31269: Business Requirements Modelling

Week 11 Lecture : Agile System Development

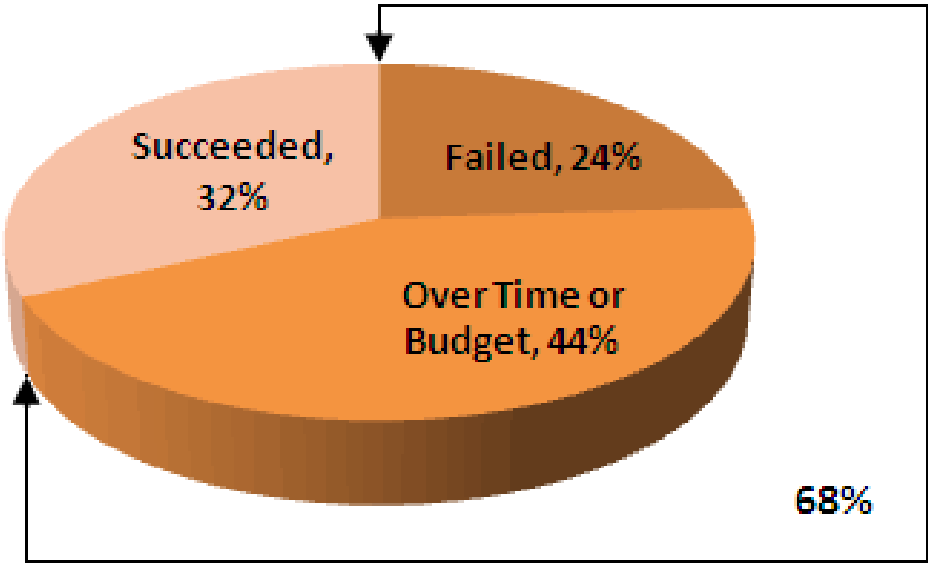
Objectives

- Last Week
  - State Transition Diagram (we finished the series of Object Oriented Models)
- This Week
  - Software Crisis
  - Software Processes
  - Process Crisis
  - Agility
  - **Agile Methodologies**

Software Crisis

- June 1994
- IBM consulting group released a study of 24 leading companies that had developed large distributed systems. They concluded:
  - 55% of the projects cost more than expected
  - 68% overran their schedules and
  - 88% required substantial redesign

Software Crisis



Standish Group's 2009 CHAOS Report

Software Crisis

- The California Department of Motor Vehicles (DMV) in the USA
  - Started in 1987 and in 1994, after struggling over a period of seven years with the project and spending $44.3 million it was declared a failure
- CONFIRM Project
  - A joint effort between Hilton, Marriott, and Budget Rent-A-Car (initiated in 1988) was terminated in 1992, after spending $125 million.
  - The consultant firm AMRIS in and out of court settlement paid $160 million.
- Project “Trawlerman” of the Ministry of Defense
  - In Nov 1996 the project was abandoned – Writing off £41 million

Solution ?

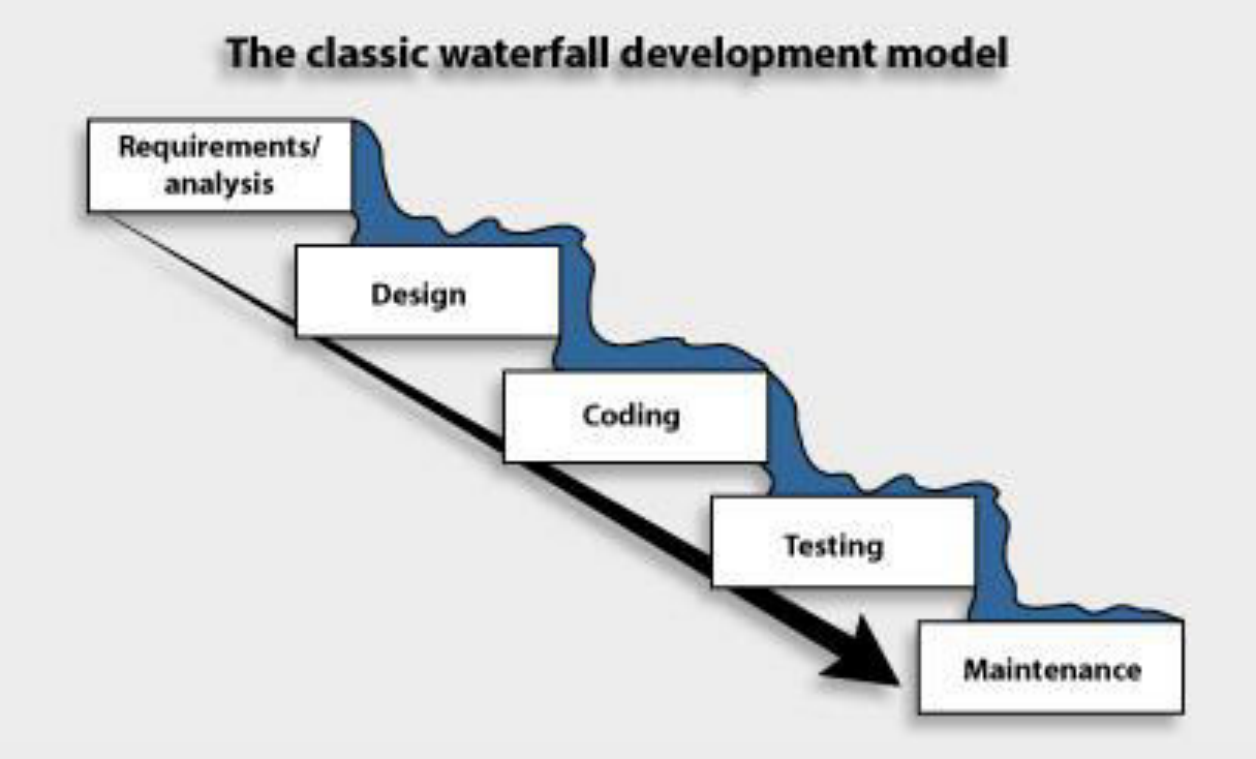
- NATO Conference…..
- “Software Engineering” was coined as a term.
- The idea was to *engineer* software rather to *develop* software.
- Software Development vs Software Engineering

Software Engineering

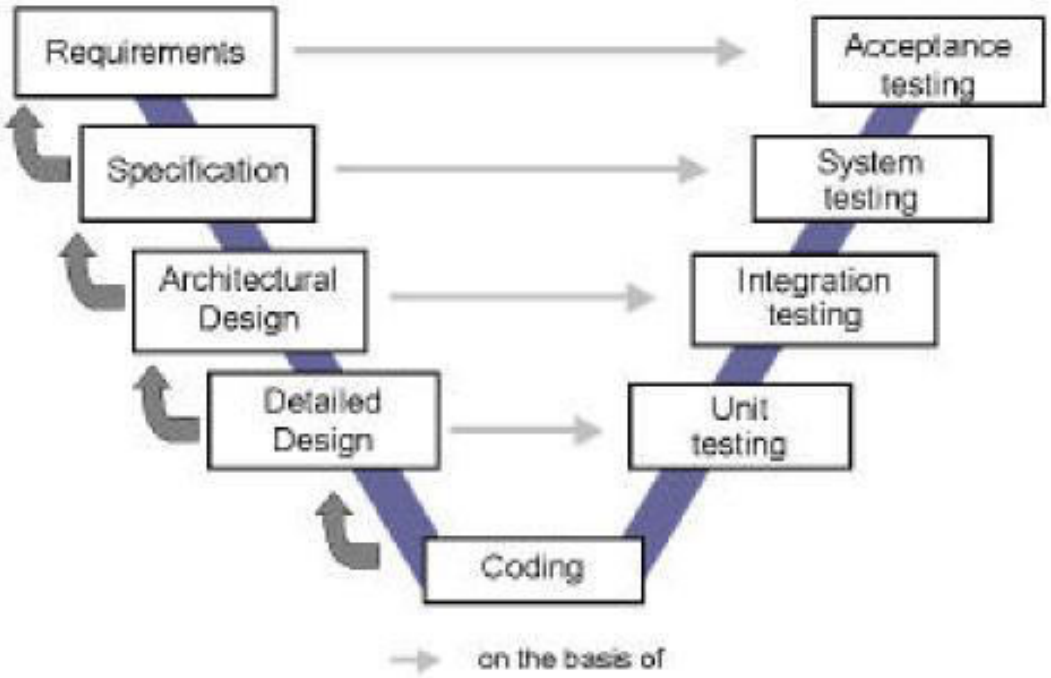
- A systematic, disciplined and quantifiable approach to the development, operation and maintenance of software.
- Applying engineering principles to the development of software.

