

CPSC 3720

Lesson 5

Connie Taylor
Professor of Practice



Today's Objectives

- Quick recap of prior class
- Deeper understanding of Agile/Scrum
 - Agile origins
 - Agile principles
 - Scrum practices

The Tar Pit – Complexity of a Program vs. Product



Single program <i>Couple devs in a garage – used by the devs</i>	Programming System <i>Dependencies/integration, performance testing</i>
Programming Product <i>General usage, testing, doc</i>	Programming Systems Product <i>Product+Systems needs</i>

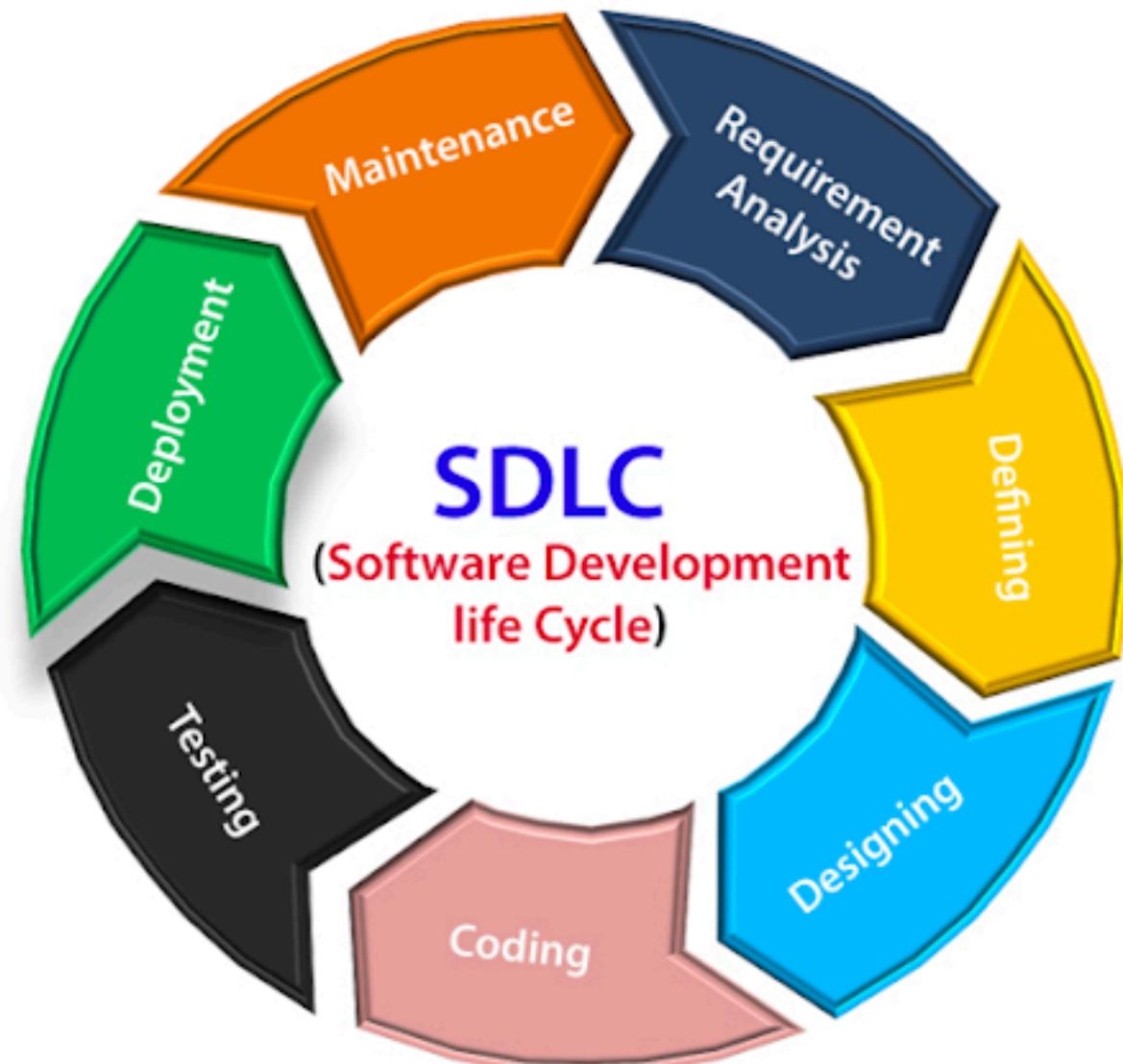
How do we manage this complexity??

