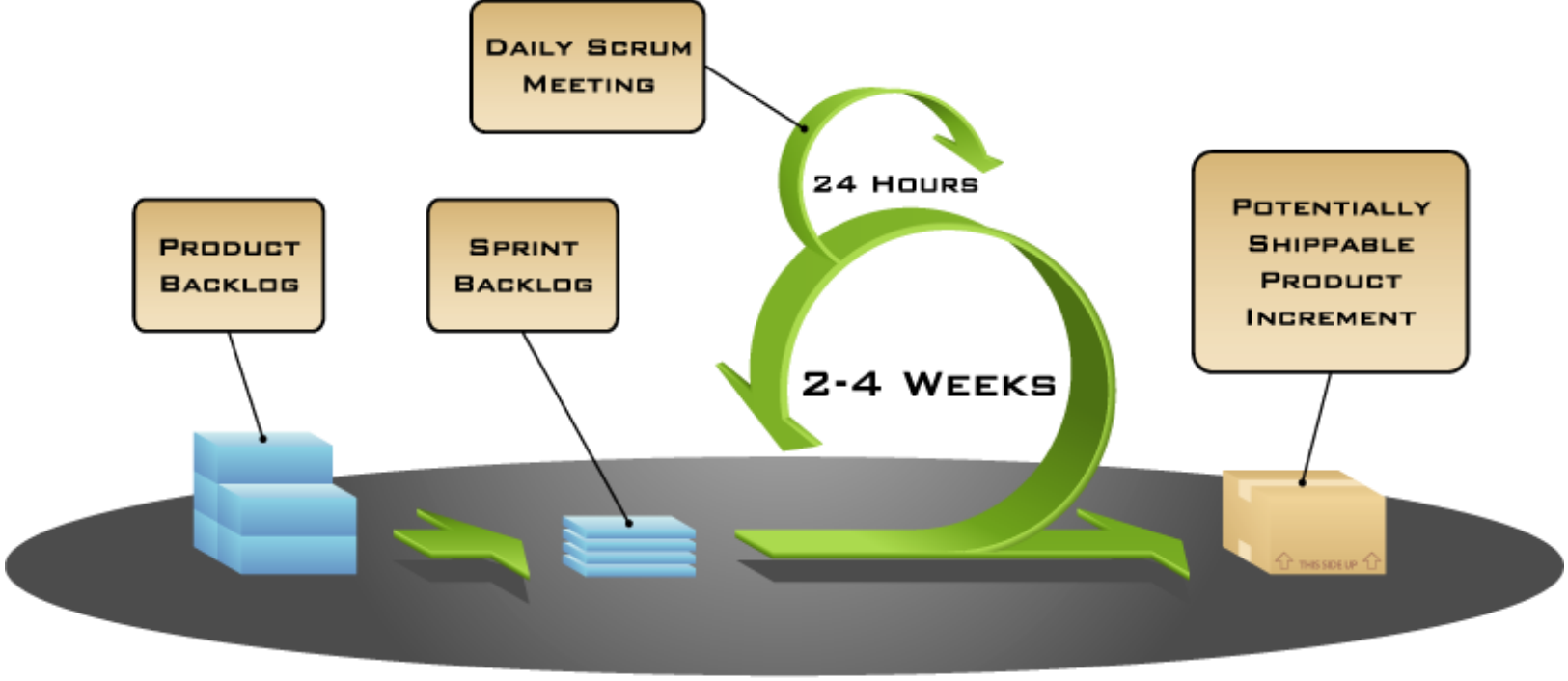


Lecture 2: An agile software development methodology (SCRUM)- Part 2

Overview of Lecture

- What is Scrum?
- A Model of SCRUM
- Scrum Roles
- Scrum Process
- Scrum Artifacts
- Advantages and Disadvantages

What is Scrum?

- Scrum is named after the game of Rugby in which a group is responsible for picking up the ball and moving it forward.
- It is an iterative, incremental process for developing any product or managing any work.
- Scrum focuses on the entire organization for its implementation to be a success.