Software Processes

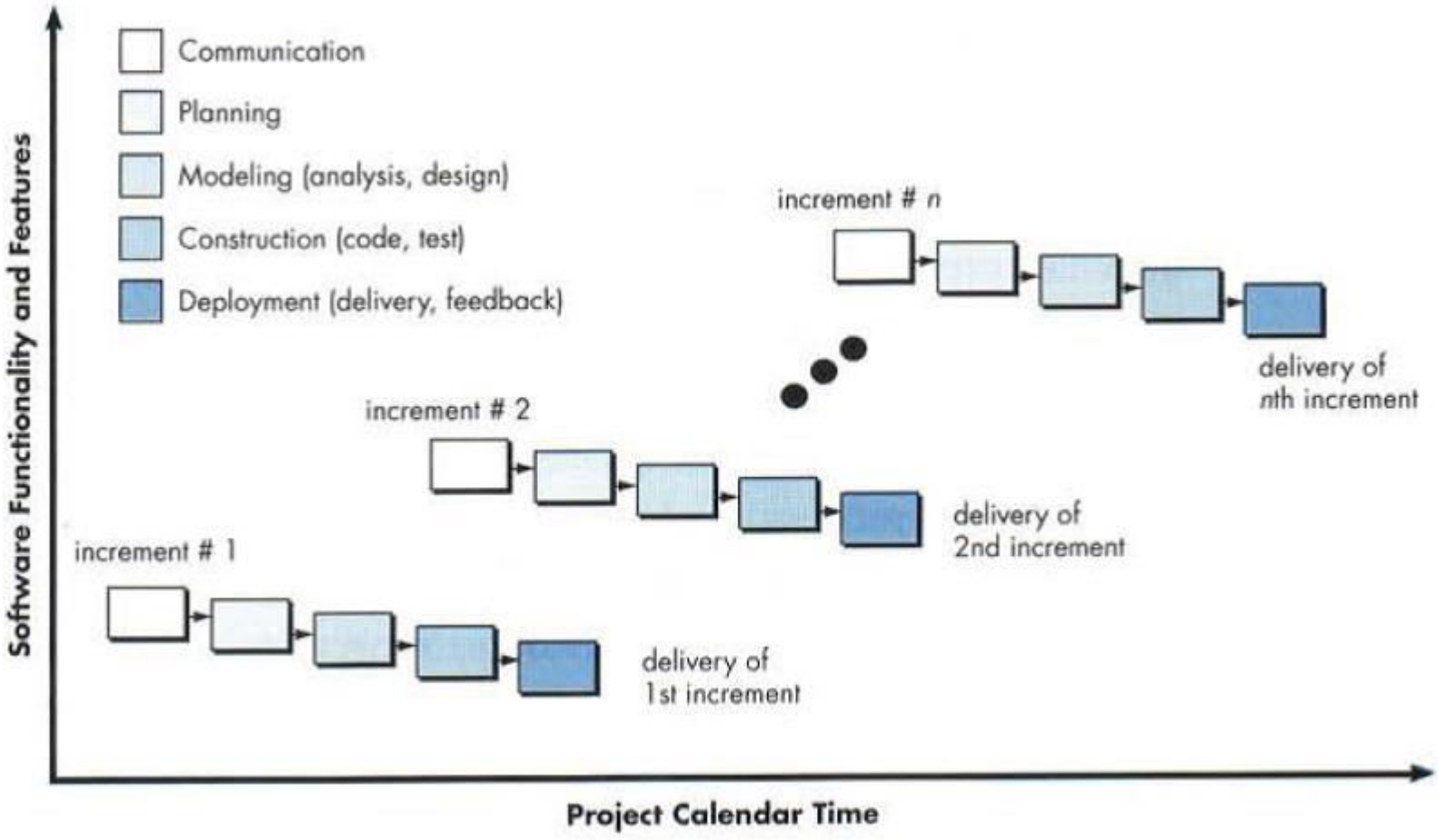
- Many software processes have been introduced since then…



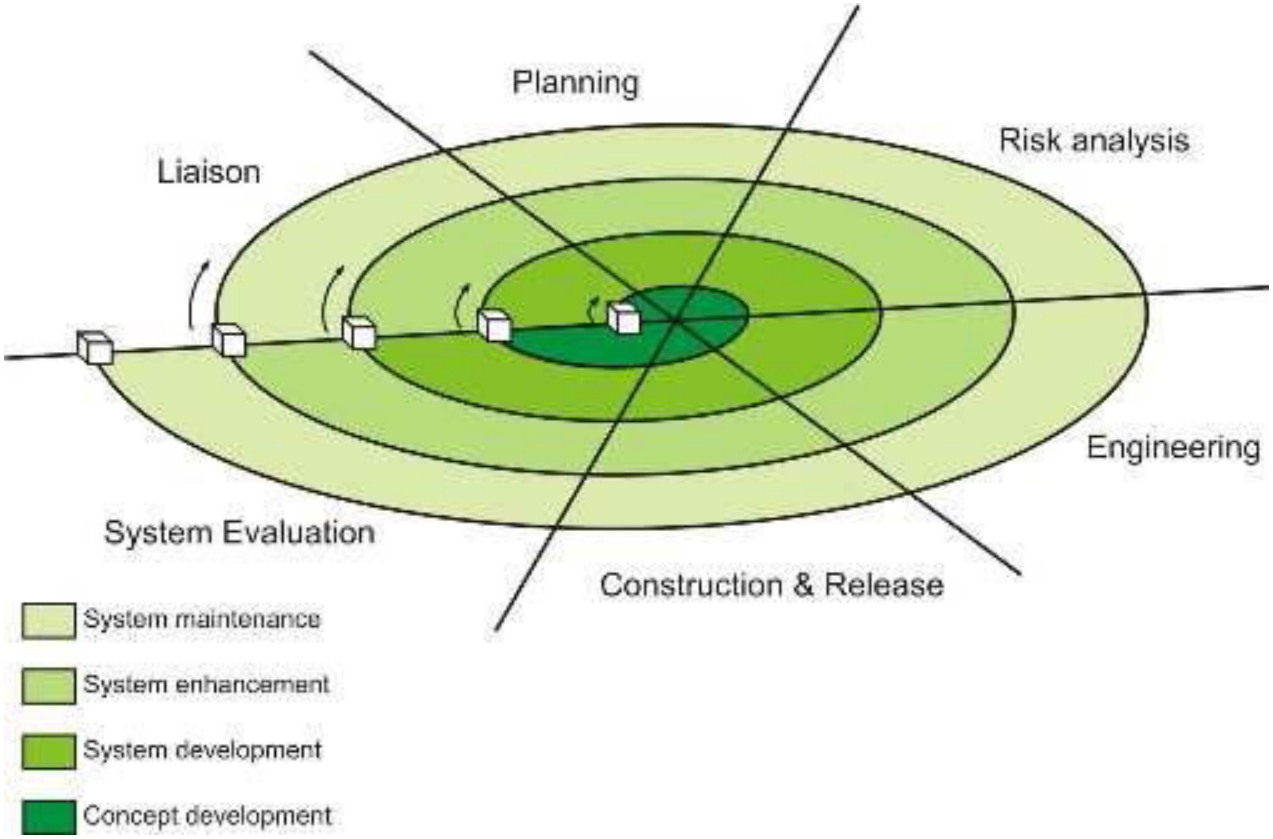
V-Model


Requirements
Specification
Architectural Design
Detailed Design
Coding
Unit testing
Integration testing
System testing
Acceptance testing
on the basis of