Software Development Process

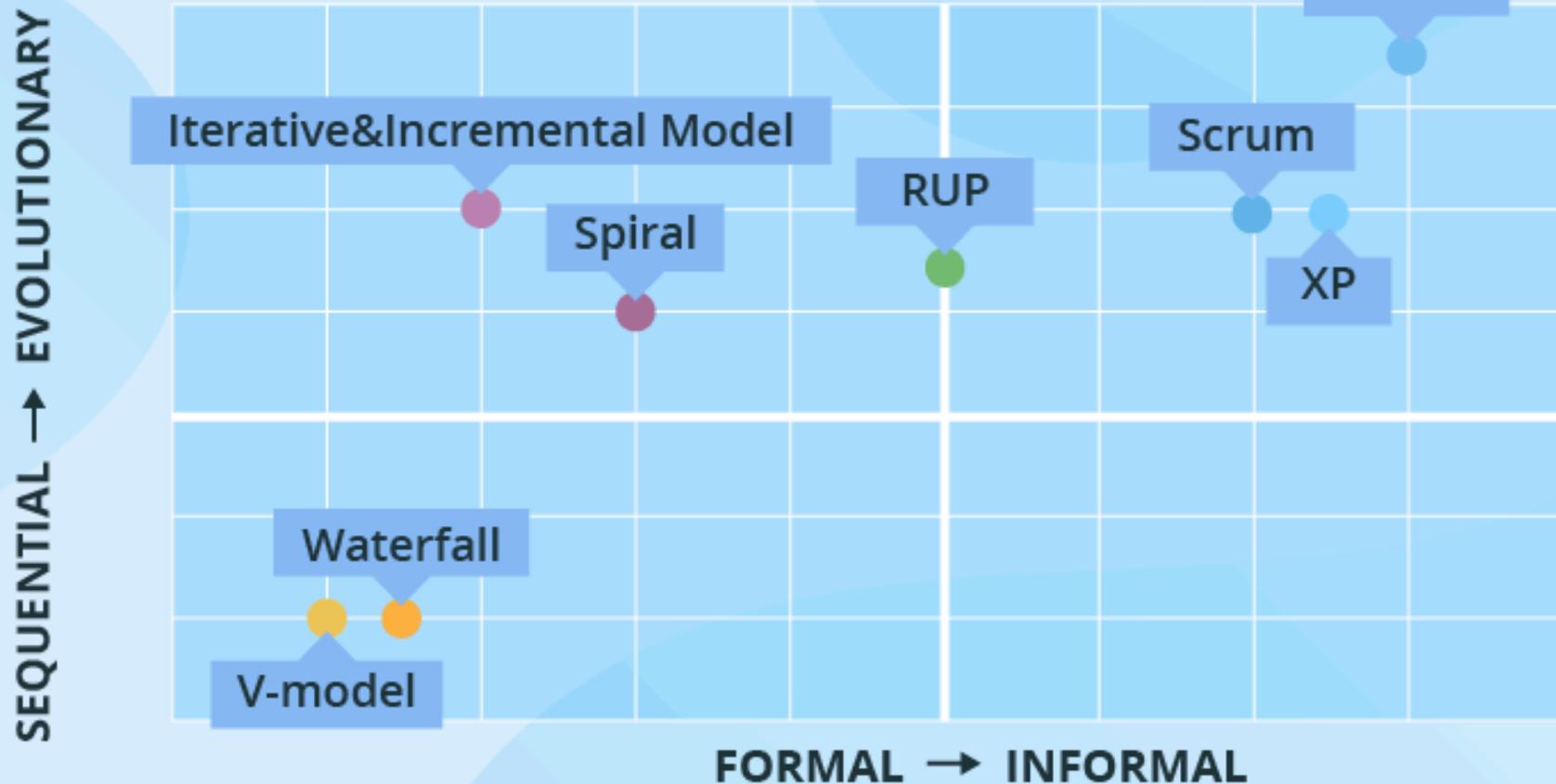
Software Process: a way of breaking down the overall software development work into manageable sub-tasks; systematic and somewhat formal

