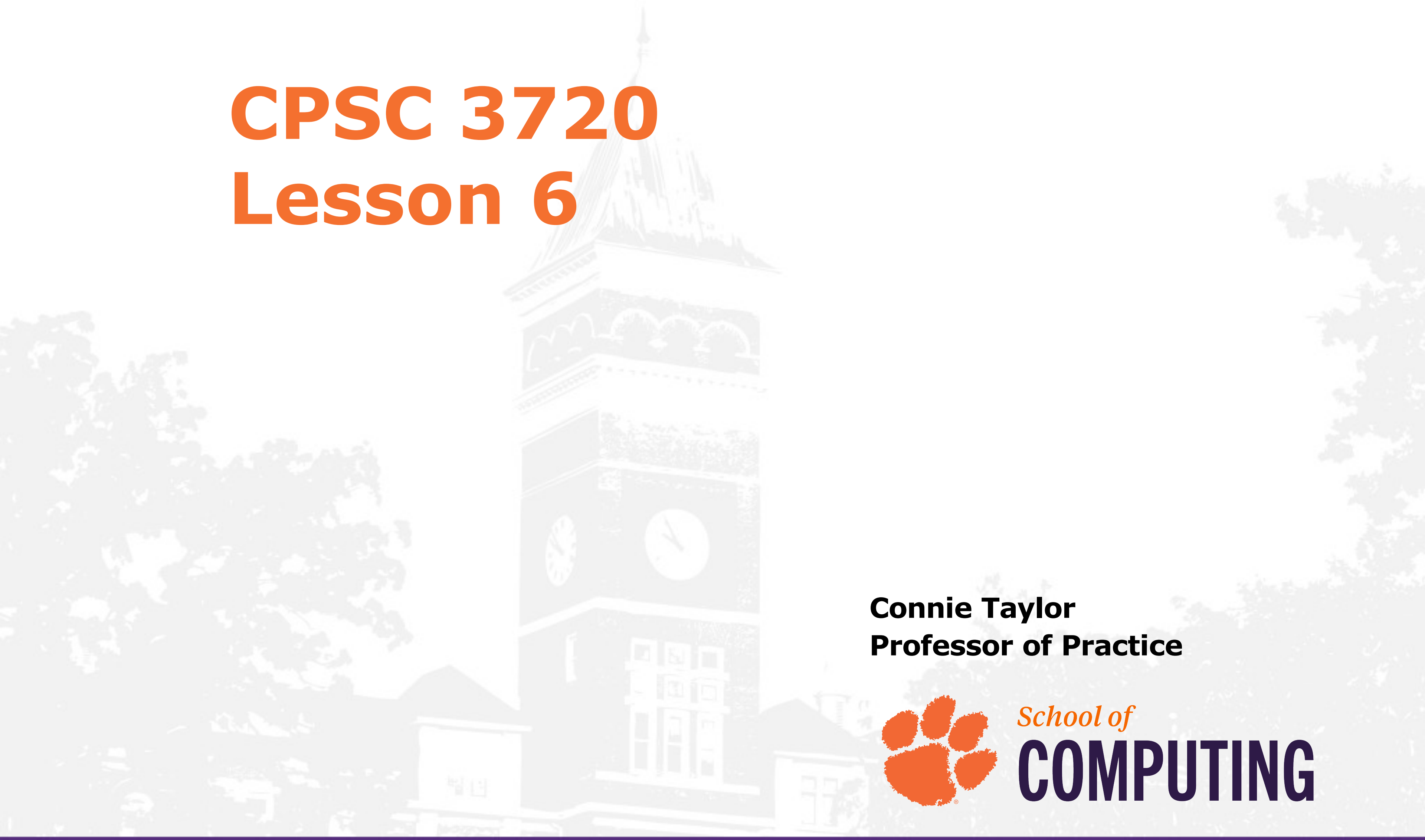
CPSC 3720 Lesson 6



Connie Taylor
Professor of Practice

Today's Objectives

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

The Tar Pit – Complexity of a Program vs. Product

HOW LONG WILL YOUR PROJECT TAKE IF I ADD TWO PEOPLE?
ADD ONE MONTH FOR TRAINING, ONE MONTH FOR THE EXTRA COMPLEXITY, AND ONE MONTH TO DEAL WITH THEIR DRAMA.
BUT AFTER ALL OF THAT. . .
THEY'LL BE AS USEFUL AS THIS MEETING.
Dilbert.com DilbertCartoonist@gmail.com
4-29-10 ©2010 Scott Adams, Inc./Dist. by UFS, Inc.