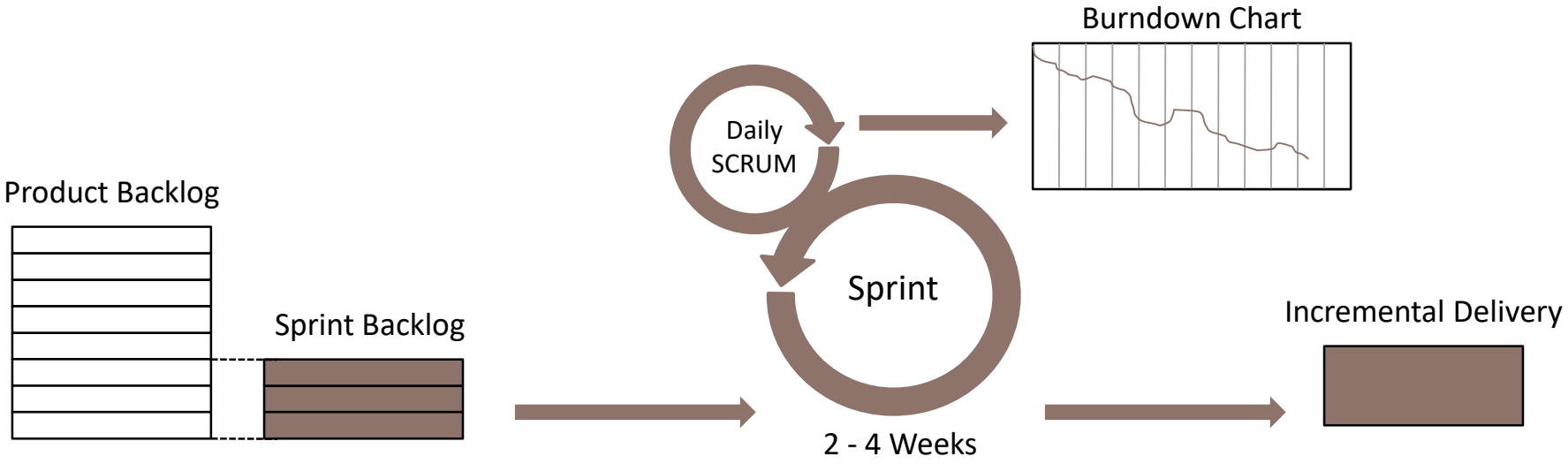
What is Scrum?

- 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.

Characteristics of Scrum

- Self-organizing teams
- Product progresses in a series of month-long "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
- One of the "agile processes"

