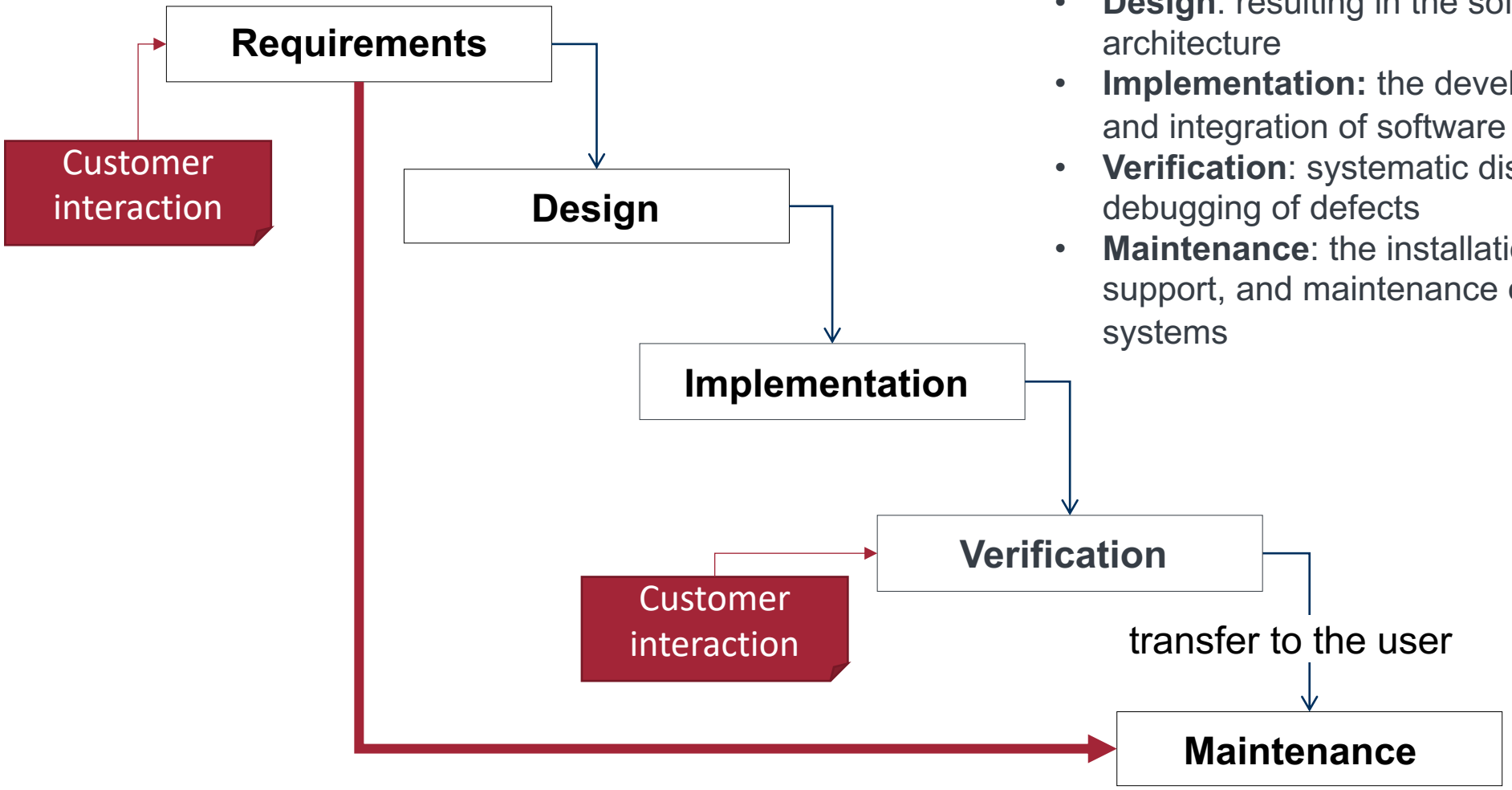
Origin of Scrum

- Software managers needed a process to control software projects
- Plan-based methods assumed that the development process was *predictable*
  - Any problem could be solved with a little more planning
- BUT some projects required more *adaptation* than expected
  - React to *changing* customer requirements
- Defined vs Empirical processes

Assumptions of *defined processes* (used by plan-driven approaches)

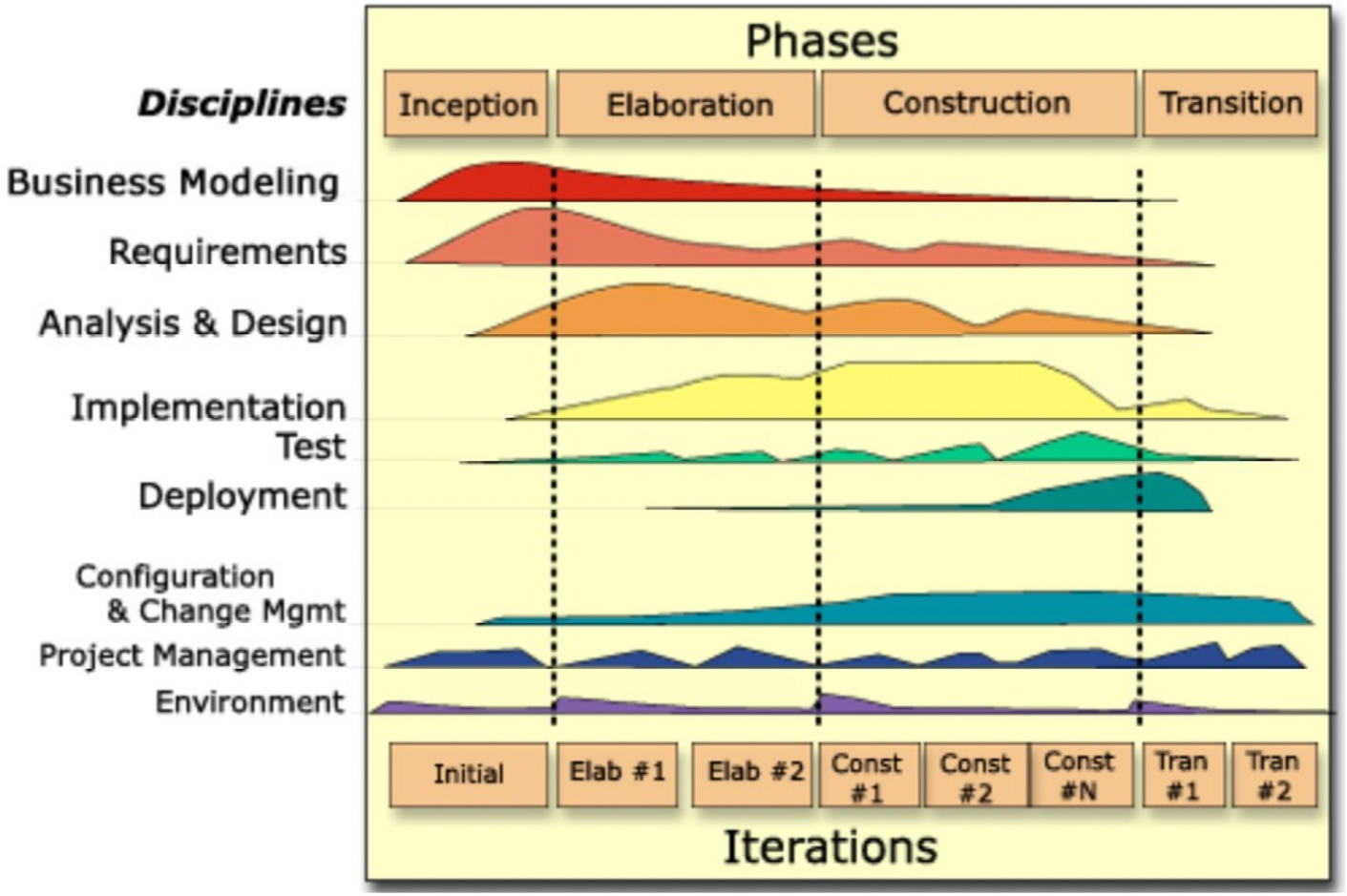
- System may be described by equations that predict response to stimulus or input (UML Diagrams)
  - E.g., given input X should produce output Y
  - It works for manufacturing, so it should work for software, right?
- Variables that influence the process are well-understood:
  - may be predicted from the task
  - may be measured in the environment
- Management consists of controlling and/or measuring the variables
- Unfortunately, software projects don't always work that way…