Software Development

Process Steps



TYPES OF POPULAR SDLC MODELS



Classical methods of software development have many disadvantages:

- huge effort during the planning phase
- poor requirements conversion in a rapid changing environment
- treatment of staff as a factor of production

- Agile methods are considered
 - Lightweight
 - People-based rather than Plan-based
- No single Agile method
 - Scrum
 - XP
 - Kanban
 - Lean
- Agile Manifesto closest to a definition
 - Set of principles
 - Developed by Agile Alliance in 2001

The Agile Manifesto

Individuals and interactions

over

Process and tools

Working software

over

Comprehensive documentation

Customer collaboration

over

Contract negotiation

Responding to change

over

Following a plan

12 Agile 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.
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.

12 Agile Principles cont'd

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.
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.
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.

Breakout #1

- Spend 10 minutes going through the following game; pick one person to launch and share/control the game in each room
- <https://sevawise.com/tools/games/scrum-values>

“The... ‘relay race’ approach to product development...may conflict with the goals of maximum speed and flexibility. Instead a holistic or ‘rugby’ approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today’s competitive requirements.”

Hirotaka Takeuchi and Ikujiro Nonaka, “The New New Product Development Game”,
Harvard Business Review, January 1986.

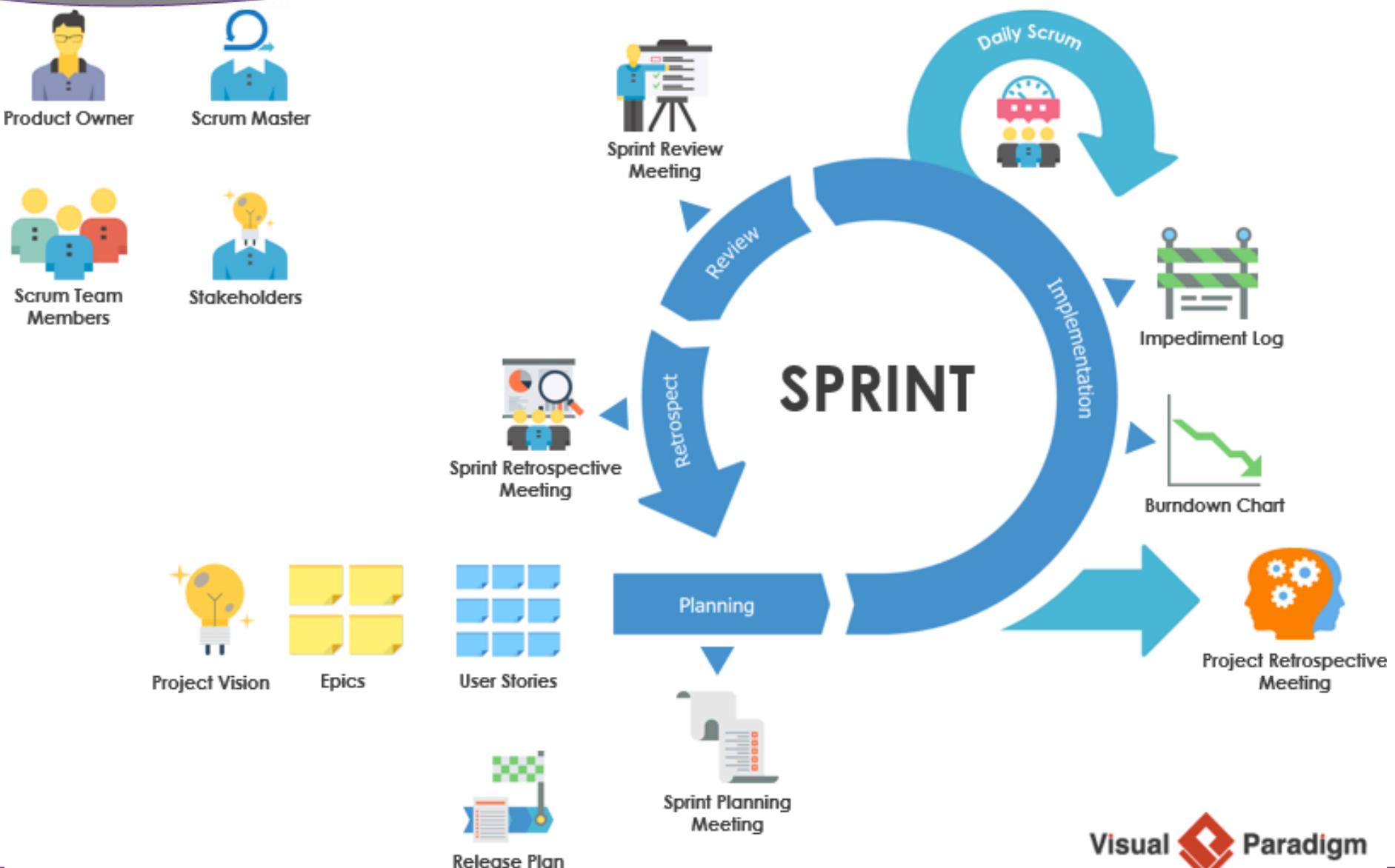
Scrum



Scrum in 100 Words

- 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.

Scrum in 1 Picture



Sprints

- Scrum projects make progress in a series of “sprints”
- 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

Requirements

Design

Code

Test

Rather than doing all of
one thing at a time...

...Scrum teams do a little
of everything all the time

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

No changes during a sprint



Plan your sprint durations around how long you can commit to keeping change out of the sprint

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts
- Impediment Log

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts
- Impediment Log

Product Owner



Product Owner Stakeholders

- Define the features of the product working with stakeholders
- 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



Scrum Master

- Represents management to the project
- Servant leadership (they work for the team)
- 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)
- Teams are self-organizing
 - Ideally, no titles, but rarely a possibility
- Membership should change only between sprints but team consistency is best

Breakout #2

- Spend 10 minutes going through the following game; pick one person to launch and share/control the game in each room
- <https://sevawise.com/tools/games/scrum-roles>