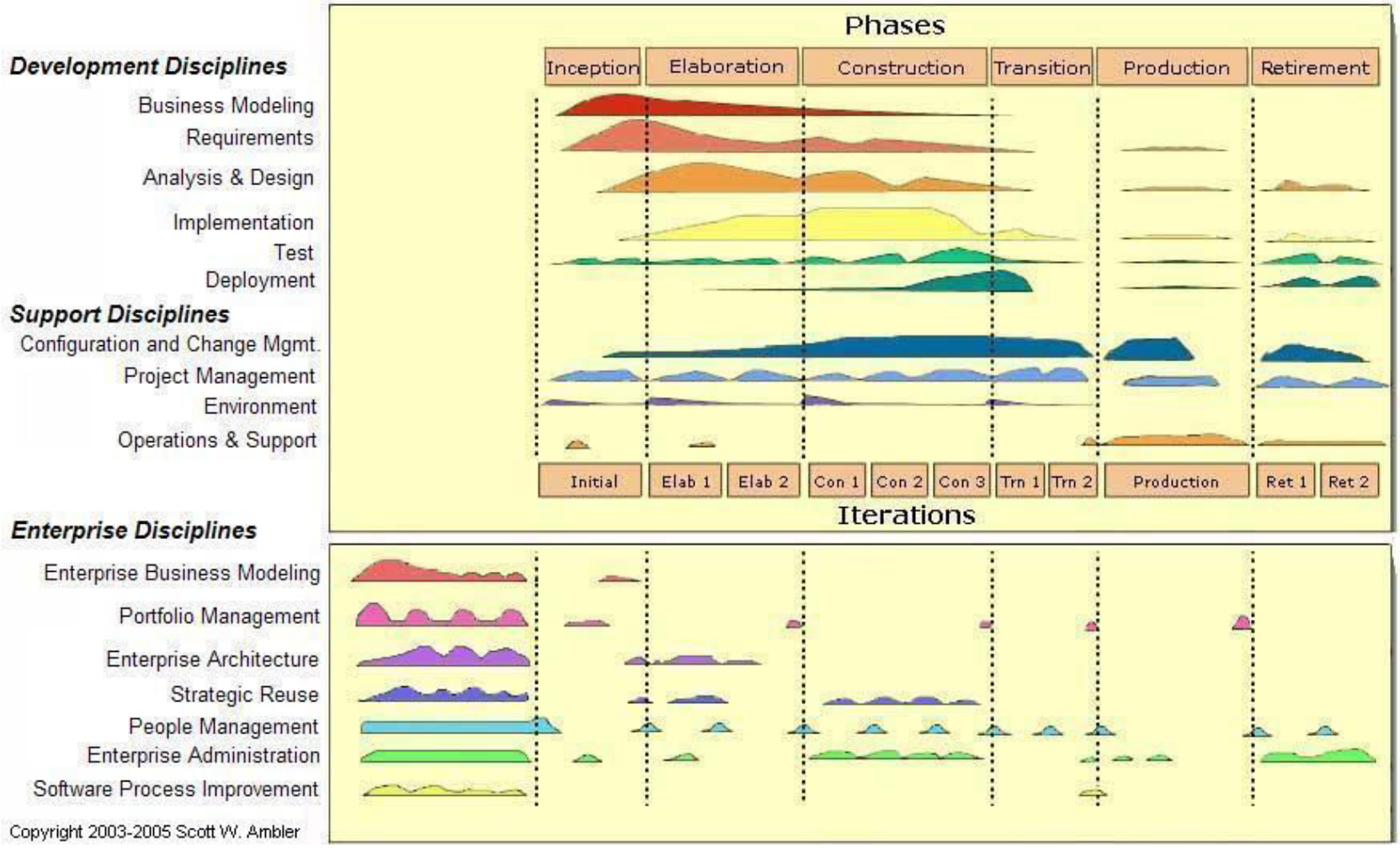
Incremental Process



Spiral Process



Rational Unified Process (RUP)



Process Crisis….

- The processes became so
  - Large – Heavy Weight
  - Formal, rather complex
- For example, RUP has
  - More than 40 roles
  - More than 100 artifacts

Process Crisis….

- The processes became so
  - Rigid - difficult to change
- We can’t change processes for
  - Different projects
    - Different scope and requirements
  - Different domains
    - E.g. Information systems – medical systems – real time systems

Software Crisis

Software Processes

Process Crisis

?

Solution

- Process should be light weight
- Process should be easy to change
- Process should be……
- Agile

Agile

- Software development methods based on similar principles.
  - Iterative
  - Incremental
- The Agile framework was introduced in 2001 and was named “Agile Manifesto”

Agile Manifesto - Values

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

1) Individuals and interactions over processes and tools.

2) Working software over comprehensive documentation.

3) Customer collaboration over contract negotiation.

4) Responding to change over following a plan.

Agile Software Development

- Promotes
  - adaptive planning
  - evolutionary development
  - early delivery
  - continuous improvement
  - encourages rapid and flexible response to change.

Twelve Agile Manifesto Principles

- Customer satisfaction through early and continuous delivery
- Welcome changing requirements, even late in development
- Working software is delivered frequently (weeks rather than months)
- Close, daily cooperation between business people and developers
- Build projects around motivated individuals
- Face-to-face conversation is the best form of communication (co-location)
- Working software is the principal measure of progress
- Agile processes promote sustainable development
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- Self-organizing teams
- Regular adaptation to changing circumstance

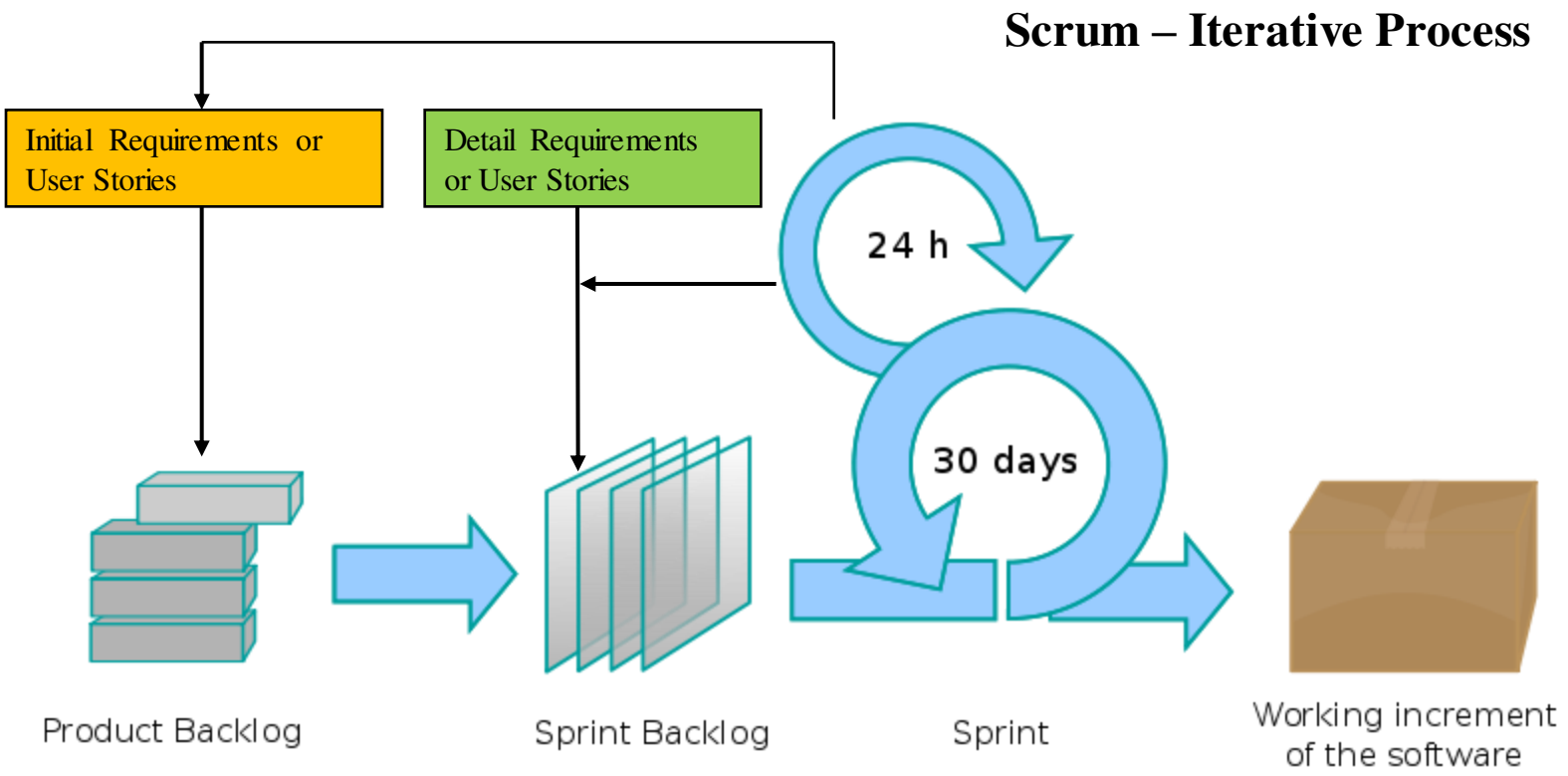
Agile Methodologies

- SCRUM
- Extreme Programming
- Feature Driven Development
- Test Driven Development
- Lean
- Crystal
- ….