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

3x

3x

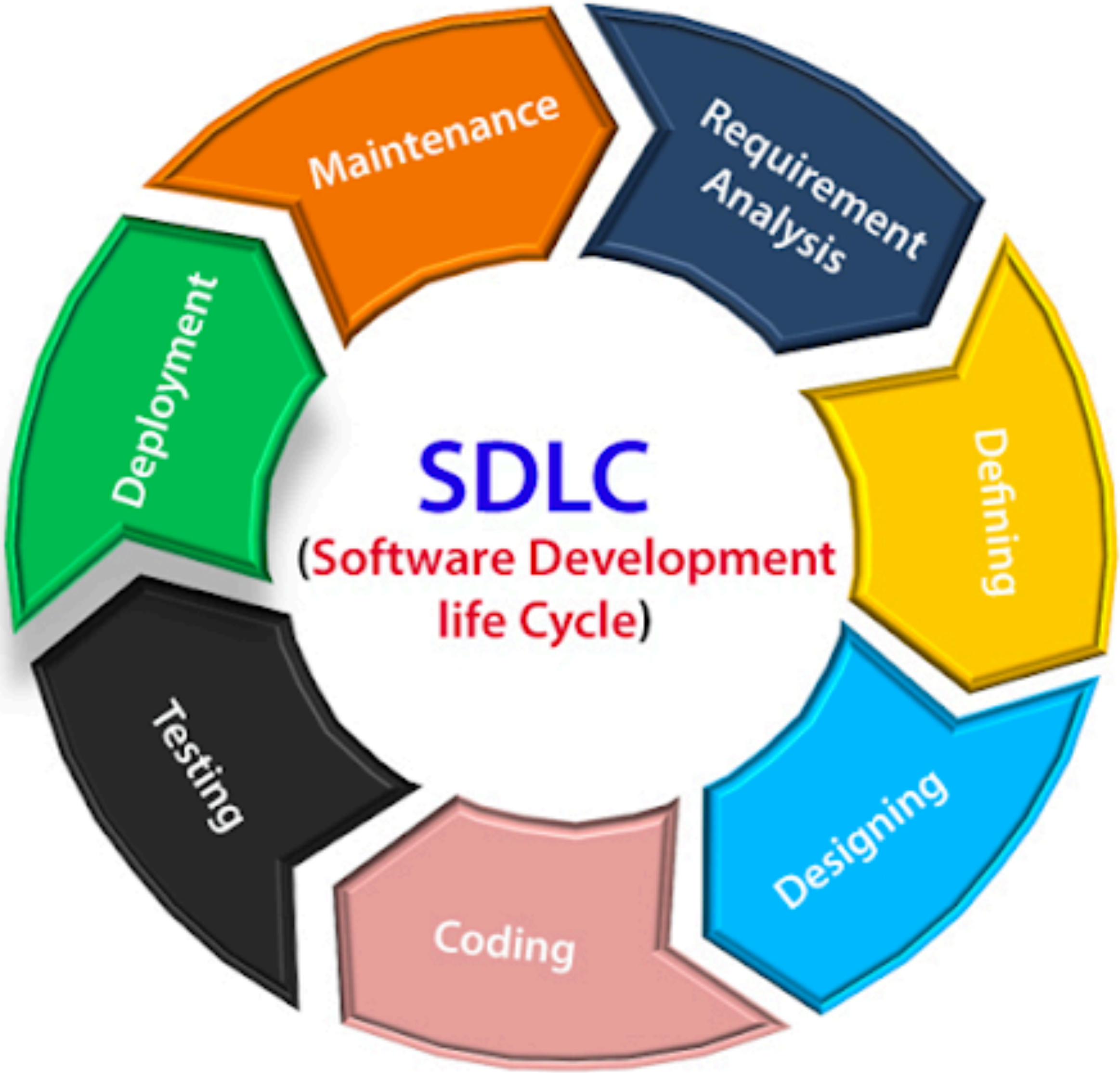
9x

How do we manage this complexity??