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

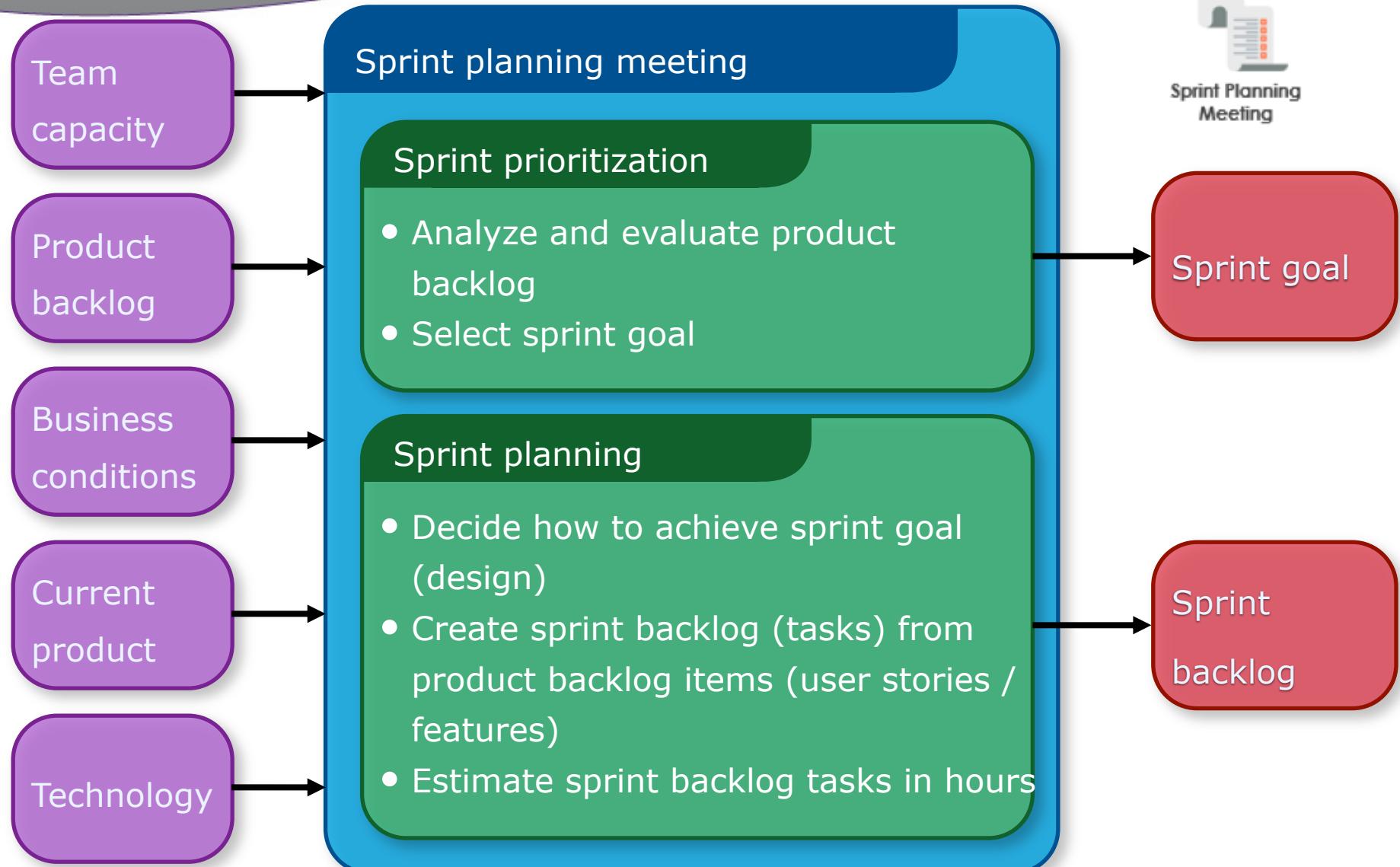
Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts
- Impediment Log

Sprint planning



Sprint Planning
Meeting

Sprint planning

Team selects items from the product backlog they can commit to completing

Sprint backlog is created

High-level design is considered

- Tasks are identified and each is estimated (1-16 hours)
- Collaboratively, not done alone by the ScrumMaster

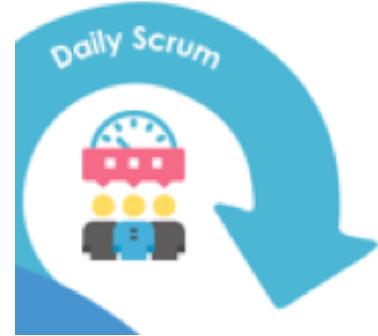
As a vacation planner, I want to see photos of the hotels.



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

- Parameters
 - Daily
 - 15-minutes
 - Stand-up
- Not for problem solving
 - Whole world can be invited, BUT
 - Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings



The Daily Scrum: Everyone answers 3 questions

1

What did you do yesterday?

2

What will you do today?

3

Is anything in your way?

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

The Daily Scrum



<https://youtu.be/GzQjGhD5tSU>

The Daily Scrum



<https://youtu.be/oLmDe8pAc6I>

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
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
 - ScrumMaster
 - Product owner
 - Team
 - Possibly customers and others



Sprint Retrospective
Meeting

Sprint Retrospective: Start/Stop/Keep

The whole team gathers and discusses what they'd like to:

Start doing

Stop doing

This is just one
of many ways
to do a sprint
retrospective.