Waterfall Model



- **Requirements**: captured in a product requirements document
- **Design**: resulting in the software architecture
- **Implementation:** the development, proving, and integration of software
- **Verification**: systematic discovery and debugging of defects
- **Maintenance**: the installation, migration, support, and maintenance of complete systems

RUP phases, iterations and disciplines



Explore more details: https://sceweb.uhcl.edu/helm/RationalUnifiedProcess/

Original reference: https://www.ibm.com/support/pages/rational-unified-process-rup-plug-ins-rational-method-composer-751

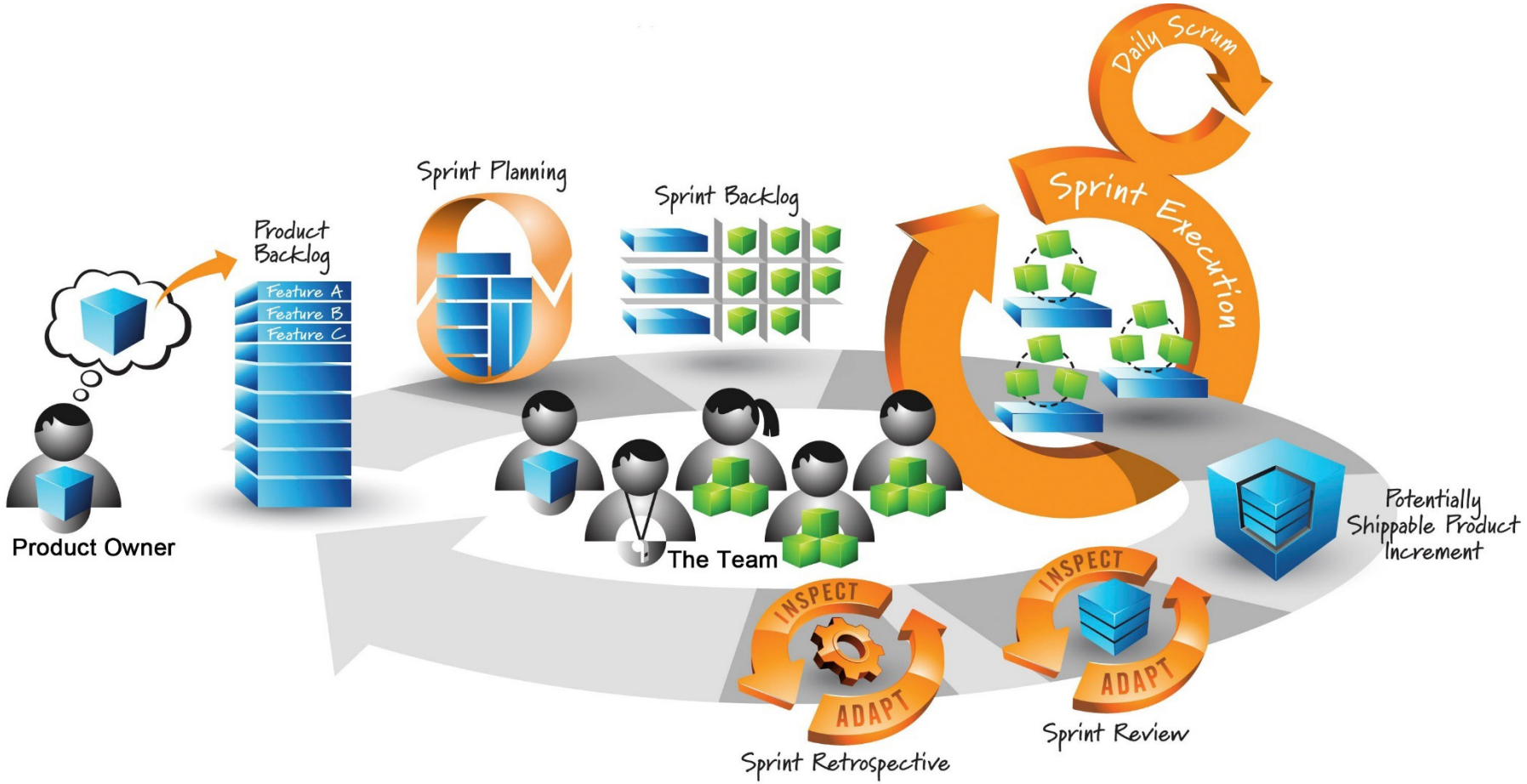
Customer Chaos Model

- Many customers asking developers for many features
- Developers try to accommodate “loudest” voices (or biggest bosses)
- Leads to chaos and low morale and low productivity

Assumptions of *empirical methods* (used in Agile approaches)