Scrum has been used by:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce

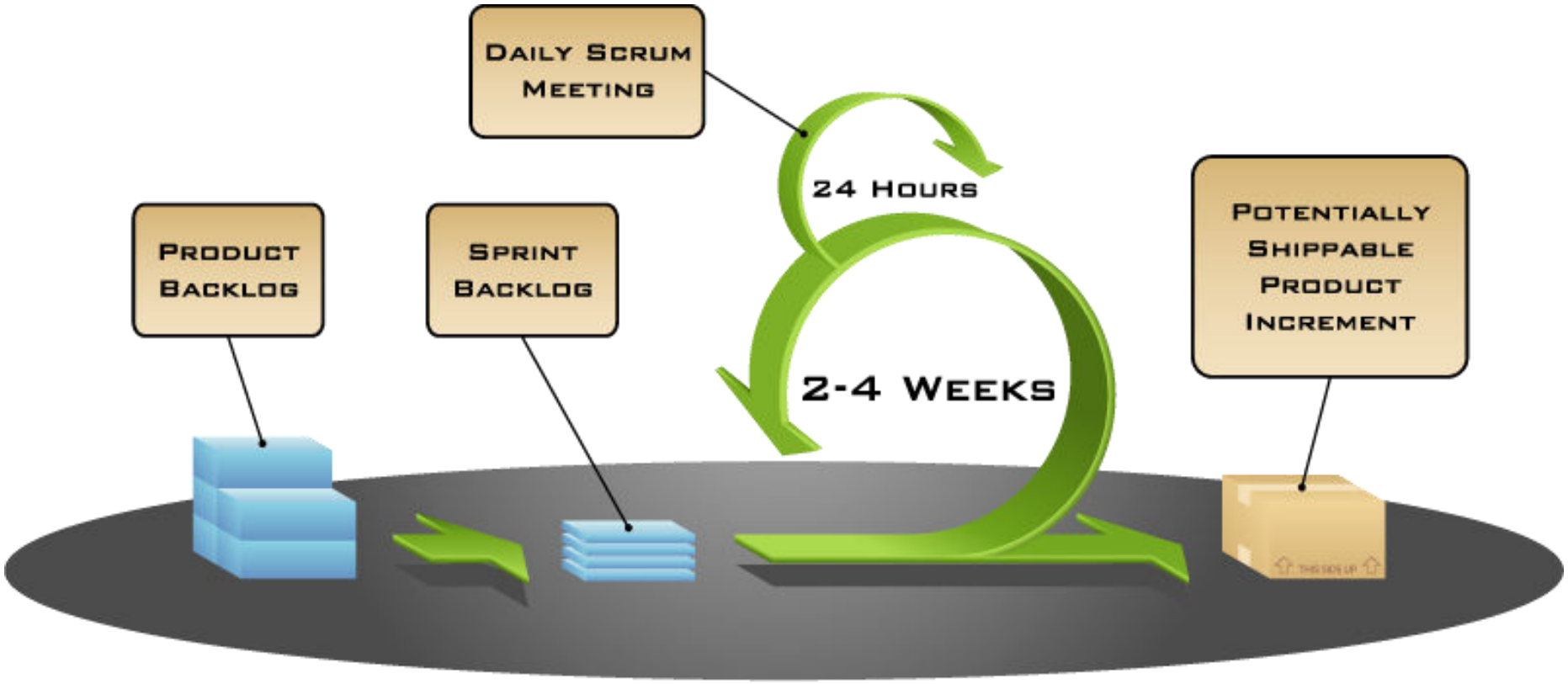
Agile System Development Process



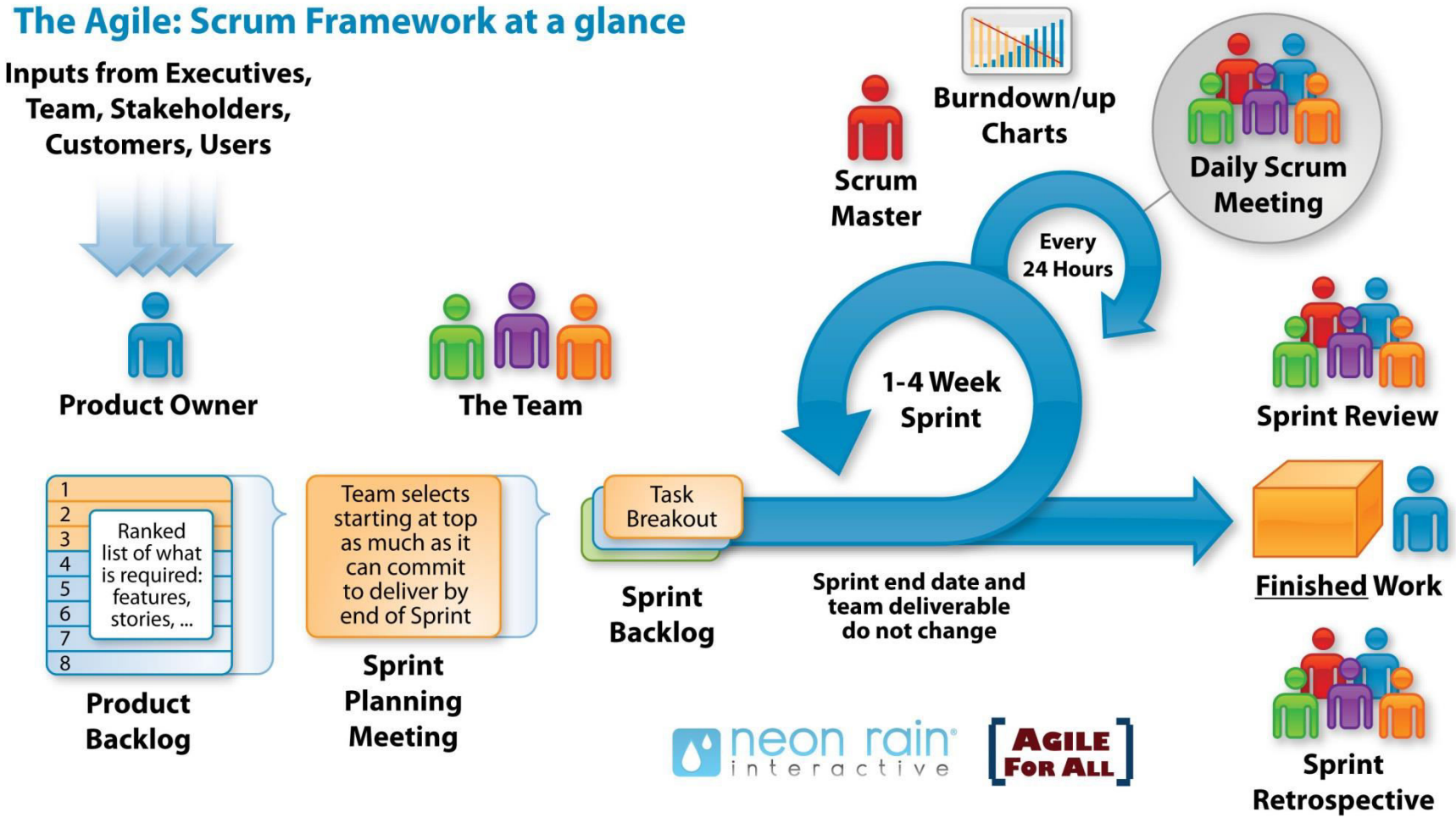
source: http://en.wikipedia.org/wiki/Scrum_(development)

- To understand Scrum in under 10 minutes, please watch the video available at http://www.youtube.com/watch?v=XU0llRltyFM

SCRUM at a glance



SCRUM at a glance



https://eternalsunshineoftheismind.files.wordpress.com/2013/02/scrum_process_big.jpg

Scrum Diagram:

1 Product Backlog

The Product Backlog contains a wish list of all the User Stories of a product.