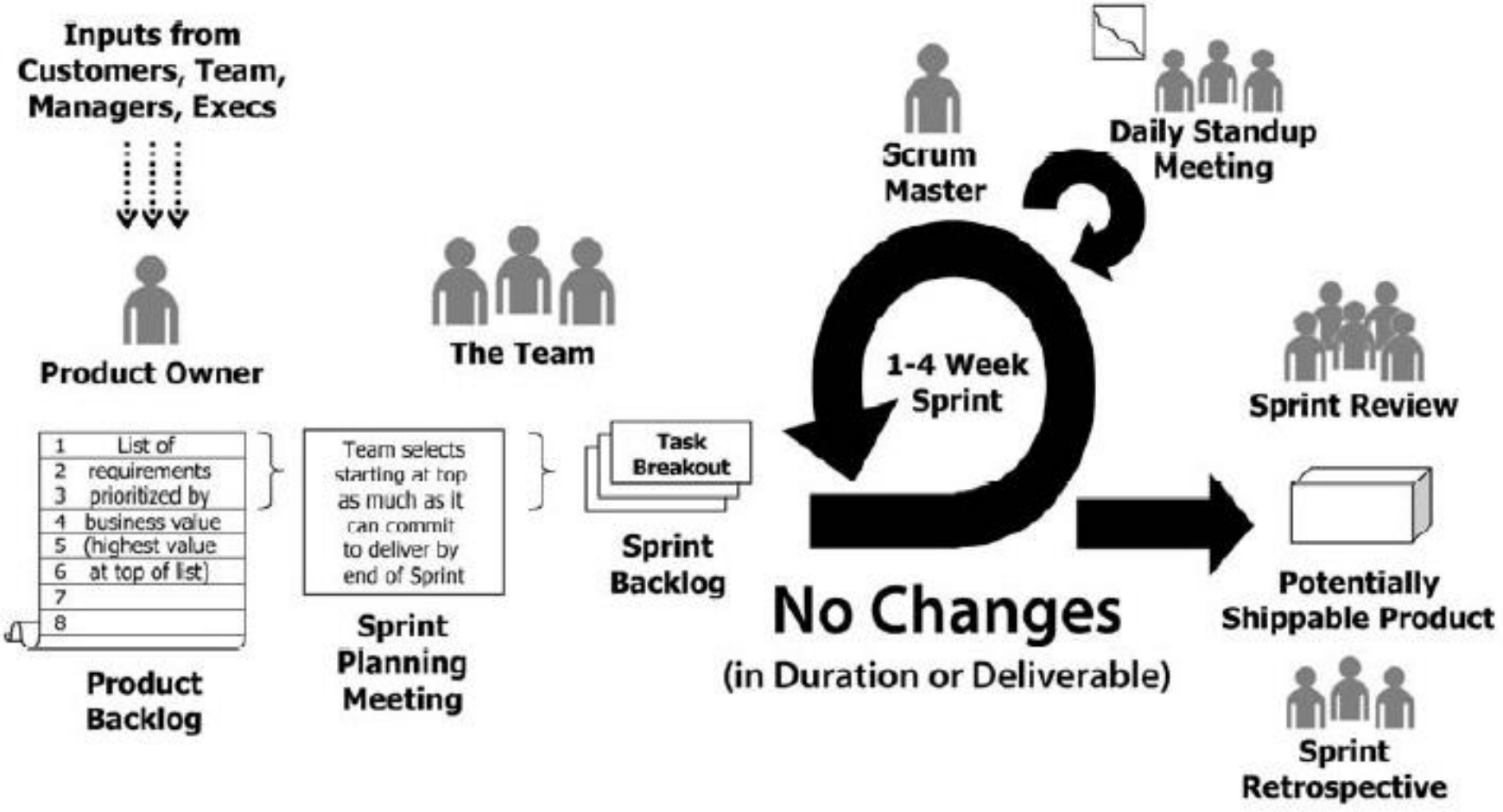
A Model of SCRUM



A Model of SCRUM

- **SCRUM has the following ELEMENTS**:
- A project team called a **SCRUM Team**
- A **Product Backlog** of all known Requirements
- A **Sprint Backlog** of requirements being worked on
- A period of work referred to as a **Sprint**
- Daily **Stand-up Meetings** with the SCRUM Team
- A **Burndown Chart** to track progress of the Sprint
- An **Incremental Delivery** at the end of each sprint