Keep doing

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts
- Impediment Log

Product Backlog



- The requirements
- A list of all desired work on the project
- 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

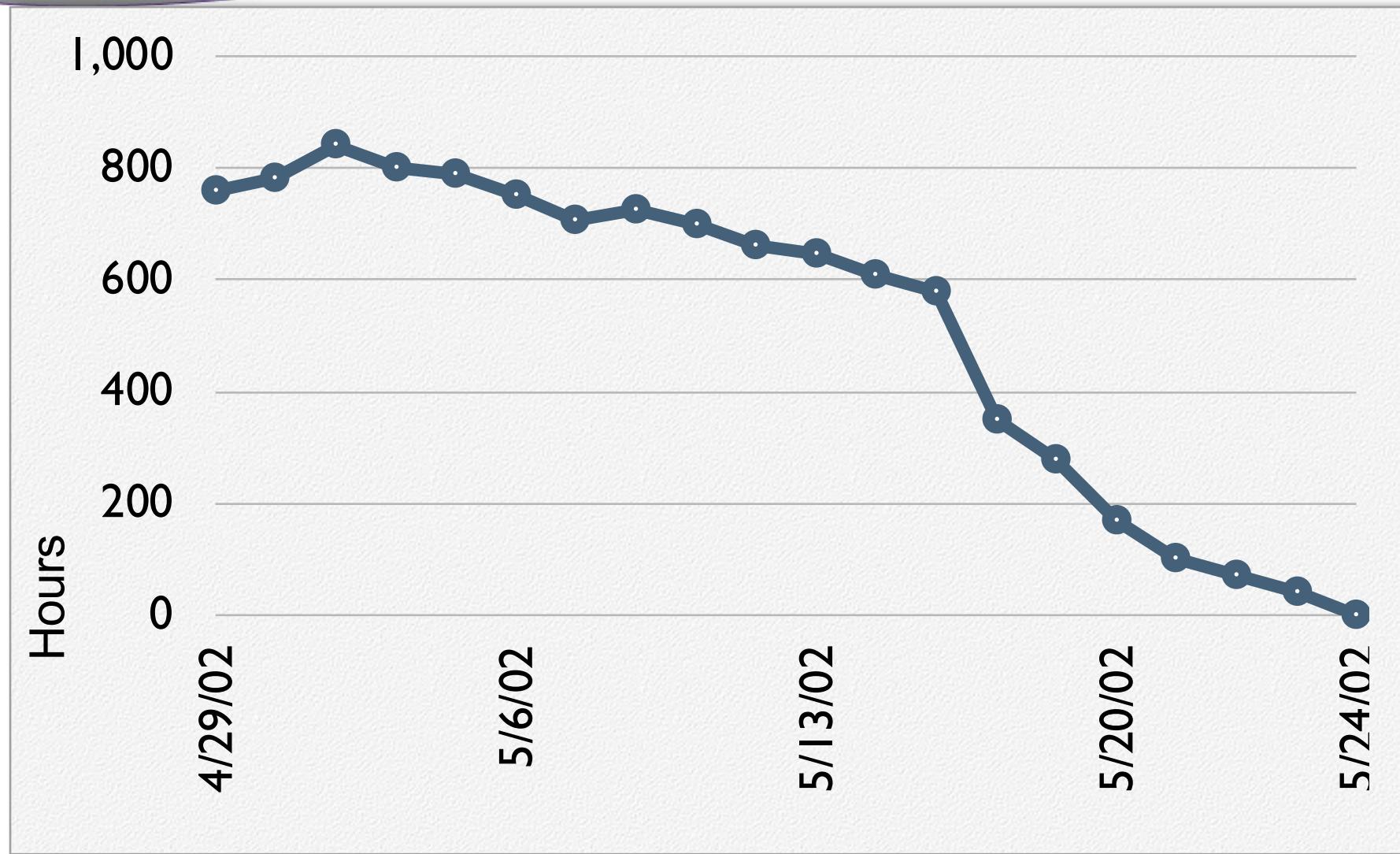
Example Product Backlog

Backlog item	Storypoint 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

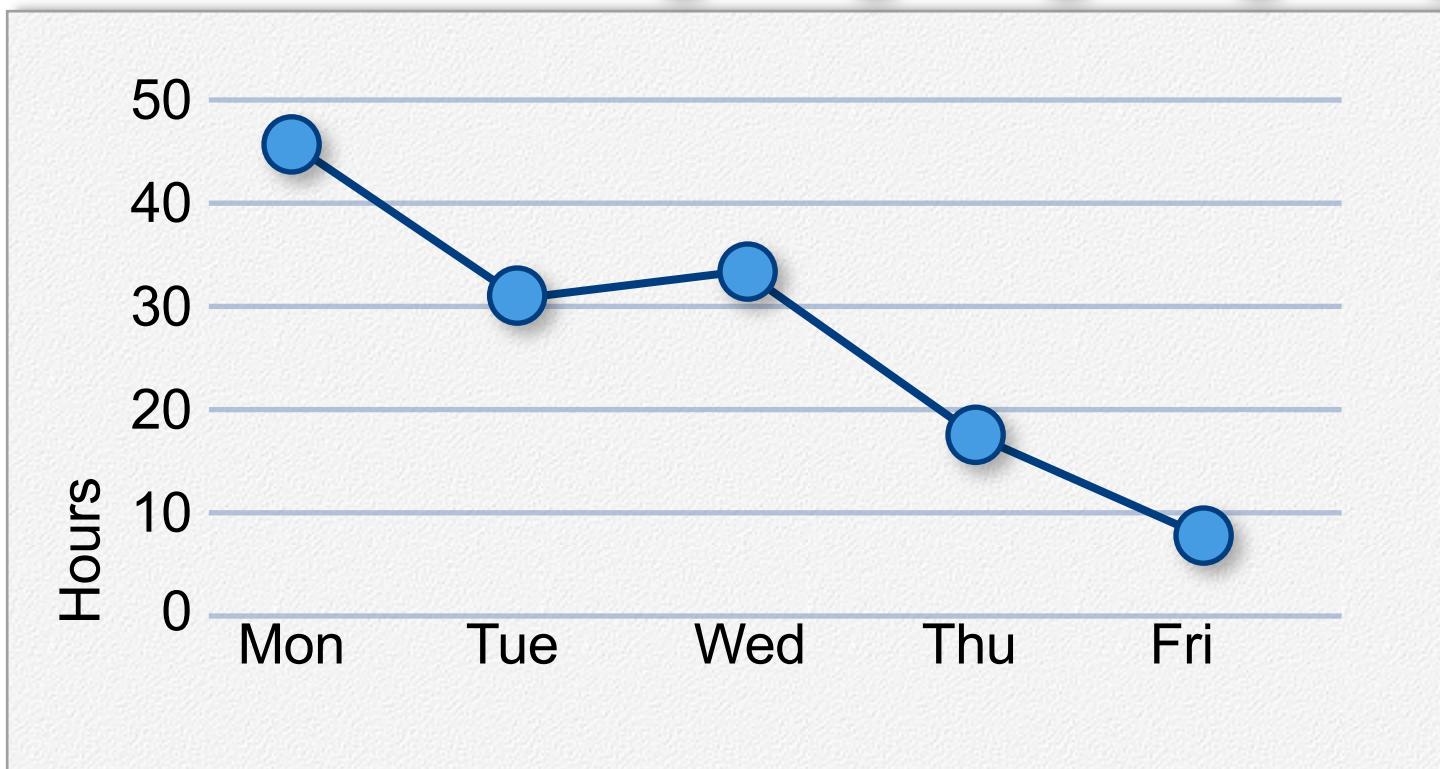
Sprint Backlog

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	

Sprint Burndown Chart



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				



Impediment Log

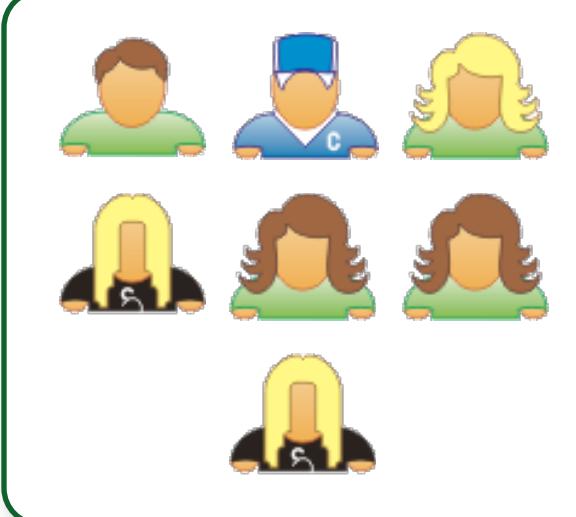
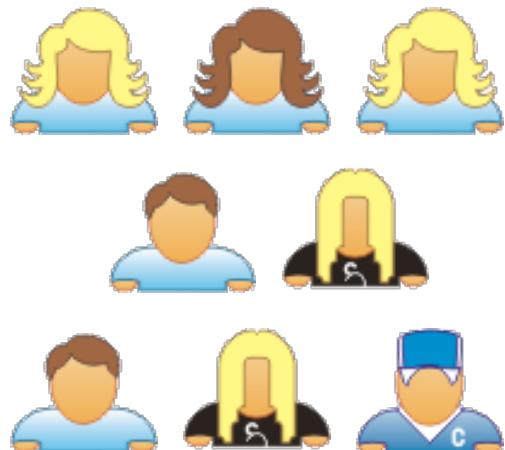


- The ScrumMaster is managing all impediments to the team that is impacting their ability to get work done
- Examples
 - Build server keeps crashing
 - Joe Sr. Developer keeps getting pulled into code reviews for other teams