2 Release Backlog

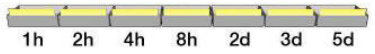
The goal of a given release is to deliver a subset of the Product Backlog, known as the Release Backlog.

3 Sprint Backlogs

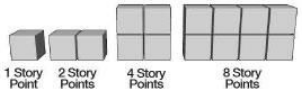
Each Sprint (or short duration milestone) takes a small chunk of the Release Backlog and gets it Ship-Ready!

Estimation Techniques:

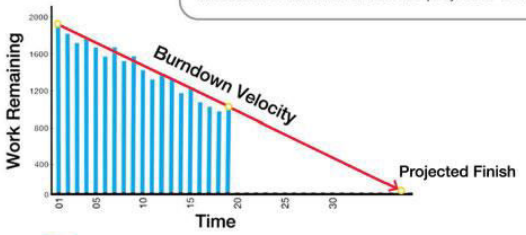
Hours: When estimating work, it's important to have the entire team use some standards. Use 1h, 2h, 4h, 8h, etc. No estimations in-between.



Story Points: You can also estimate work in comparison to the complexity of a well known but simple component.



Remember that in Scrum, the estimates are only part of the story. The Burndown Velocity is the true indicator of whether or not the project is on track.



4 Burndown Chart

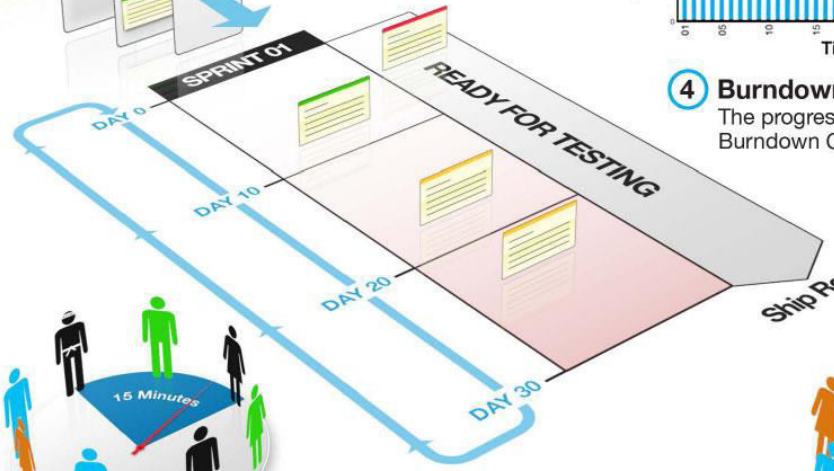
The progress of the team is monitored using a Burndown Chart.

Team Roles:

Product Owner: Is responsible for what goes into the product backlog and prioritizes it. Would probably make a good dictator if given the chance.

Scrum Master: A team facilitator. Ensures teams have what they need to get the job done. Also, sets up meetings and monitors everything. Also, kicks ass when necessary.

Developers & Testers: They write code and make sure it does what it's suppose to do. Duh!



5 Daily Scrum Meetings

Short daily standup meetings ensure everything is on track and everyone has the tools they need.

6 Sprint Retrospective

After each sprint, a longer retrospective meeting helps fine-tune the process.

http://www.asg.cc/Scrum.html

Sprints

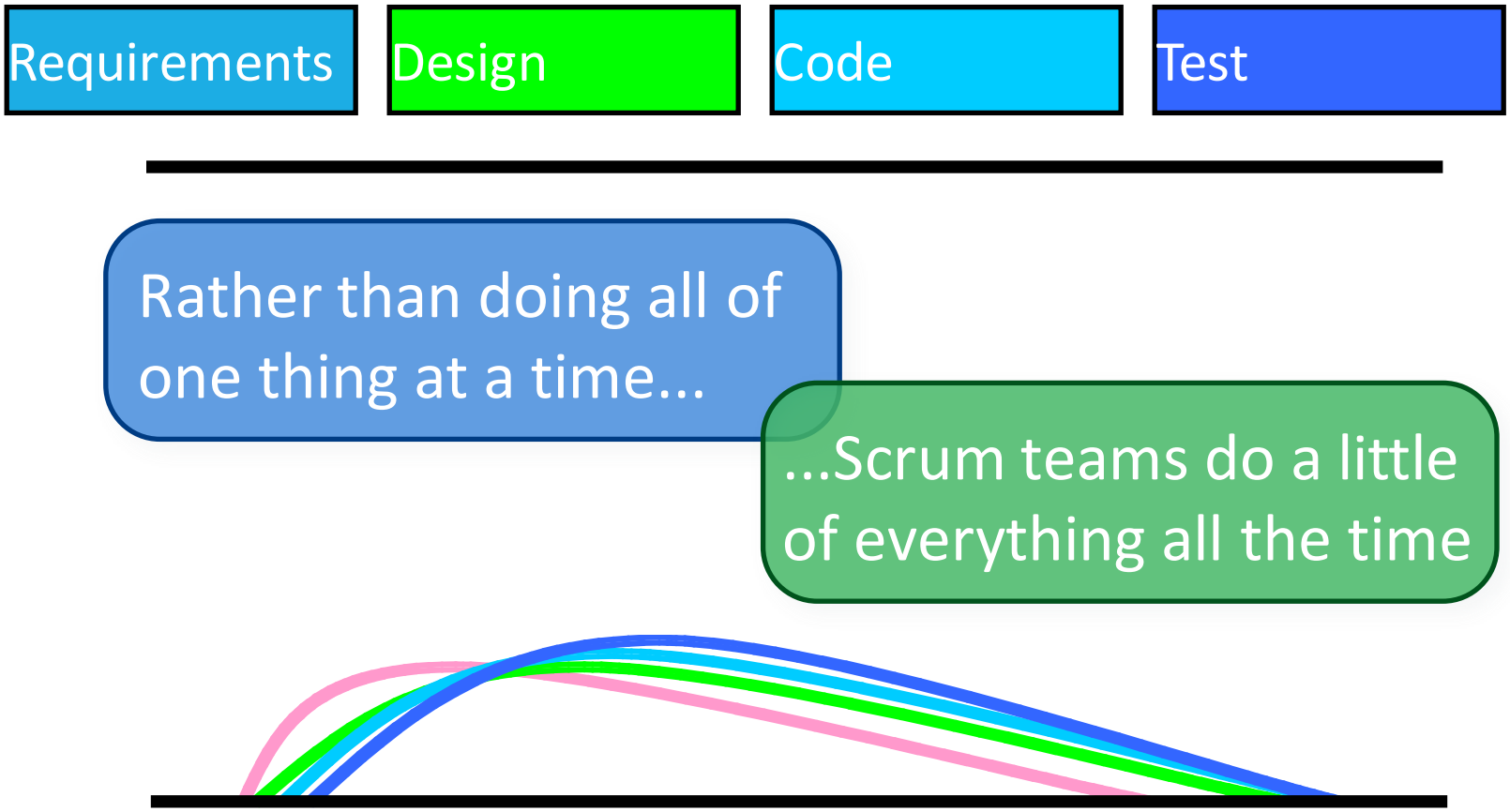
- A Sprint (or iteration) is the basic unit of development in Scrum. Scrum projects make progress in a series of "sprints".
  - Analogous to Extreme Programming iterations
- Typical duration is **2–4 weeks** or a calendar month at most
- Product is designed, coded, and tested during the sprint.
- Each sprint is started by a **planning meeting**, where the tasks for the sprint are identified and an estimated commitment for the sprint goal is made, and ended by a sprint **review-and-retrospective meeting**, where the progress is reviewed and lessons for the next sprint are identified.