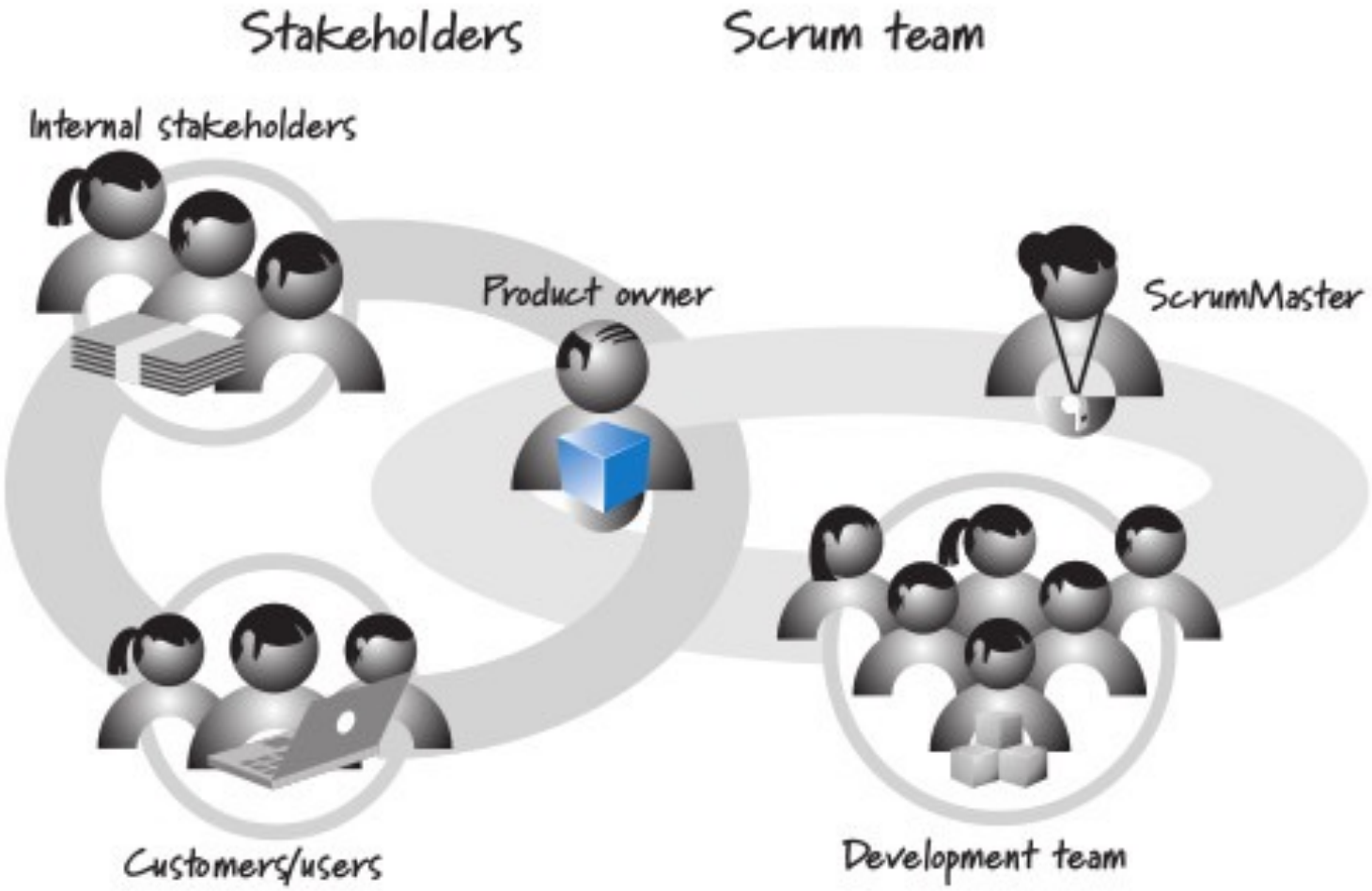
- System cannot be described with simple equations
  - transformation from input to output is complex
  - some aspects of transformation may have no science
- Cannot easily repeat the process from input to output
  - variables that influence the process are not all known
  - noise in environment is too difficult to avoid or predict
- Management consists of constantly monitoring and adapting

Scrum Process



http://www.innolution.com/essential-scrum/table-of-contents/chapter-2-scrum-framework

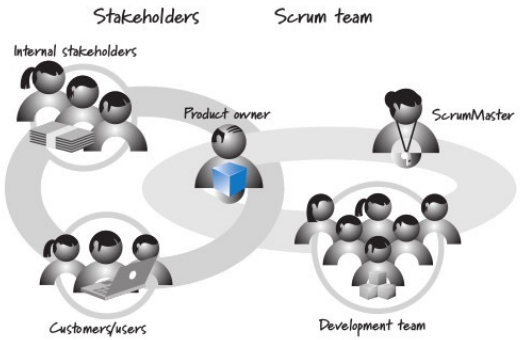
Scrum Roles



Source: https://chintanjariwala.files.wordpress.com/2016/05/po.jpg?w=507&h=322

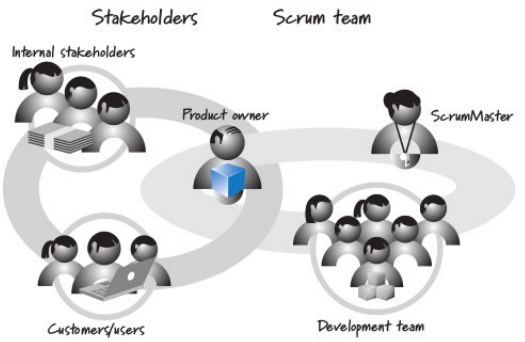
Product Owner

- Represents the customer and the customer's needs
- Agile principle #4: "Businesspeople and developers must work together daily throughout the project"
- Product visionary
  - What should the product do?
  - What are the features?
  - Provides user stories
- Responsible for Return On Investment (ROI)
  - Communicates with stakeholders/investors



Product Owner

- Has final say over the product and releases
  - Decides whether to release increments to customers/users
- Manages and maintains the Product Backlog of desired features
  - Sets priorities for sprints so developers are always working on most important features
- Is an active member of the Scrum Team
  - Ideally co-located with the developers
  - Available to developers to answer questions about features and priorities



Development Team

- Small: 3-10 software developers
  - < 3: may encounter skill constraints
  - > 10: too much coordination
  - Ideally, co-located in one location
- Cross-functional
  - Possess all needed skills: development, test, specialists
- Self-organizing
  - ***Team*** chooses process, roles, and tasks; not the “boss”

Scrum Master

- Scrum Master is a servant-leader (not the boss)
  - Does not make technical or business decisions
  - Developers “report” to the team, not to the Scrum Master
- Different from Plan Driven Project Manager (PM)
  - PM responsible for accountability and enforcement
  - Scrum Master is a coach and collaborator
- Different from Plan Driven IT Manager
  - IT Managers make all the decisions and direct the team
  - Scrum teams are self organizing

Scrum Master

- Helps development team practice Scrum with Best Practices
  - Manages the Scrum artifacts
    - E.g. Burn Down charts and Kanban Boards
  - Facilitates Scrum events
    - E.g. Daily Stand-up meetings
  - Remove impediments
    - Protects the development team from outside interference
- Helps product owner in several ways:
  - Finding techniques for Product Backlog management
  - Understand project planning
  - Facilitate Scrum events as requested or needed

Scrum Examples

Scrum can be applied to different sized projects and teams

Small Start-Up	Mid-sized company	Large Multi-national
One Scrum Team One project	Multiple scrum teams Independent projects	Many scrum teams One large project
Roles: • Product Owner: Marketing/Sales Director • Dev Team: 3 developers, 1 tester, multiple roles • Scrum Master: developer	Roles: • Stakeholders: Sales/Account managers • Product Owner: Product owner • Dev Team: 10-12 developers, 4 testers dedicated roles • Scum Master: Dedicated Scrum Master	Roles: • Stakeholders: CIO, CTO, CFO, VPs • Product Owner: Product Manager for each part • Dev Teams: Multiple teams of 10-12 developers, 4 testers dedicated roles • Scrum Master: Dedicated Scrum Masters

Introduction to Agile Methods: Ashmore and Runyan, 2015

Backlogs – what must be accomplished

Product Backlog

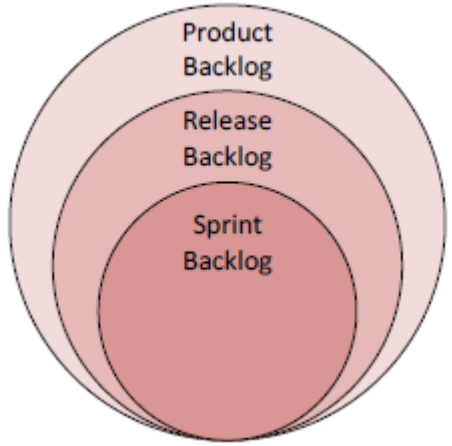
- all features needed in product or service
- any stakeholder can add items, including development team
- may include non-visible features, like underlying technology