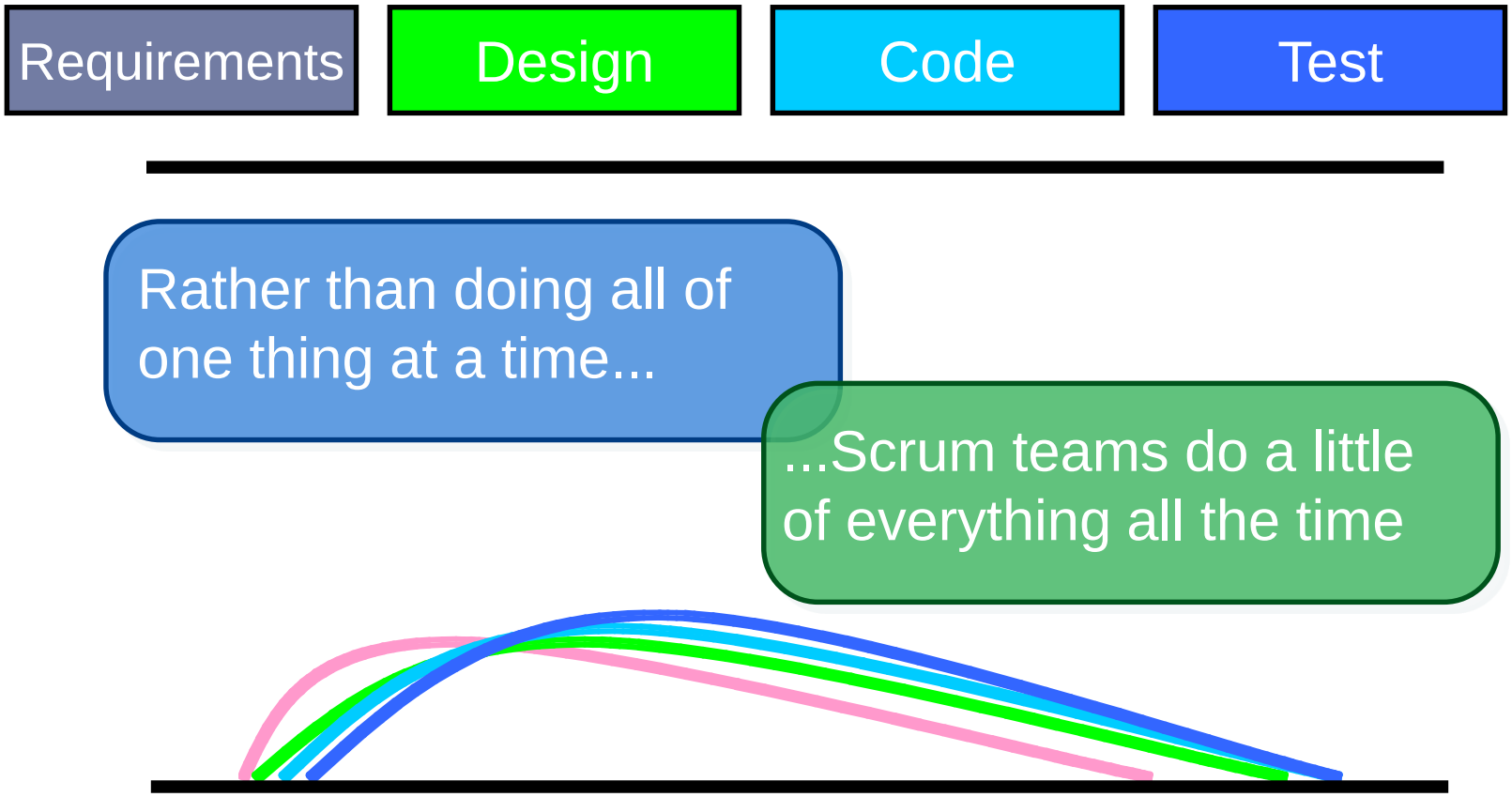
A Model of SCRUM



Sprints

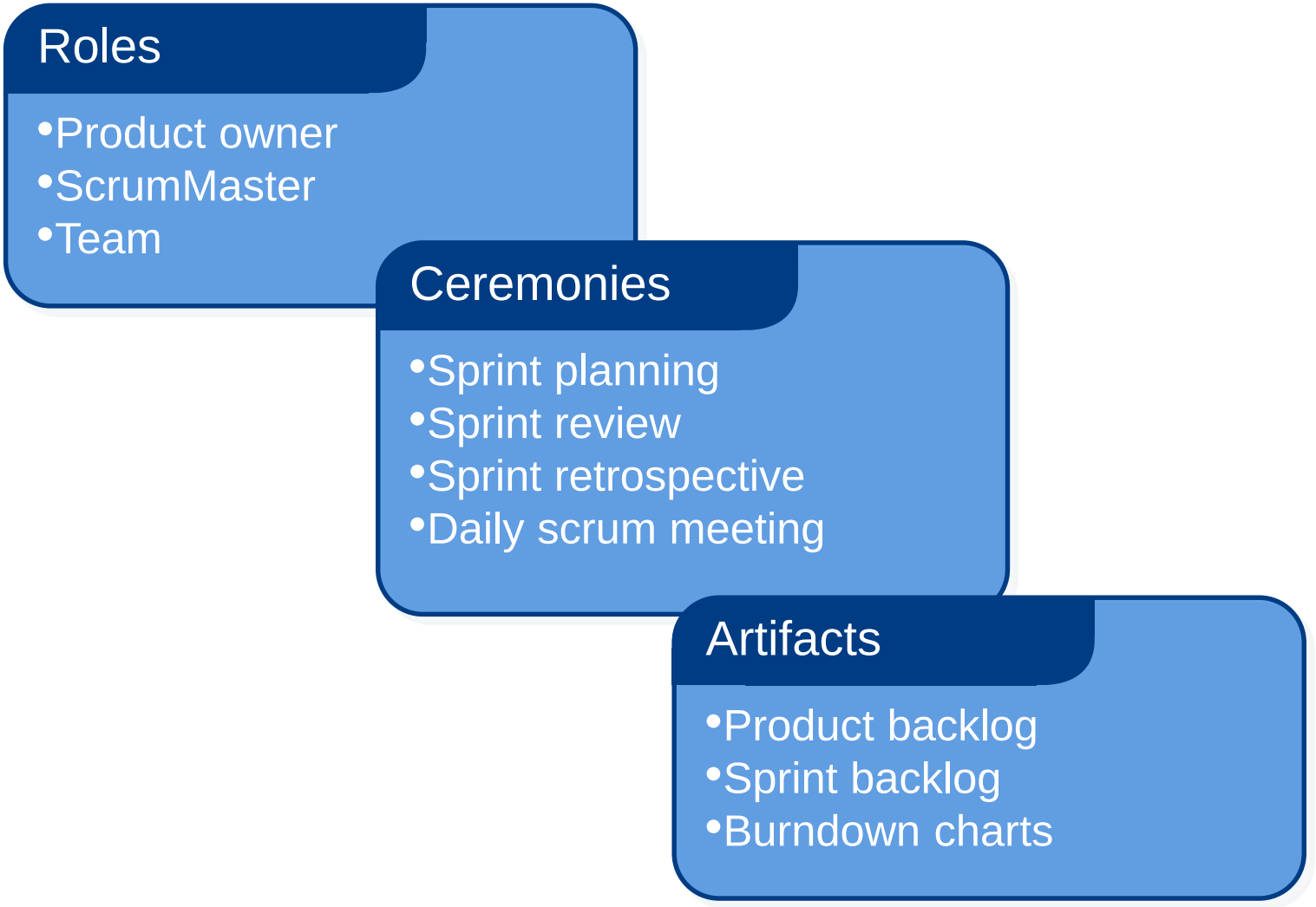
- Scrum projects make progress in a series of "sprints"
  - Analogous to Extreme Programming Iterations
- Typical duration is 2–4 weeks or a calendar month at most
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint

Sequential vs. Overlapping Development



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

Scrum framework



Components of Scrum

Scrum Roles

- Product Owner
- Scrum Master
- Team

The Process

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

Scrum Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Scrum Roles

- Scrum Master
  - Represents management to the project
  - Typically filled by a Project Manager or Team Leader
  - Responsible for enacting scrum values and practices
  - Main job is to remove impediments
  - Ensure that the team is fully functional and productive
  - Enable close cooperation across all roles and functions
  - Shield the team from external interferences

- The Scrum Team
  - Typically 5-10 people
  - Cross-functional (QA, Programmers, UI Designers, etc.)
  - Members should be full-time
  - Team is self-organizing
  - Membership can change only between sprints

Scrum Roles

- 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
  - Accept or reject work results
  - Charged with maximizing ROI and managing project risk
  - Prioritize features according to market value. Takes inputs from:
    - Customer, Team, Executives, Competitors, Other stakeholders
  - Adjust features and priority every iteration, as needed; accept or reject work results.
  - Determines release plan and communicates to all