Iterative and incremental development

- The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system.
- Learning comes from both the development and use of the system, where possible key steps in the process start with a simple implementation of a **subset of the software requirements** and iteratively enhance the evolving versions until the full system is implemented.
- **Incremental** development slices the system functionality into increments (portions). **In each increment, a slice of functionality is delivered** through cross-discipline work, from the requirements to the deployment.

(http://en.wikipedia.org/wiki/Iterative_and_incremental_development)

Sequential vs. overlapping development



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

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

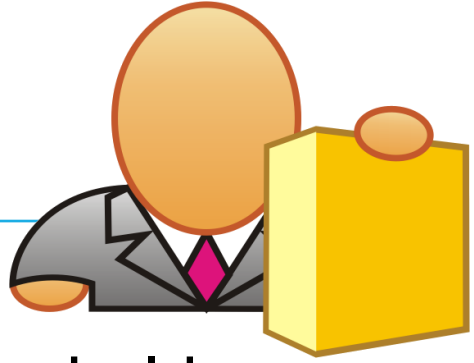
Ceremonies

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

Artifacts

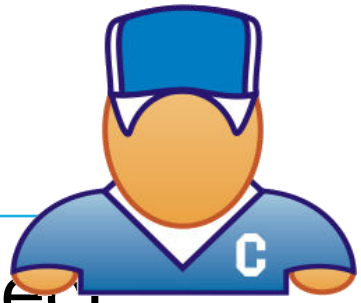
- Product backlog
- Sprint backlog
- Burndown charts

Product owner



- The product owner represents the stakeholders and is the voice of the customer and is accountable for ensuring that the team delivers value to the business.
- 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

The ScrumMaster



- 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

The Team

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

The Team

- Teams are self-organizing
  - Ideally, no titles but rarely a possibility
- Membership should change only between sprints

Scrum Framework

Roles

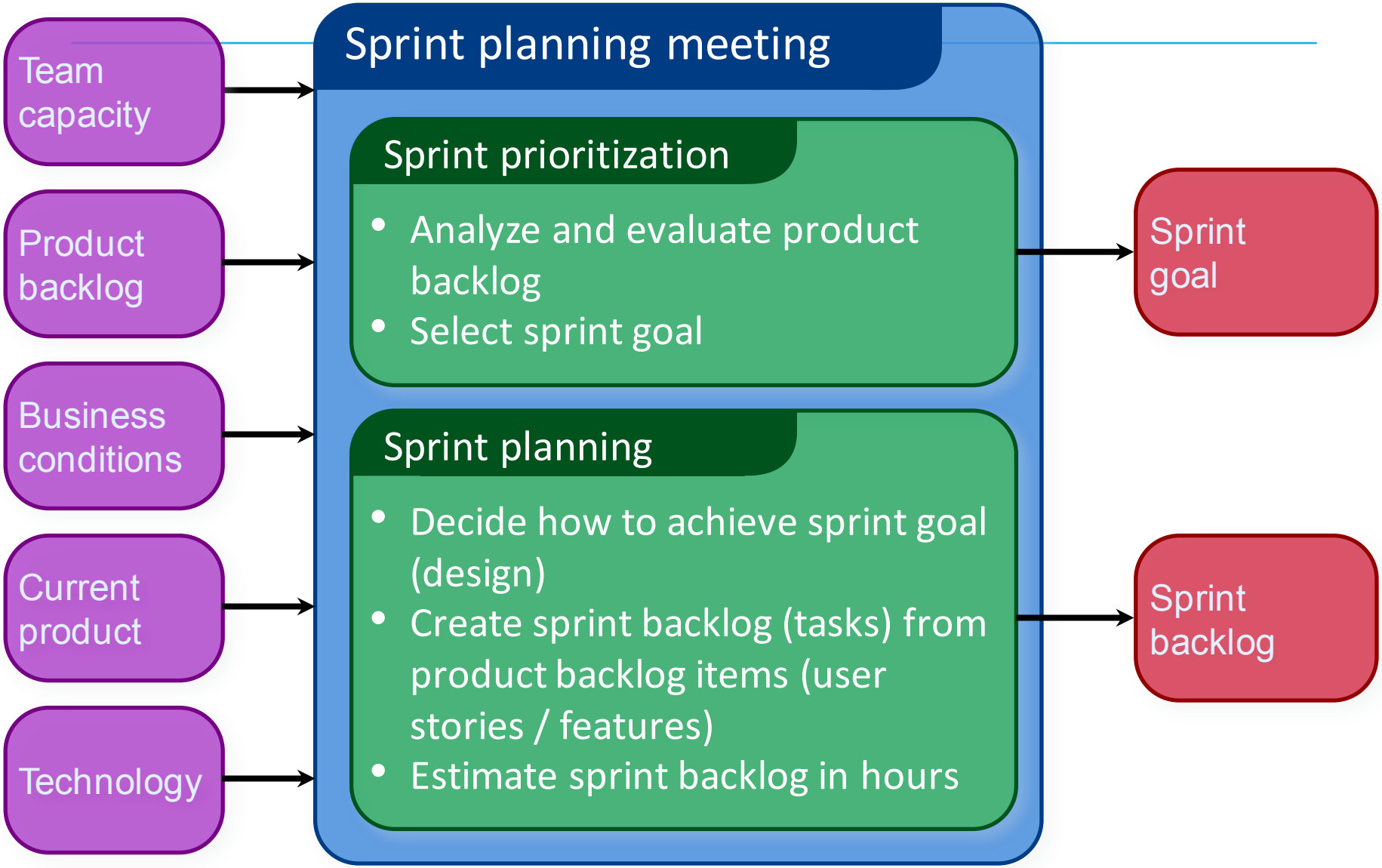
- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts


Team capacity
Product backlog
Business conditions
Current product
Technology
Sprint planning meeting
Sprint prioritization
Analyze and 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 goal
Sprint backlog

Sprint Planning

- Team **selects items from the product backlog they can commit to** completing
- Sprint backlog is created
  - **Tasks to implement/complete the product backlog items are identified and each task is estimated** (1-16 hours)
  - Collaboratively, not done alone by the ScrumMaster
- High level design is considered