Release Backlog

- subset of Product Backlog
- features needed in next product release
- defined by the Product Owner

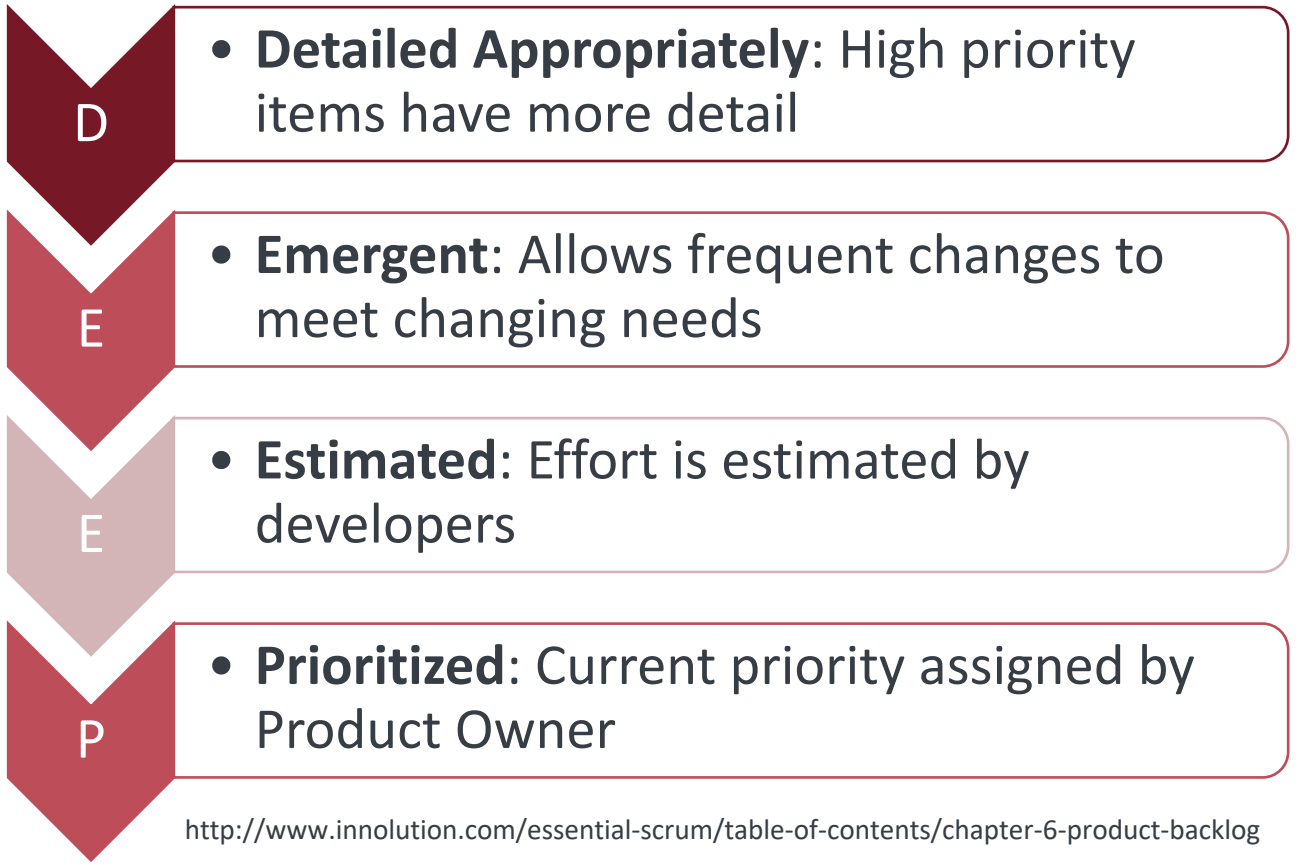
Sprint Backlog

- subset of Release Backlog
- features to be completed in next sprint



Managing the Product Backlog

- Product backlog drives the Scrum process
  - Developers should deliver the highest priority items in the next sprint
- User stories should be **DEEP**

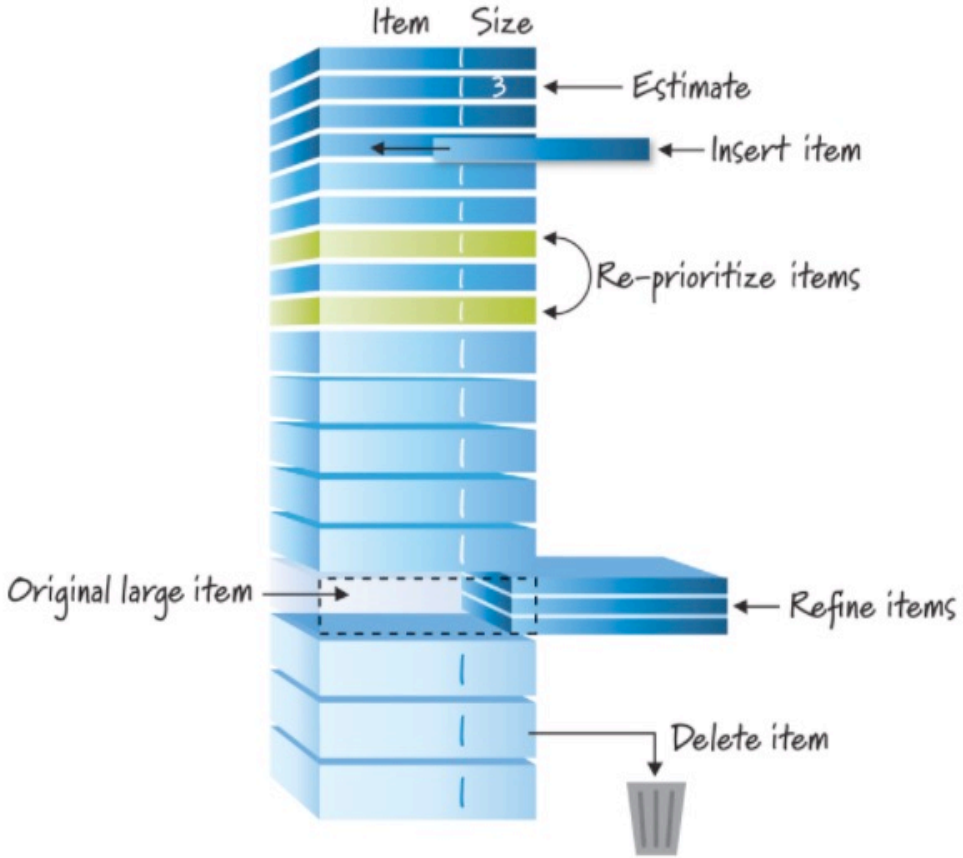


http://www.innolution.com/essential-scrum/table-of-contents/chapter-6-product-backlog