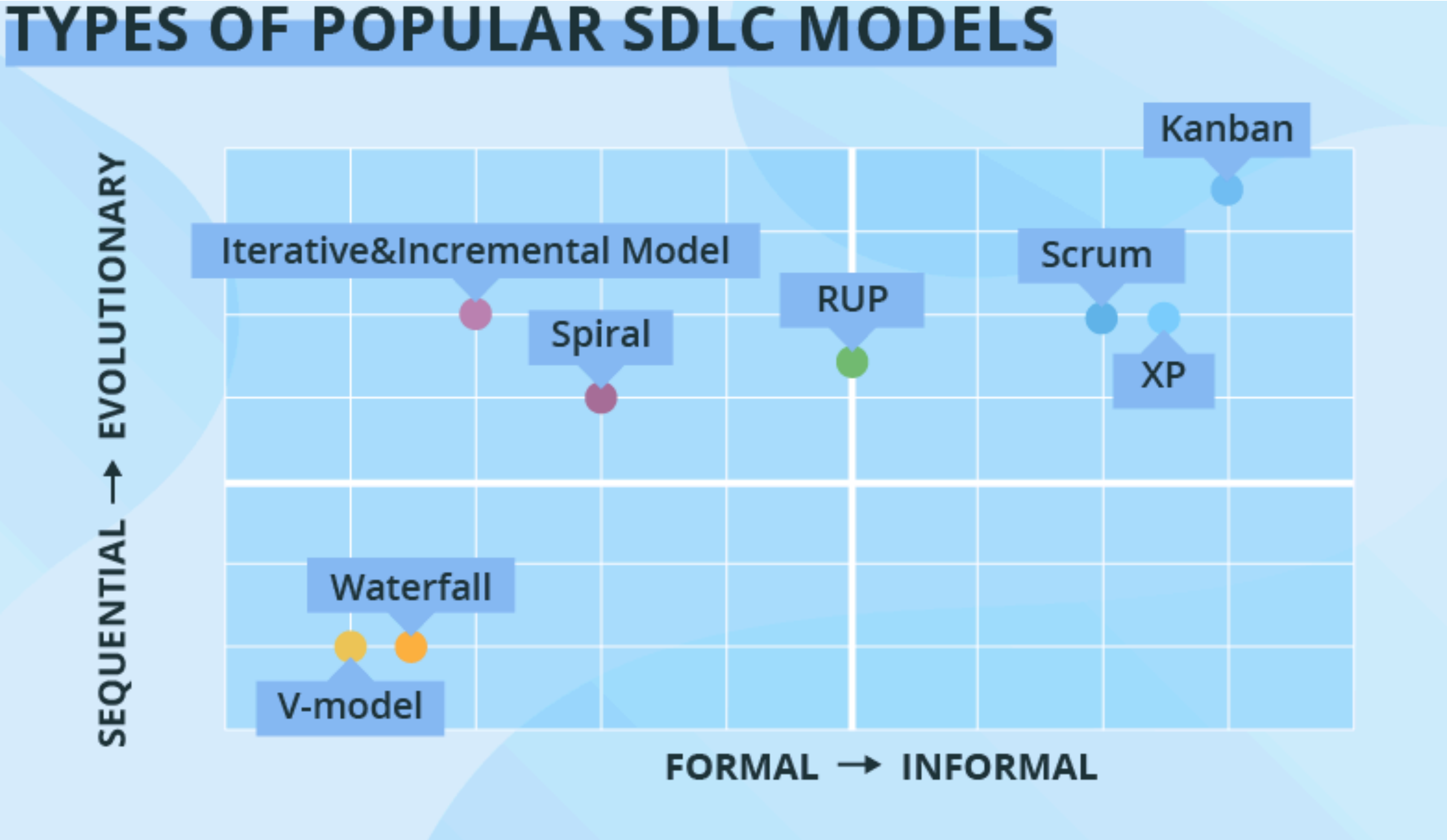
Software Development Process

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

Software Development Process Steps



SDLC Processes



Process concerns

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

- Agile methods are considered
  - Lightweight
  - People-based rather than Plan-based

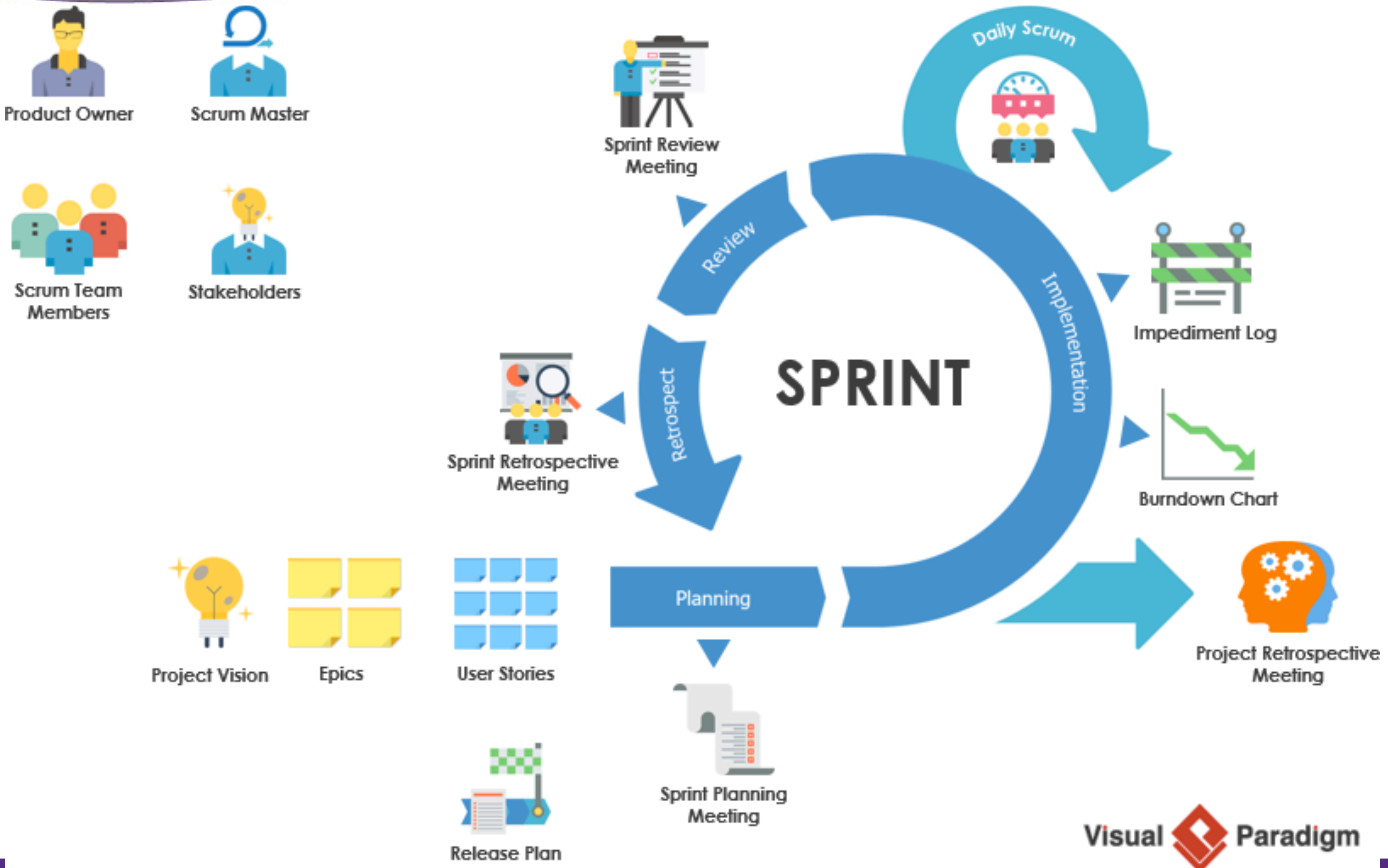
- No single Agile method
  - Scrum
  - XP
  - Kanban
  - Lean