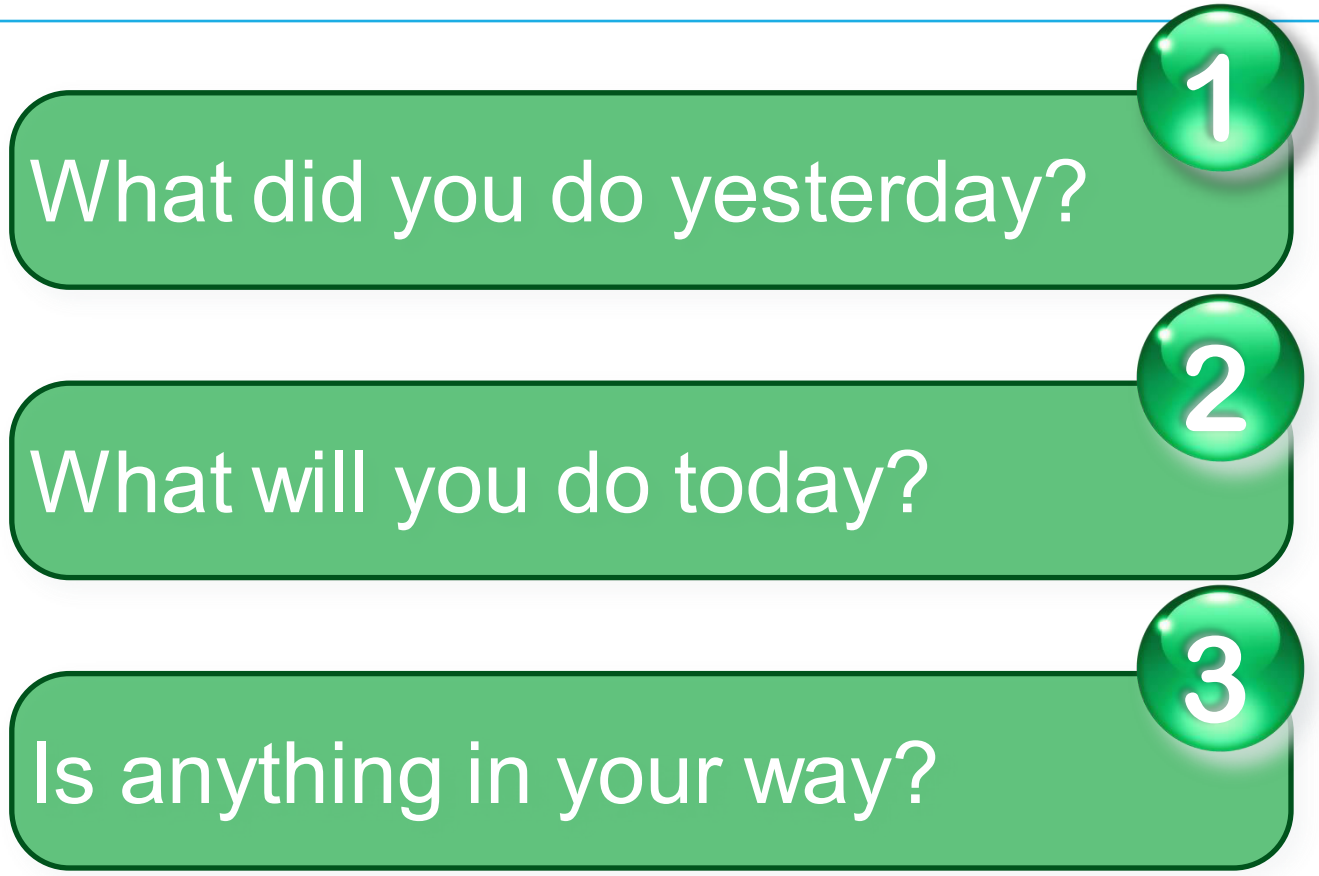
As a vacation planner, I want to see photos of the hotels.

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

The Daily Scrum

- To ensure everything is on track
- Parameters
  - Daily
  - 15-minutes
  - Stand-up
- Not for problem solving
  - Whole world is invited
  - Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings

Everyone answers 3 questions



- These are *not* status for the ScrumMaster
  - They are commitments in front of peers

The Sprint Review

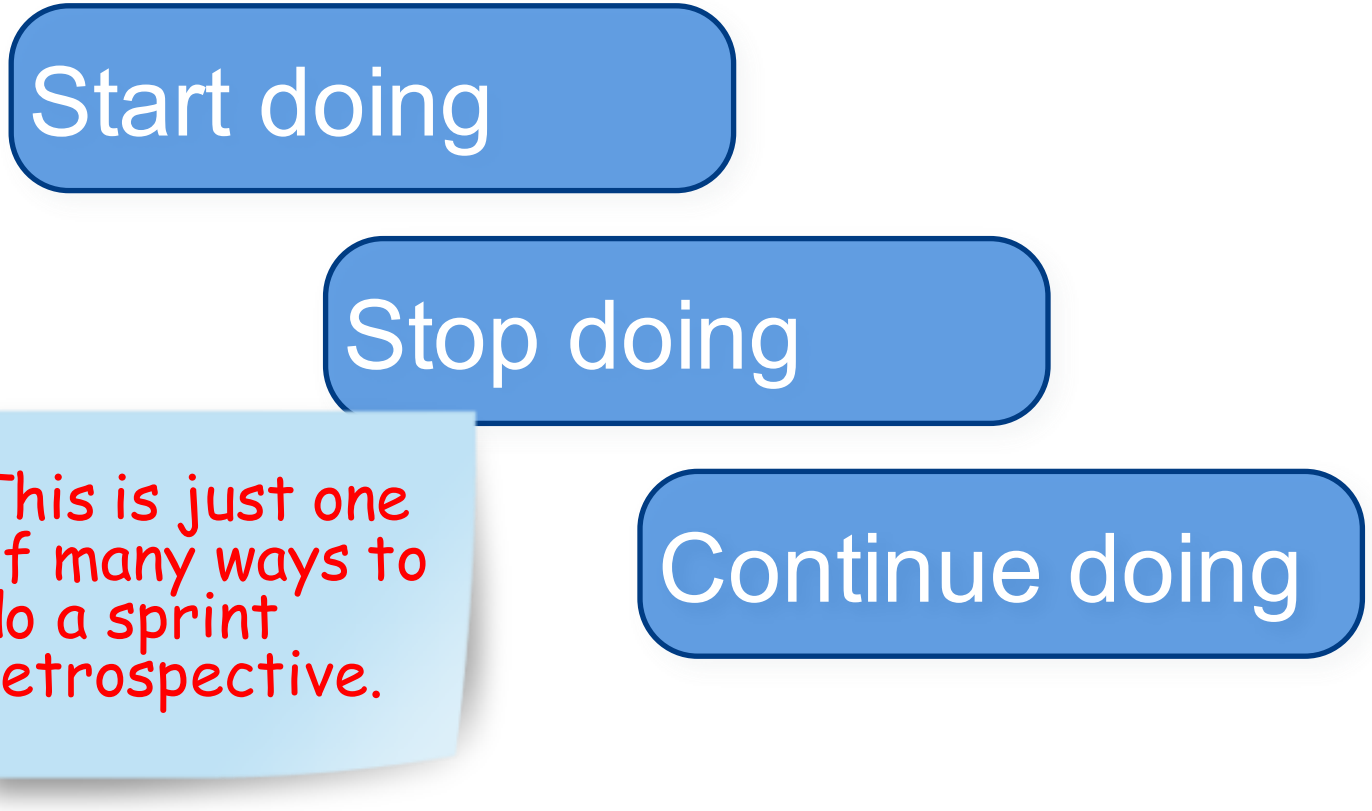
- Team presents what it accomplished during the sprint
- Typically takes the **form of a demo** of new features or underlying architecture
- Informal
  - 2-hour prep time rule
  - No slides
- Whole team participates
- Invite the world

Sprint Retrospective

- Periodically take a look at what is and is not working
- **What went right and areas of improvement**
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
  - ScrumMaster
  - Product owner
  - Team
  - Possibly customers and others

Start / Stop / Continue

- Whole team gathers and discusses what they'd like to:



Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product Backlog

This is the product backlog

- The requirements/features
- A list of all desired work on the project
- **Collection of user stories: a wish list of all the user stories of a product.**
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Reprioritized at the start of each sprint