Grooming the Product Backlog

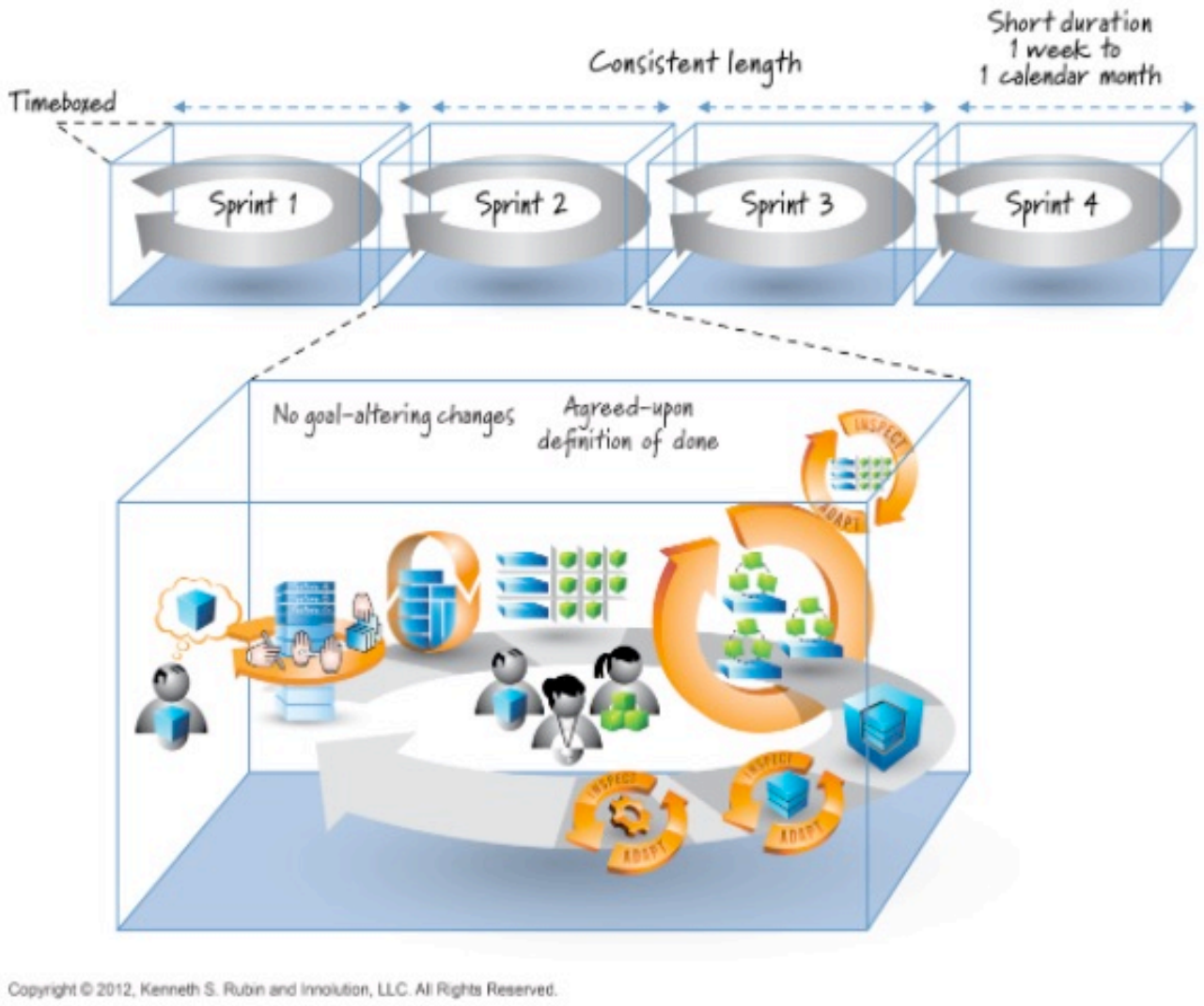
3 tasks

- Creating and refining
  - Anyone can suggest items
- Prioritizing items
  - Product Owner
- Estimating items
  - Developers



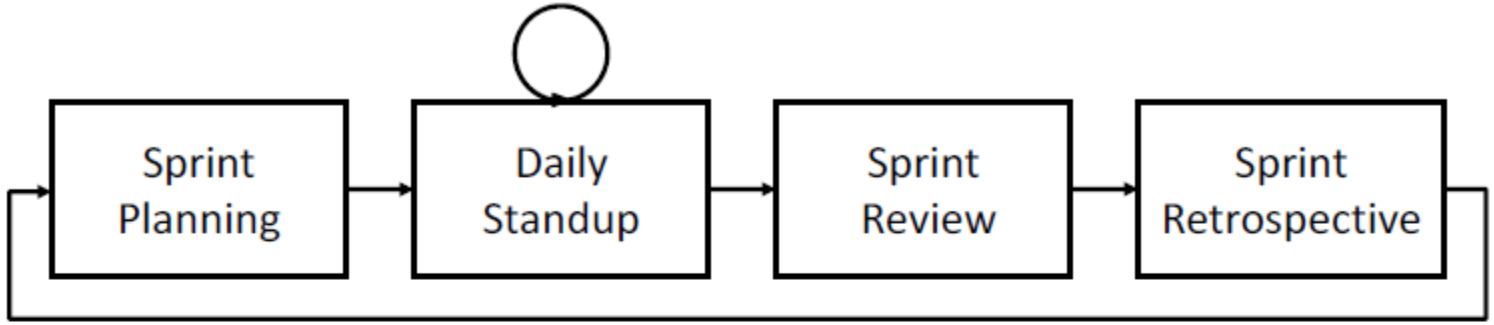
http://www.innolution.com/essential-scrum/table-of-contents/chapter-6-product-backlog

Sprints



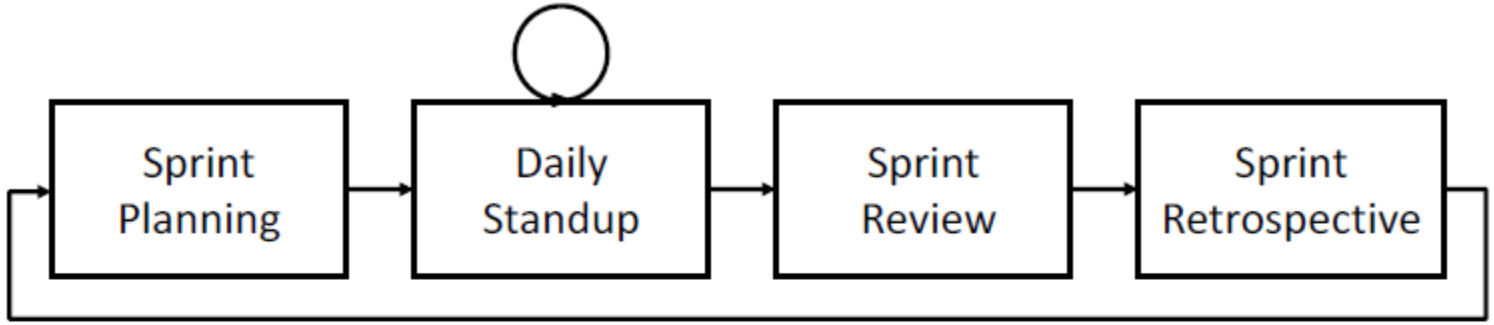
http://www.innolution.com/essential-scrum/table-of-contents/chapter-4-sprints

Scrum Meetings



	Sprint Planning	Daily Standup	Sprint Review	Sprint Retrospective
Who	Product Owner Scrum Master Dev Team	Dev Team Scrum Master	Product Owner Stakeholders Scrum Master Dev Team	Dev Team Scrum Master
Purpose	Which features?	Where are we? Any problems? Any questions?	Review Sprint results with Product Owner	How to improve the process?

Scrum Meetings



	Sprint Planning	Daily Standup	Sprint Review	Sprint Retrospective
Who	Product Owner Scrum Master Dev Team	Dev Team Scrum Master	Product Owner Stakeholders Scrum Master Dev Team	Dev Team Scrum Master
Purpose	Which features?	Where are we? Any problems? Any questions?	Review Sprint results with Product Owner	How to improve the process?