- Agile Manifesto closest to a definition
  - Set of principles
  - Developed by Agile Alliance in 2001

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



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

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

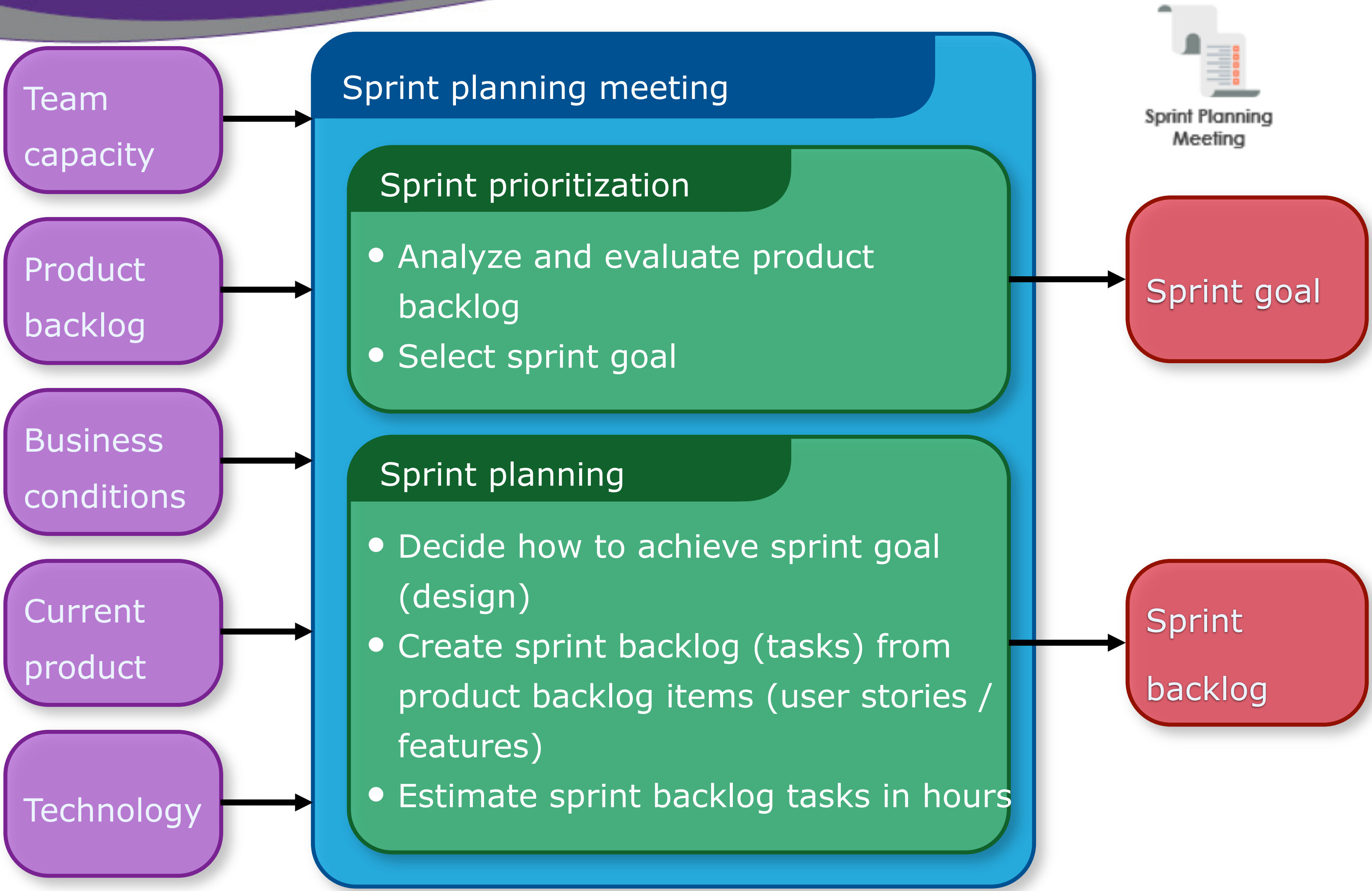
Ceremonies

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

Artifacts

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

Sprint planning



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 tasks in hours

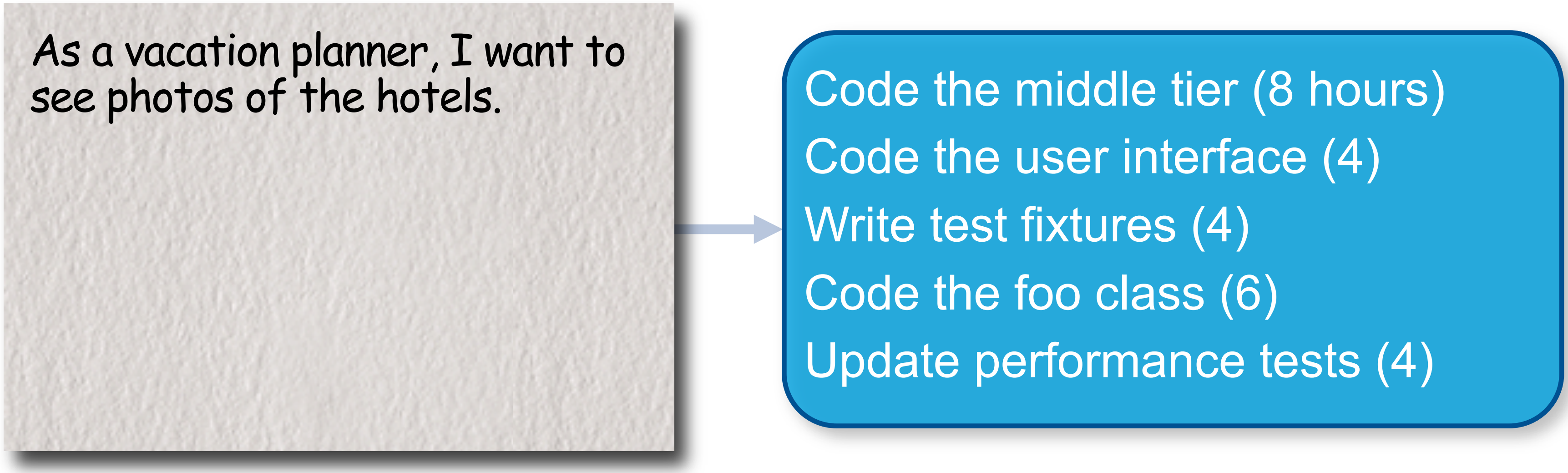
Sprint Planning Meeting

Sprint goal

Sprint backlog

Sprint planning

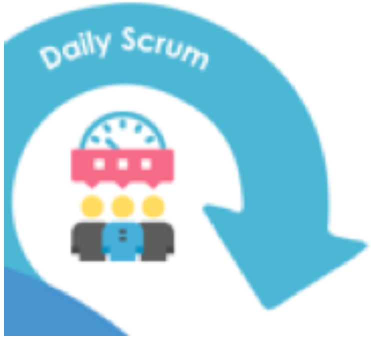
- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
- High-level design in considered
  - Tasks are identified and each is estimated (1-16 hours)
  - Collaboratively, not done alone by the ScrumMaster