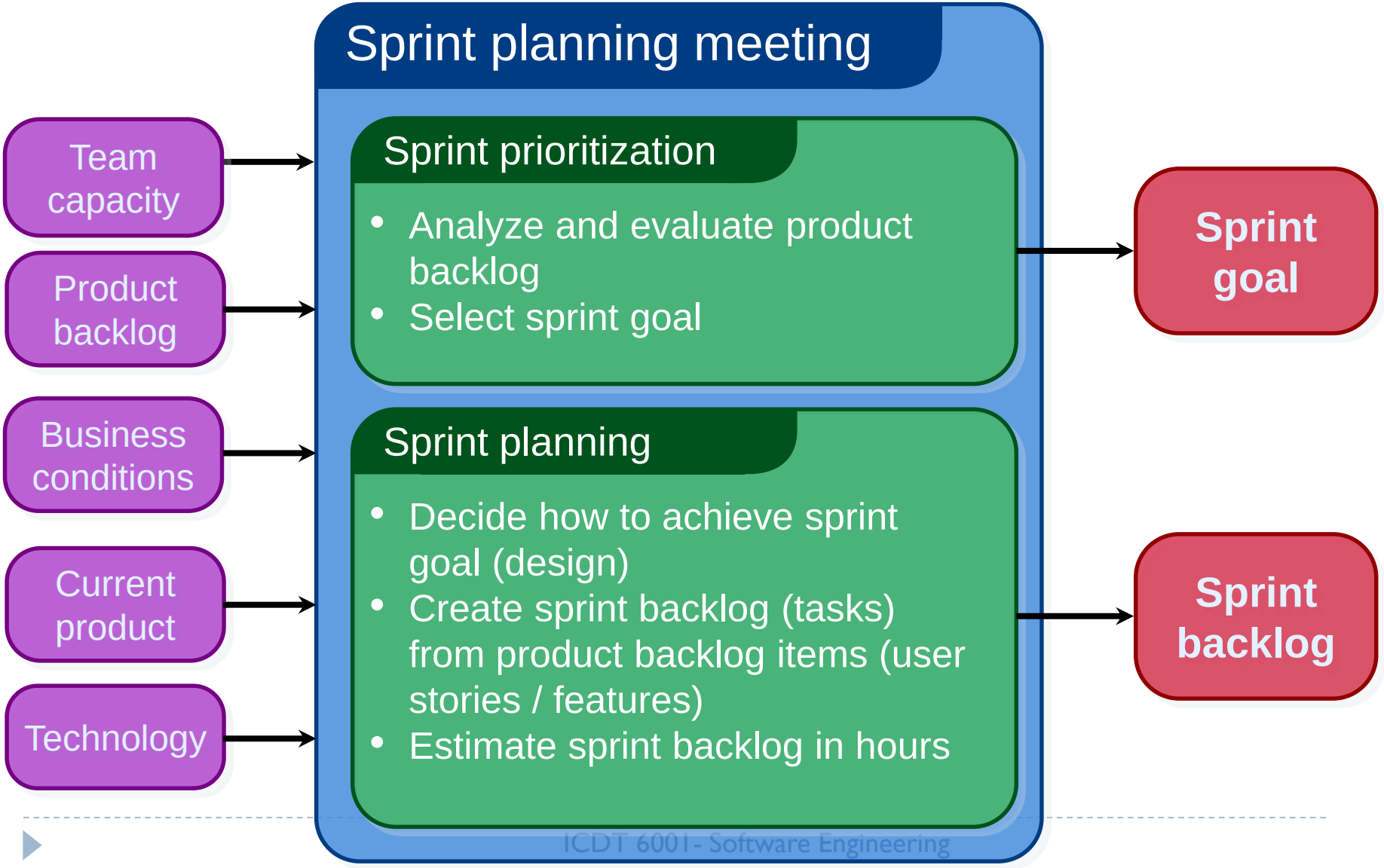
Scrum Process

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

Sprint planning

- Sprint Planning Meeting
  - A collaborative meeting in the beginning of each Sprint between the Product Owner, the Scrum Master and the Team
  - Takes 8 hours and consists of 2 parts ("before lunch and after lunch")
- 1st Part:
  - Determining the Sprint Goal
    - A short statement of what the work will be focused on during the sprint
  - Participants: Product Owner, Scrum Master, Scrum Team
- 2nd Part:
  - Participants: Scrum Master, Scrum Team
  - Creating Sprint Backlog (tasks) from product backlog items (user stories / features)