Sprint Planning

- Who: Product Owner, Development Team, Scrum Master
- Define sprint backlog

  Do ***1. Product Owner*** identifies needed features

  2. Developers identify tasks required to deliver features

  3. Developers develop a task list

  4. Developers compute time needed to complete task list

  until development time < sprint duration (2-4 weeks)
- Define sprint goal: business purpose for tasks in this sprint

  e.g., define and implement the UI
- Create “Definition of Done”

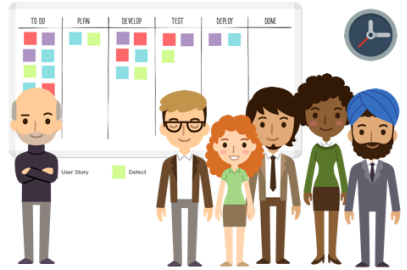
How do we know it’s “done”?

- Definition of Done
  - Everyone (Product Owner and Development Team) must agree on what it means for a backlog item to be "Done”
- This might depend on what the larger organization expects of items for delivery
- This might depend on what the team chooses for standards, e.g.
  - Code passes separate QA process?
  - Code reviews?
  - Documentation available?

Daily Standup Meeting

- Who: Dev Team, Scrum Master, [Product Owner optional]
- Features and priorities are ***"locked" (or frozen)*** for duration of sprint
  - Provides stability for developers during the sprint
- Team meets daily to discuss progress – daily standup meeting
  - Need to monitor progress and adapt to changes
  - Each developer reports:
    - what they did since last Scrum
    - what they will do before next Scrum
    - what impediments require action
- Developers create a new build at least once per day

The Daily Standup: The Art of Standing up and Talking



Daily Standup Meeting

Creative ways to limit the length of the daily standup

https://www.linkedin.com/pulse/daily-plank-meeting-going-agile-literally-jeyaraj-nagarajan

Sprint Review Meeting

Who: Product Owner, **Stakeholders**, Dev Team, Scrum Master

Demonstrate features implemented during the sprint to the customer

Review progress

- compare planned features to actual features
- re-estimate incomplete features and add to product backlog
- collect new features as user stories
- update product architecture if needed

Reflect on sprint –What went well? What must improve?

Brainstorm and plan for next sprint

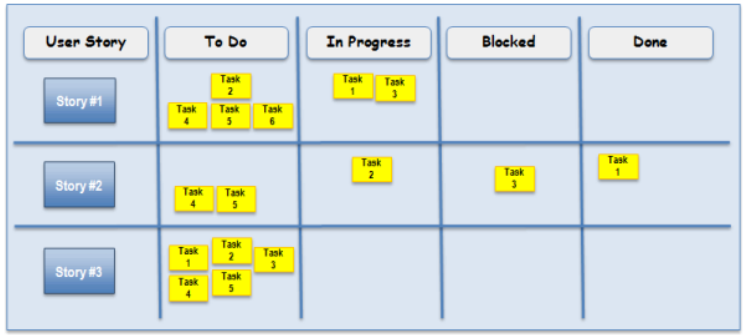
Informal --- no PowerPoint allowed

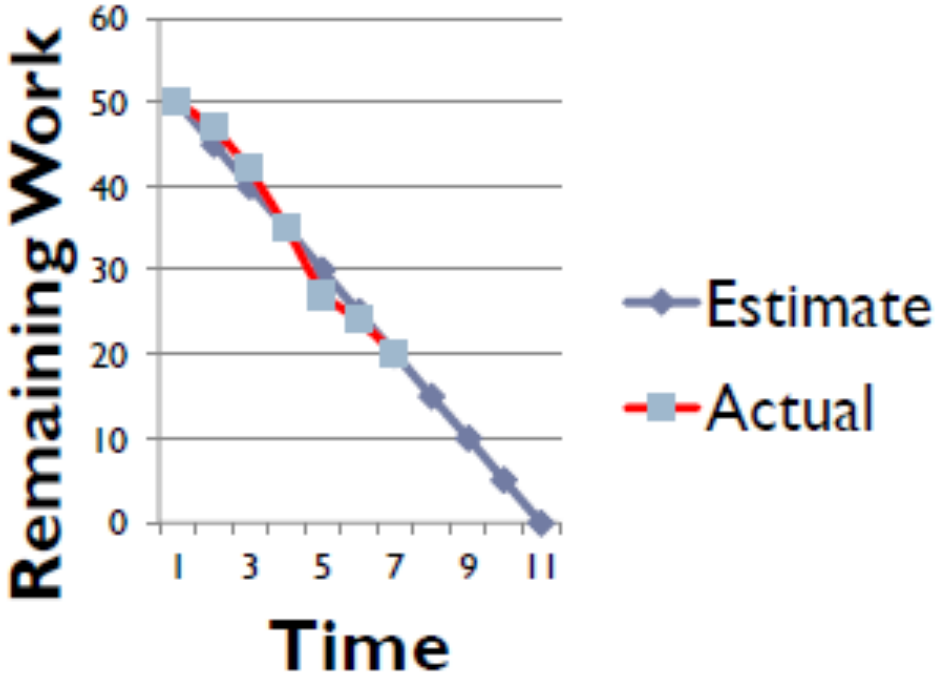
Sprint Retrospective Meeting

- Who: Development Team, Scrum Master
- **Reflect** on process
  - What are we doing well?
  - What can we improve?
  - What surprised us?
- **Improve** the process for the next sprint
- **Next**: start Sprint Planning for next Sprint
  - Which features needed most?
  - Revise the Product/Release back log