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

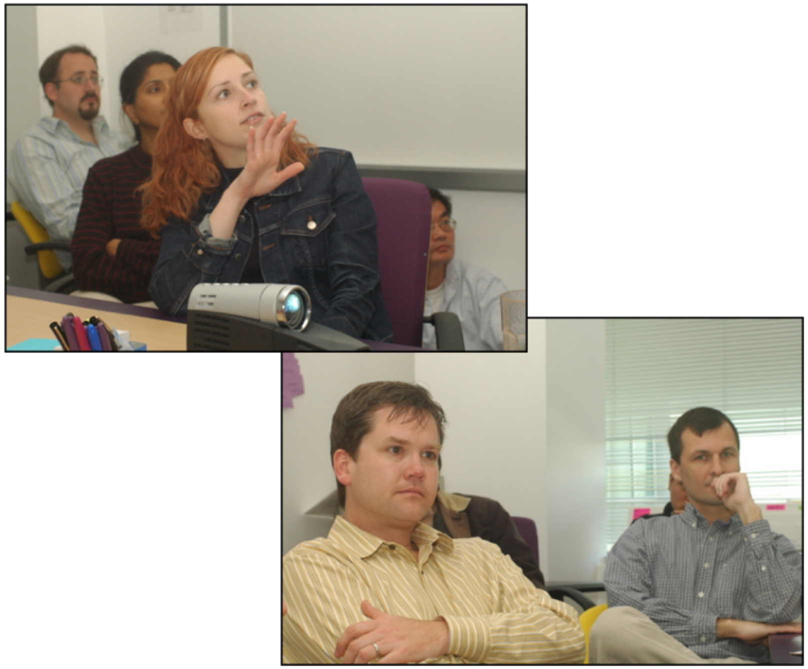
The Daily Scrum

https://youtu.be/oLmDe8pAc6I

The Sprint Review

Sprint Review Meeting

- 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: Start/Stop/Keep

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