A Sample Product Backlog

Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

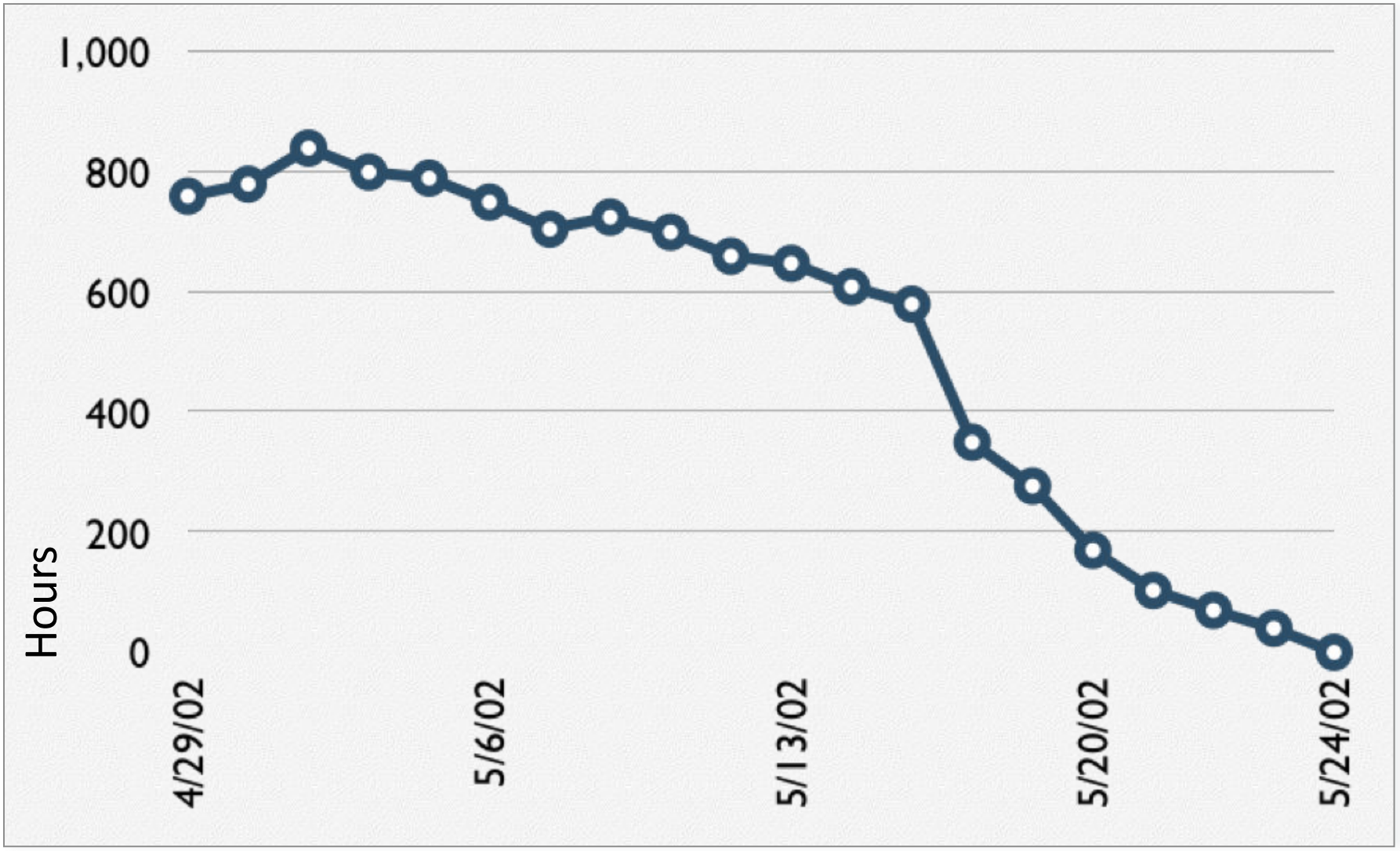
Managing the Sprint Backlog

- Individuals sign up for work of their own choosing
  - Work is never assigned
- Estimated work remaining is updated daily
- Create sprint backlog (tasks) from product backlog items (user stories / features)
- **Sprint backlog is a list of tasks required to complete/implement the user stories/features.**

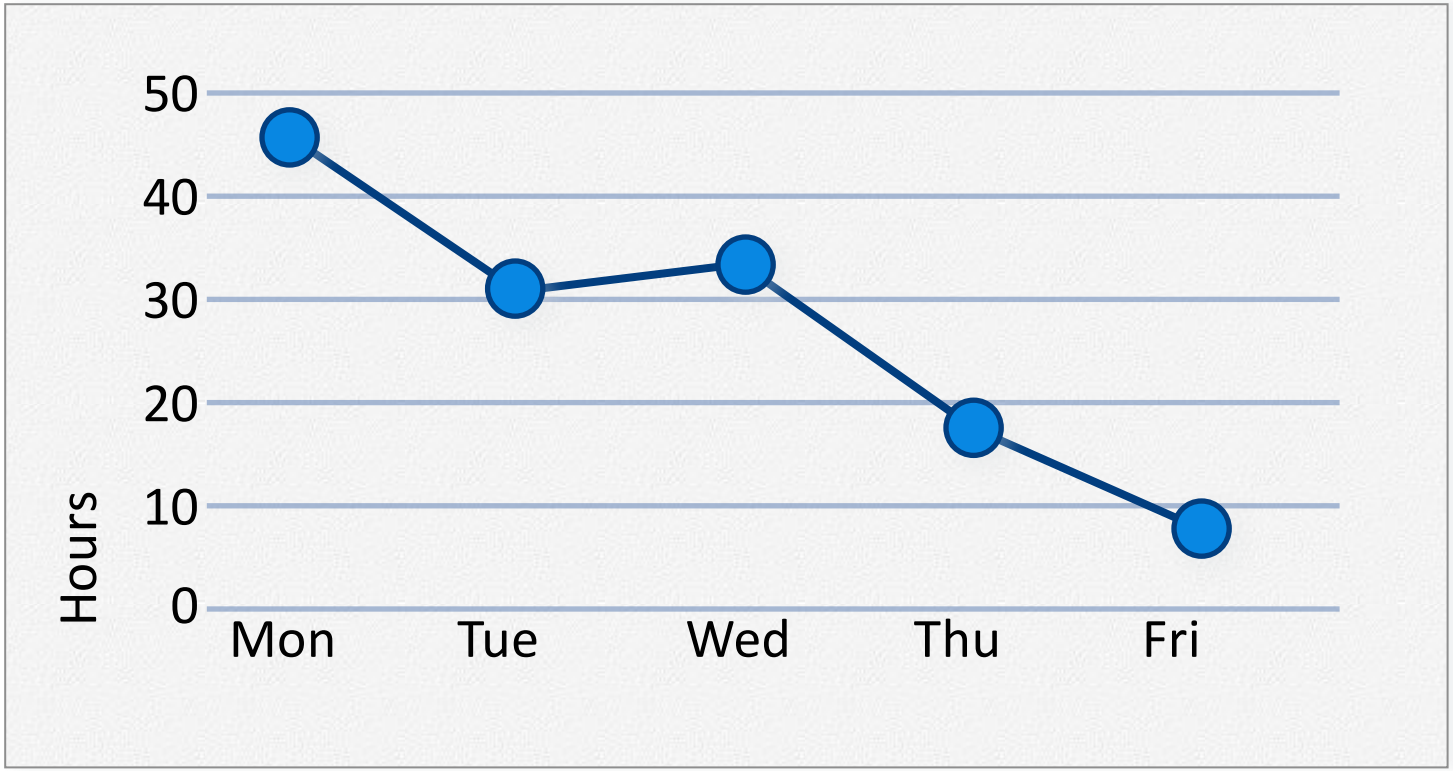
A Sprint Backlog

Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

Burndown Chart
(to monitor the progress of each sprint)



Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



Where to go next

- www.mountaingoatsoftware.com/scrum
- www.scrumalliance.org
- www.controlchaos.com
- scrumdevelopment@yahoogroups.com
- *Agile and Iterative Development: A Manager's Guide* by Craig Larman
- *Agile Estimating and Planning* by Mike Cohn
- *Agile Project Management with Scrum* by Ken Schwaber
- *Agile Retrospectives* by Esther Derby and Diana Larsen

A Scrum reading list

- Here is a series of videos on Scrum Training by http://scrummethodology.com/

Scrum Training Series:

- **Introduction to Scrum**
http://scrumtrainingseries.com/Intro_to_Scrum/Intro_to_Scrum.htm

- **Backlog Refinement Meeting**
http://scrumtrainingseries.com/BacklogRefinementMeeting/BacklogRefinementMeeting.htm

- **Sprint Planning Meeting (MUST WATCH)**
http://scrumtrainingseries.com/SprintPlanningMeeting/SprintPlanningMeeting.htm

- **Daily Scrum Meeting**
http://scrumtrainingseries.com/DailyScrumMeeting/DailyScrumMeeting.htm

- **Sprint Review Meeting**
http://scrumtrainingseries.com/SprintReviewMeeting/SprintReviewMeeting.htm

- **Sprint Retrospective Meeting**
http://scrumtrainingseries.com/SprintRetrospectiveMeeting/SprintRetrospectiveMeeting.htm

Conclusion

- This Week's Workshop
  - Tasks – Agile System Development
  - Quiz 9 – State and Event Modelling (3 marks)

- Next Week's Lecture
  - Revision and Sample Final Exam

- Next Week's Workshop
  - Tasks – Revision and Sample Final Exam
  - Quiz 10 – Agile System Development (3 marks)