 - A team member is not showing up to daily standups

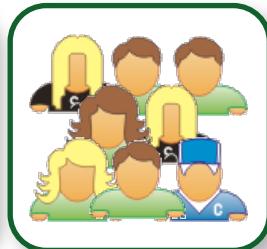
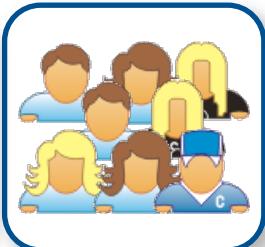
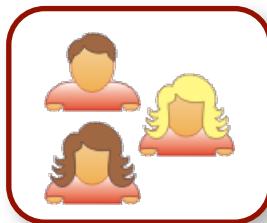
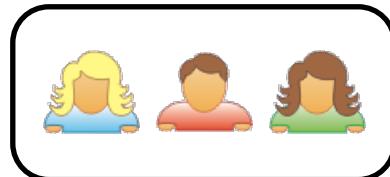
Scrum Scalability

- Typical individual scrum team is 7 ± 2 people
 - Scalability comes from teams of teams
- Factors in scaling
 - Type of application
 - Team size
 - Team dispersion
 - Project duration
- Scrum has been used on multiple 500+ person projects

Scaling through the Scrum of scrums



Scrum of scrums of scrums



Sources

- www.mountaingoatsoftware.com/scrum
- www.ScrumFoundations.com
- www.mountaingoatsoftware.com/agile



An Agile/Scrum reading list

- *Agile Estimating and Planning* by Mike Cohn
- *Agile Product Management: Creating Products that Customers Love* by Roman Pichler
- *Agile Project Management with Scrum* by Ken Schwaber
- *Agile Software Development Ecosystems* by Jim Highsmith
- *Essential Scrum: A Practical Guide to the Most Popular Agile Process* by Kenneth Rubin
- *Scrum and XP from the Trenches* by Henrik Kniberg
- *Succeeding with Agile: Software Development using Scrum* by Mike Cohn
- *The Scrum Guide* at www.ScrumGuides.org
- *User Stories Applied for Agile Software Development* by Mike Cohn

Next Week

- High performing teams
- Creating epics and stories
- Creating a prioritized backlog
- Estimating work
- Reading: User Stories Applied by Mike Cohn, Chapters 1-3
(available on the O'Reilly site)

A group of students are starting a **Google Developers Student Club** at Clemson University. They will be hosting an info session at a soon to be determined time next week. If you are interested, please sign up using the link below.

https://docs.google.com/forms/d/e/1FAIpQLSfJLxR1FEAh1nq71V7IPYMNevdFUs6Ic3nalvrz76Ej2am55g/viewform?usp=sf_link