Start doing

Stop doing

Keep doing

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

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

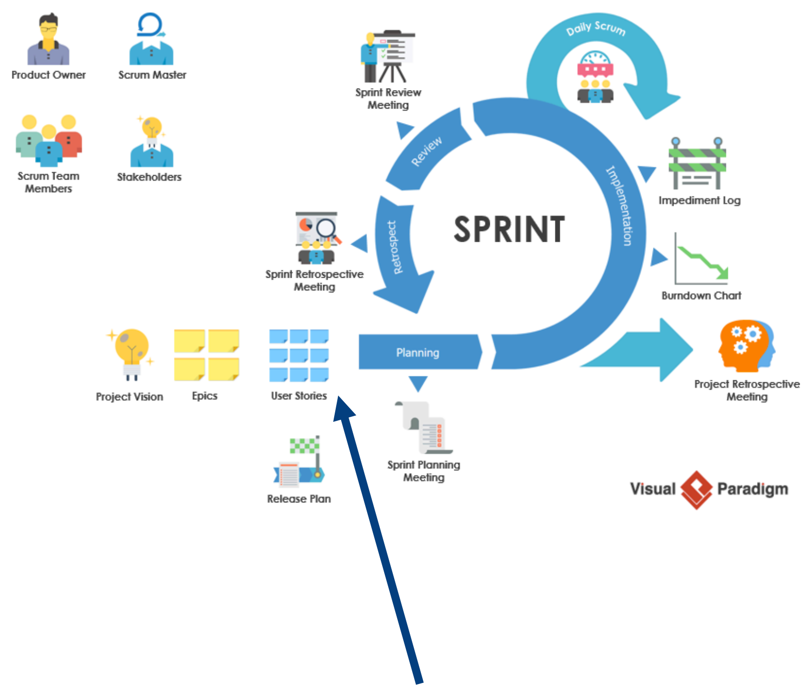
Ceremonies

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

Artifacts

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