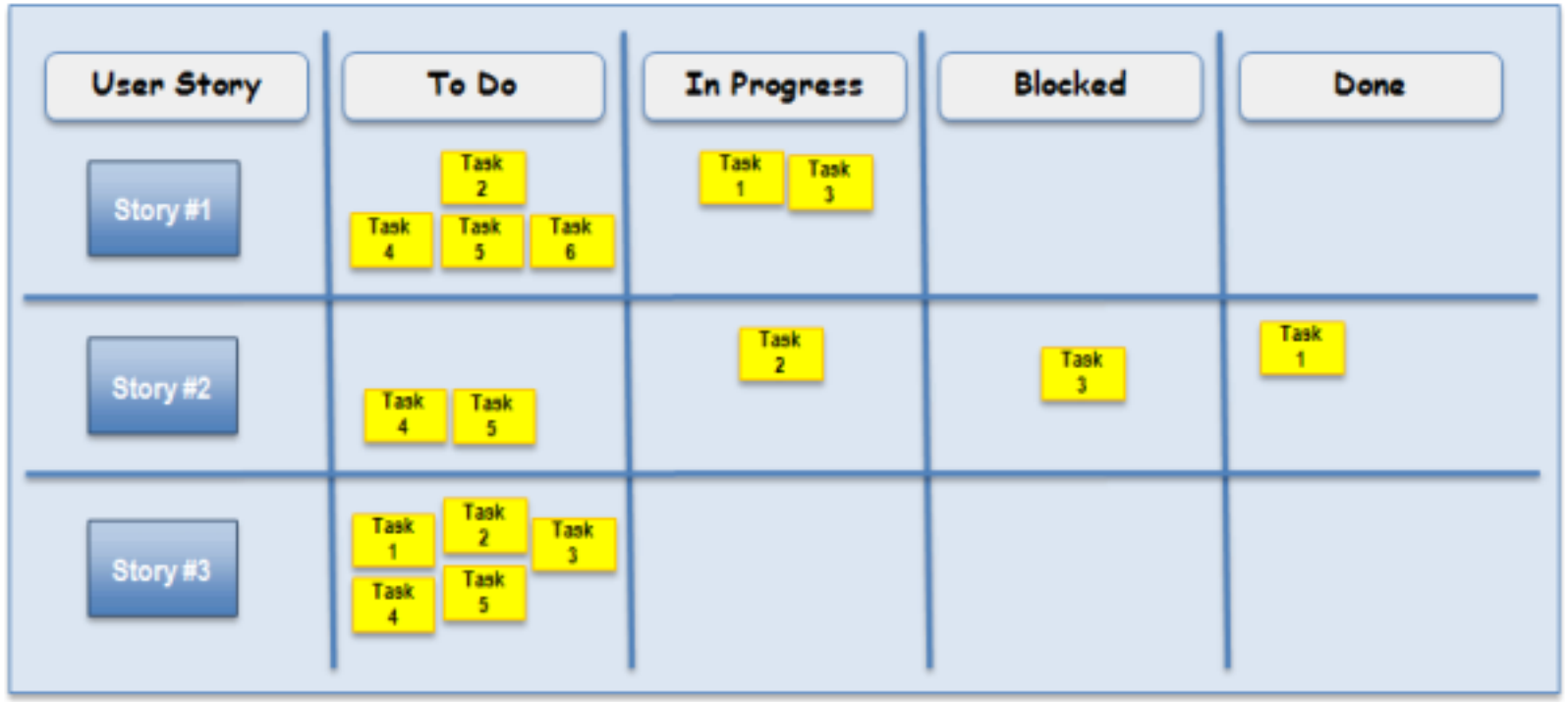
Sprint Artifacts

- Product/Release/Sprint backlog
  - Prioritized lists of desired features
- Burn Down Charts
  - Track project status





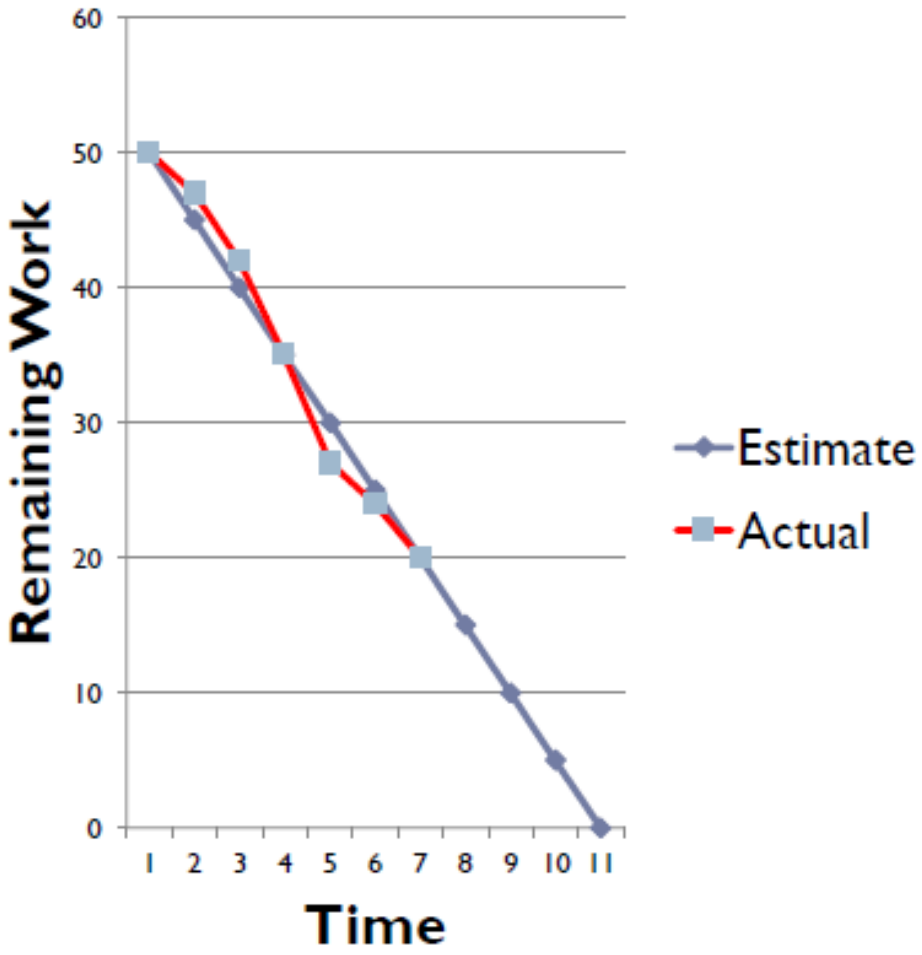
Backlogs – Tracking Progress



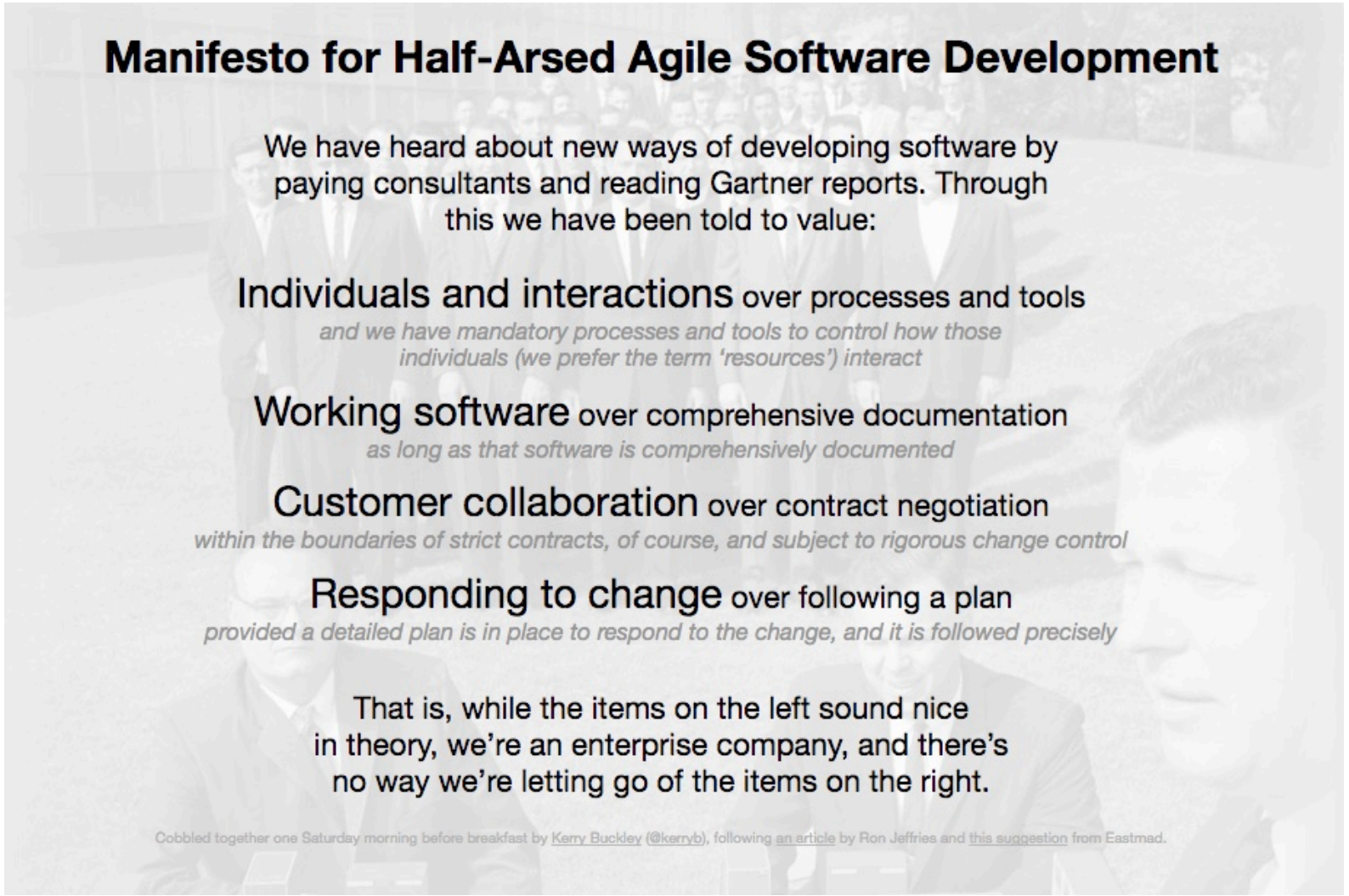
Source: https://entwickler.de/webandphp/daily-scrums-explained-125851.html