Sprint planning

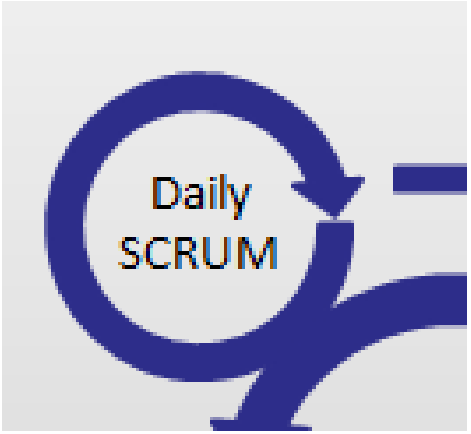


The Daily Scrum

- Is a short (15 minutes long) meeting, which is held every day before the Team starts working
- Participants: Scrum Master (is the chairman), Scrum Team
- Parameters
  - Daily, same time, same place
  - 15-minutes or less
  - Stand-up
  - Not for problem solving

The Daily Scrum

- Everyone attends – Development, Test, Product Owner
- Led by the Scrum Master (Project Manager)
- Each member quickly reviews new accomplishments, next tasks, and raises any impediments
  - Impediments resolved offline!
- Whiteboards – the more the better
- Every Team member should answer on 3 questions
  - What did you do since the last Scrum?
  - What are you doing until the next Scrum?
  - What is stopping you getting on with the work?



- Daily Scrum
  - Is NOT a problem solving session
  - Is NOT a way to collect information about WHO is behind the schedule
  - Is a meeting in which team members make commitments to each other and to the Scrum Master
  - Is a good way for a Scrum Master to track the progress of the Team

Sprint Review Meeting

- Is held at the end of each Sprint
- Business functionality which was created during the Sprint is demonstrated to the Product Owner
- Informal, should not distract Team members of doing their work
- 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

Sprint Retrospective

- Periodically take a look at what is and is not working
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
  - ScrumMaster, Product owner, Team, Possibly customers and others
- Objective:
  - What's working
  - What's could work better
  - Things to try in the next Sprint
- Why does the Retrospective matter?
  - Accelerates visibility
  - Accelerates action to improve
  - It's a key mechanism of continuous improvement
  - If Scrum is an engine, Retrospective is the "oil filter"
    - Catches the debris and keeps it from recirculation
    - Keeps the engine running clean and smooth

Scrum Artifacts

- Product Backlog
- Sprint Backlog
- Burn down Charts

Product Backlog

- Requirements for a system, expressed as a prioritized list of Backlog Items
- Is managed and owned by a Product Owner
- Spreadsheet (typically)
- Usually is created during the Sprint Planning Meeting
- Can be changed and re-prioritized before each Sprint
- List of everything that could ever be of value to the business for the team to produce
  - Ranked in order of priority
  - Priority is a function of ROI and risk
- Product Owner can make any changes they want before the start of a Sprint Planning Meeting
  - Items added, changed, removed, reordered

Product Backlog

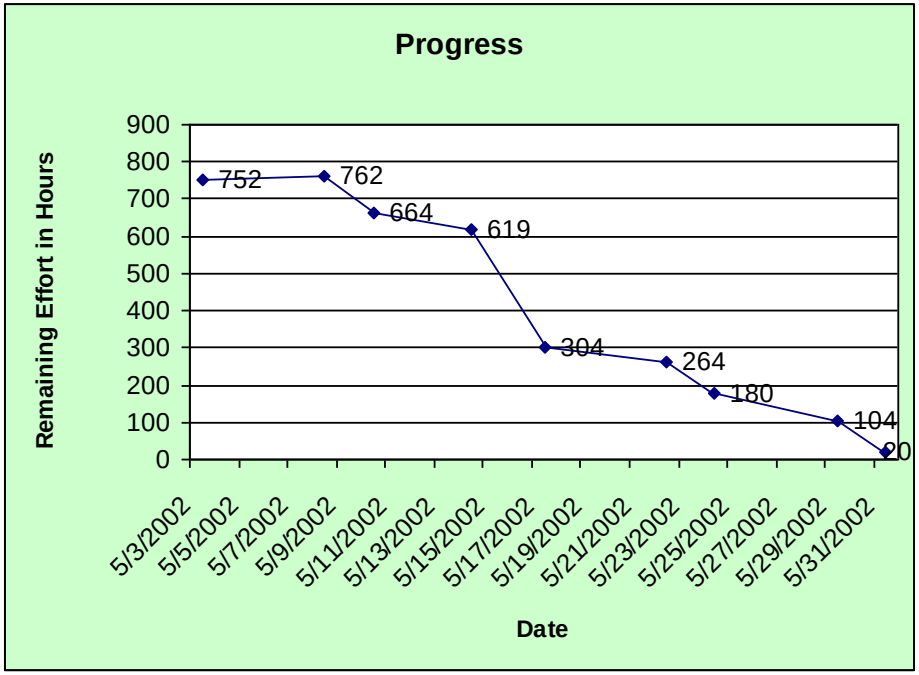
- Estimation of Product Backlog Items
  - Establishes team's velocity (how much Effort a Team can handle in one Sprint)
  - Determining units of complexity.
    - Size-category ("T-Shirt size")
    - Story points
    - Work days/work hours
  - Methods of estimation:
    - Expert Review
    - Creating a Work Breakdown Structure (WBS)
- Is only a FORECAST!-> is not exact