Product Backlog



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

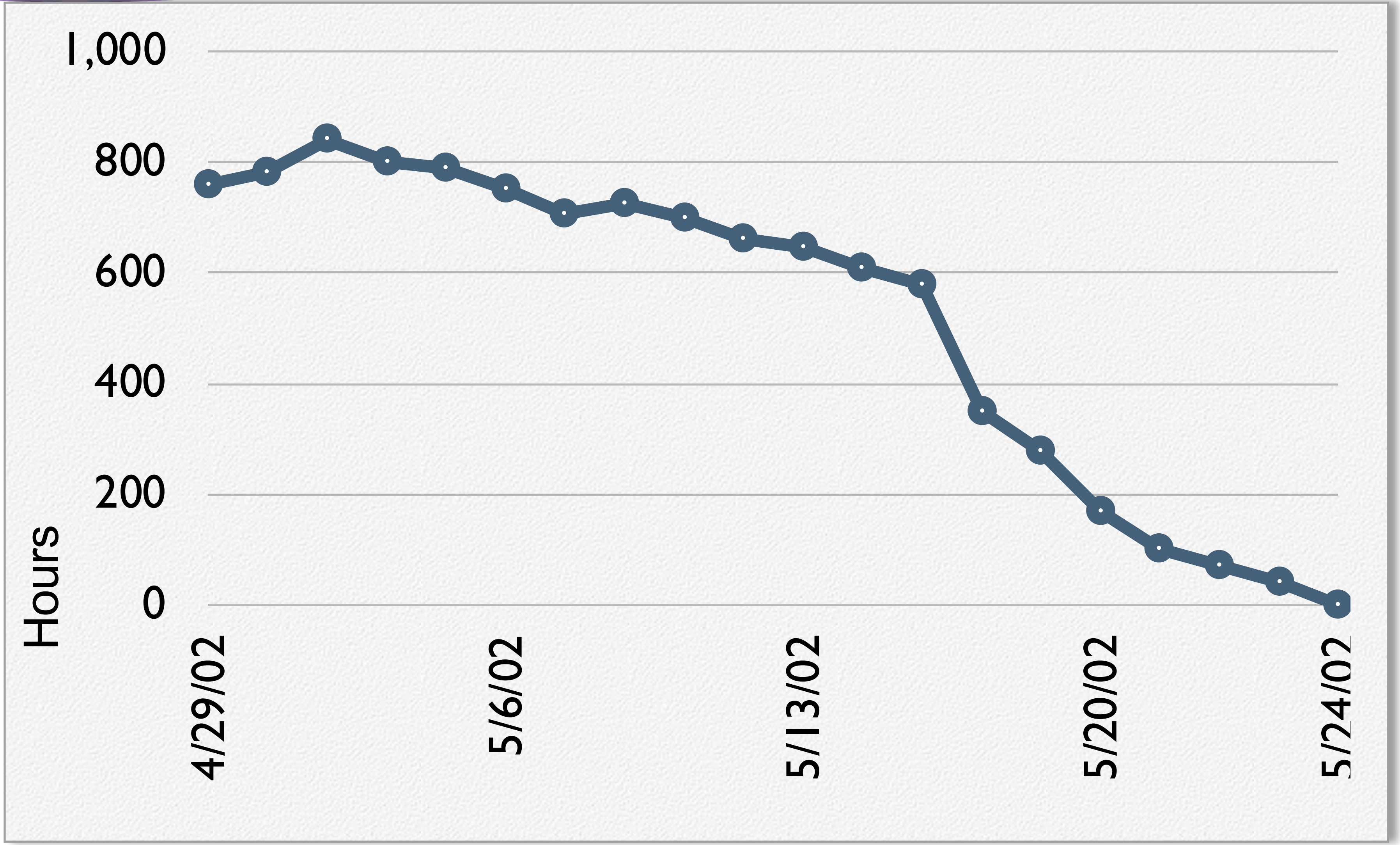
Example Product Backlog

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

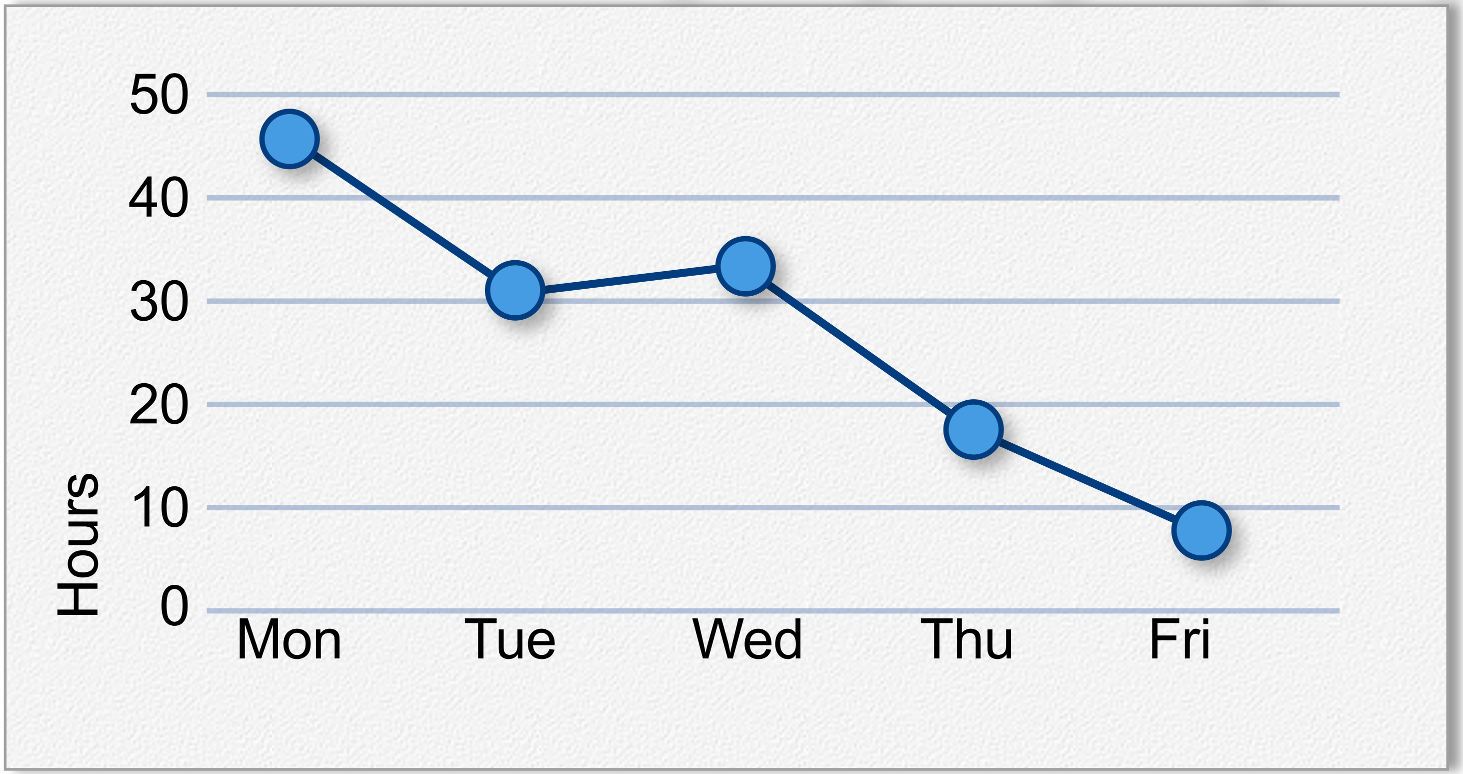
Sprint Backlog

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				