Burn Down Chart

- Graphical view of accomplishments and remaining work
- Maintained by Development Team
- May show estimated and actual values
- Slope of line is Velocity
  - Used to predict the end of project
  - Estimate when all features will be complete



What could possibly go wrong?



http://www.halfarsedagilemanifesto.org/

Committed, not just *Involved*

Product Owner, Scrum Master, and developers must be ***committed*** for success, not just ***involved***.

Scrum Bad Smells

Bad Smell	Symptom	Solution
Loss of Rhythm	Sprints with various lengths	Fixed length sprints to encourage rhythm
Talking Chickens	Daily standup meetings lose effectiveness when non-developers ask questions	Managers may listen during daily standup meetings, but only developers may ask questions
Missing Pigs	Critical people missing from the daily standup meeting	Schedule the daily standup at the same time every day and insist that all critical people attend
Unrealistic estimates	The velocity on burn down charts doesn't change from from sprint to sprint	Estimates should improve as the team has more experience working together. Learn from your experience and mistakes

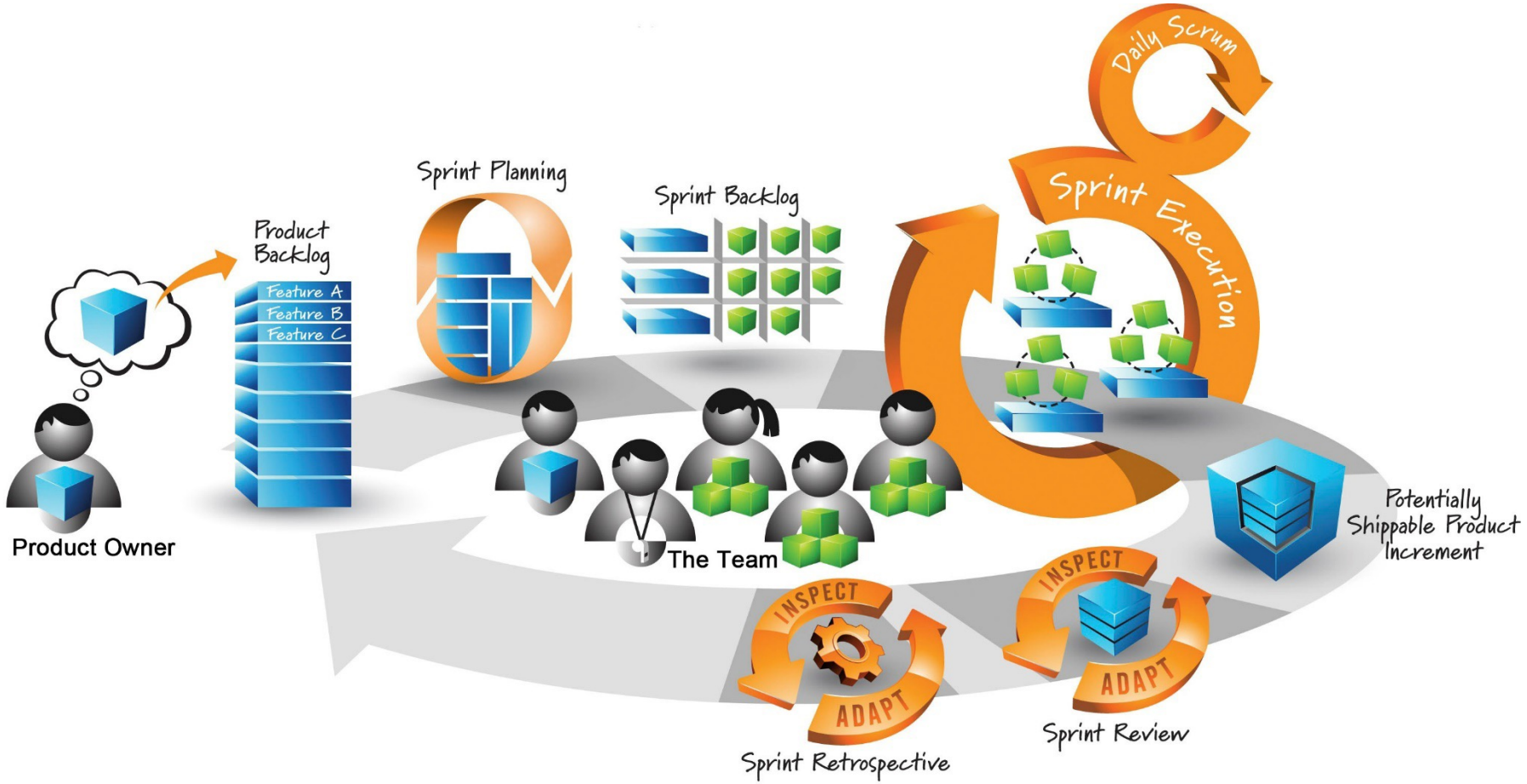
https://www.mountaingoatsoftware.com/articles/toward-a-catalog-of-scrum-smells

Scrum Bad Smells

Bad Smell	Symptom	Solution
Scrum master assigns work	Scrum master assigns tasks, rather than the developers sign up	Developers are responsible for self organization
Daily standup meeting is for the scrum master	Developers are not engaged in the standup meeting and the scrum master only tracks progress	Focus on discussion of progress and issues. Each developer should commit to team. Use guilt as a positive tool!
Specialized roles	Team members assume specialized roles, e.g. architect, tester, …	The team is responsible for overall success of the project. Everyone should do whatever is needed.

https://www.mountaingoatsoftware.com/articles/toward-a-catalog-of-scrum-smells

Scrum Process



http://www.innolution.com/essential-scrum/table-of-contents/chapter-2-scrum-framework

STEVENS
INSTITUTE OF TECHNOLOGY
1870

THANK **YOU**

Stevens Institute of Technology
1 Castle Point Terrace, Hoboken, NJ 07030