Sprint Backlog

- A subset of Product Backlog Items, which define the work for a Sprint
- Is created ONLY by Team members
- Each Item has it's own status
- Should be updated every day
- No more then 300 tasks in the list
- If a task requires more than 16 hours, it should be broken down
- Team can add or subtract items from the list. Product Owner is not allowed to do it

Burn down Charts

- Are used to represent "work done".
- Are wonderful Information Radiators
- 3 Types:
  - Sprint Burn down Chart (progress of the Sprint)
  - Release Burn down Chart (progress of release)
  - Product Burn down chart (progress of the Product)
- X-Axis: time (usually in days)
- Y-Axis: remaining effort



Burn down Charts

- Sprint Burn down Chart
  - Depicts the total Sprint Backlog hours remaining per day
  - Shows the estimated amount of time to release
  - Ideally should burn down to zero to the end of the Sprint
  - Actually is not a straight line
  - Can bump UP
- Release Burn down Chart
  - Will the release be done on right time?
  - X-axis: sprints
  - Y-axis: amount of hours remaining
  - The estimated work remaining can also burn up
- Product Burn down Chart
  - Is a "big picture" view of project's progress (all the releases)

Advantages of Scrum

- Completely developed and tested features in short iterations
- Simplicity of the process
- Clearly defined rules
- Increasing productivity
- Self-organizing
- Each team member carries a lot of responsibility
- Improved communication
- Control of very complex process of product development
- Allows Developers to focus on delivering a usable functionality to the client
- Generates productivity improvements by implementing a framework that empowers teams and thrives on change
- Insists that the Client prioritize required functionality.
- Ability to respond to the unpredictable in any project requirements.
- Flexibility
- Knowledge sharing between Developers
- Collective ownership

Disadvantages of Scrum

- Scrum is not effective for small projects
- Expensive to implement
- Training is required
- Makes all dysfunction visible
  - Scrum doesn't fix anything: the team has to do it
  - May feel like things are worse at the beginning
- Bad products will be delivered sooner
- Doomed projects will fail faster
- High risk of turnover
  - Some people will refuse to stay on a Scrum team
- Partial adoption may be worse than none at all

Conclusion

- Scrum offers:
  - a high degree of flexibility
  - promises a high probability of success
- Scrum benefits:
  - an anticipating culture
  - increases the sense of urgency
  - promotes the sharing of knowledge
  - encourages dense communications
  - facilitates honesty among developers

References and Links

- Schwaber, Ken and Mike Beedle. *Agile software Development with Scrum*. Prentice Hall, 2002.
- Sutherland, Jeff. "Inventing and Reinventing Scrum in five companies", 21 September 2001
- www.controlchaos.com/scrumwp.htm
- www.agilescrum.com/
- www.mountaingoatsoftare.com/scrum/
- www.objectmentor.com
- agilealliance.com/articles/articles/InventingScrum.pdf
- jeffsutherland.com/
- www.mountaingoatsoftware.com/scrum
- www.scrumalliance.org