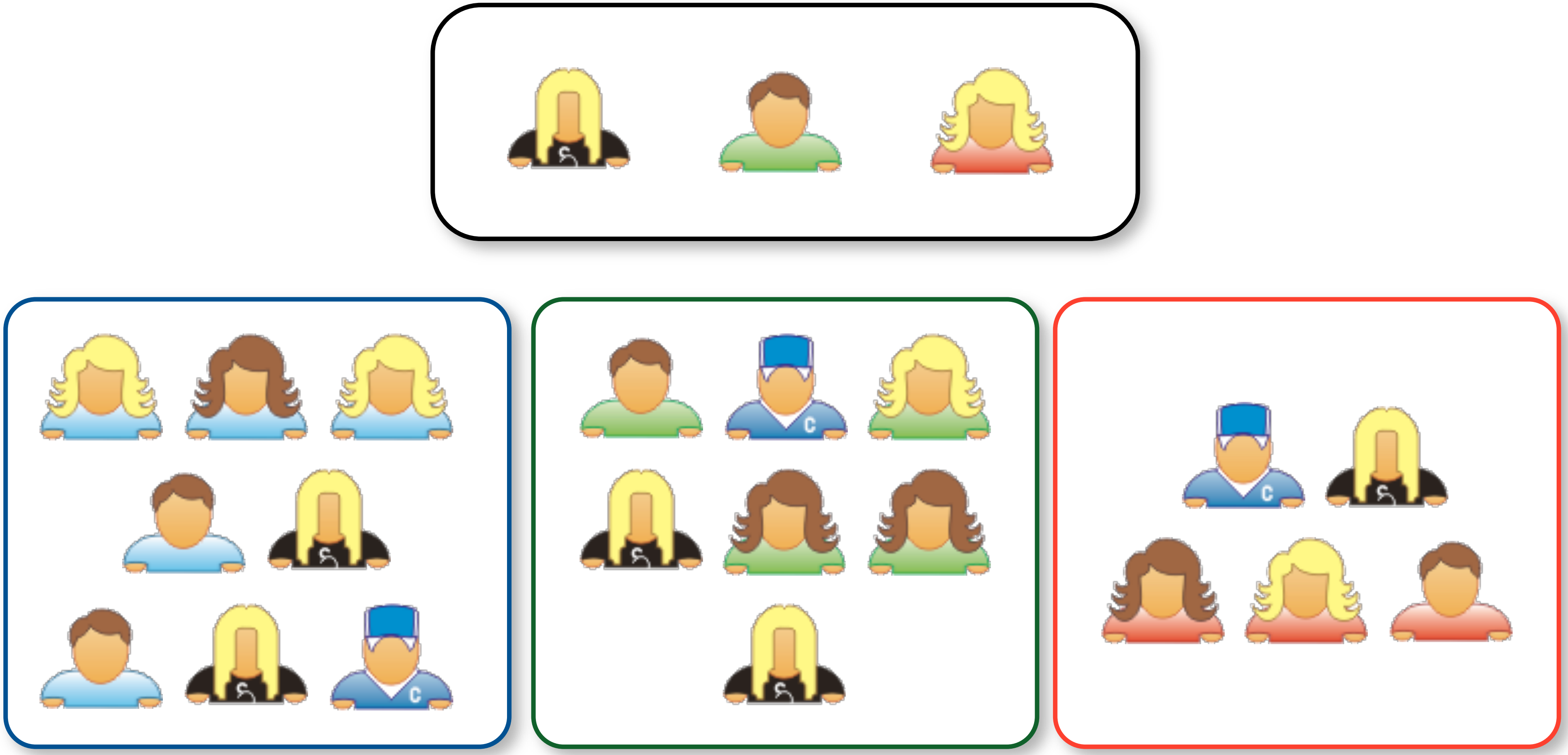
Impediment Log

- 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

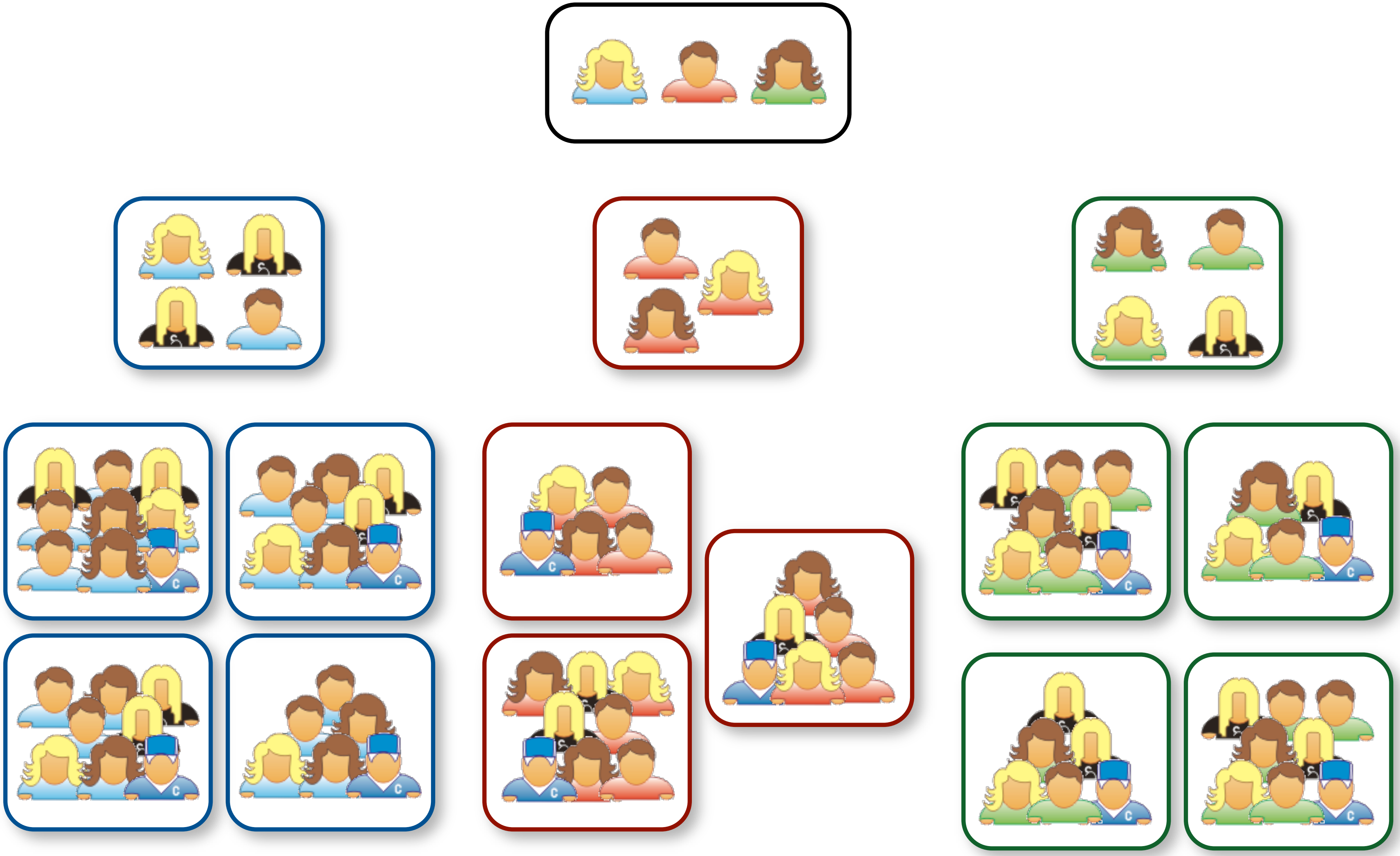
Scrum Scalability

- 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



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

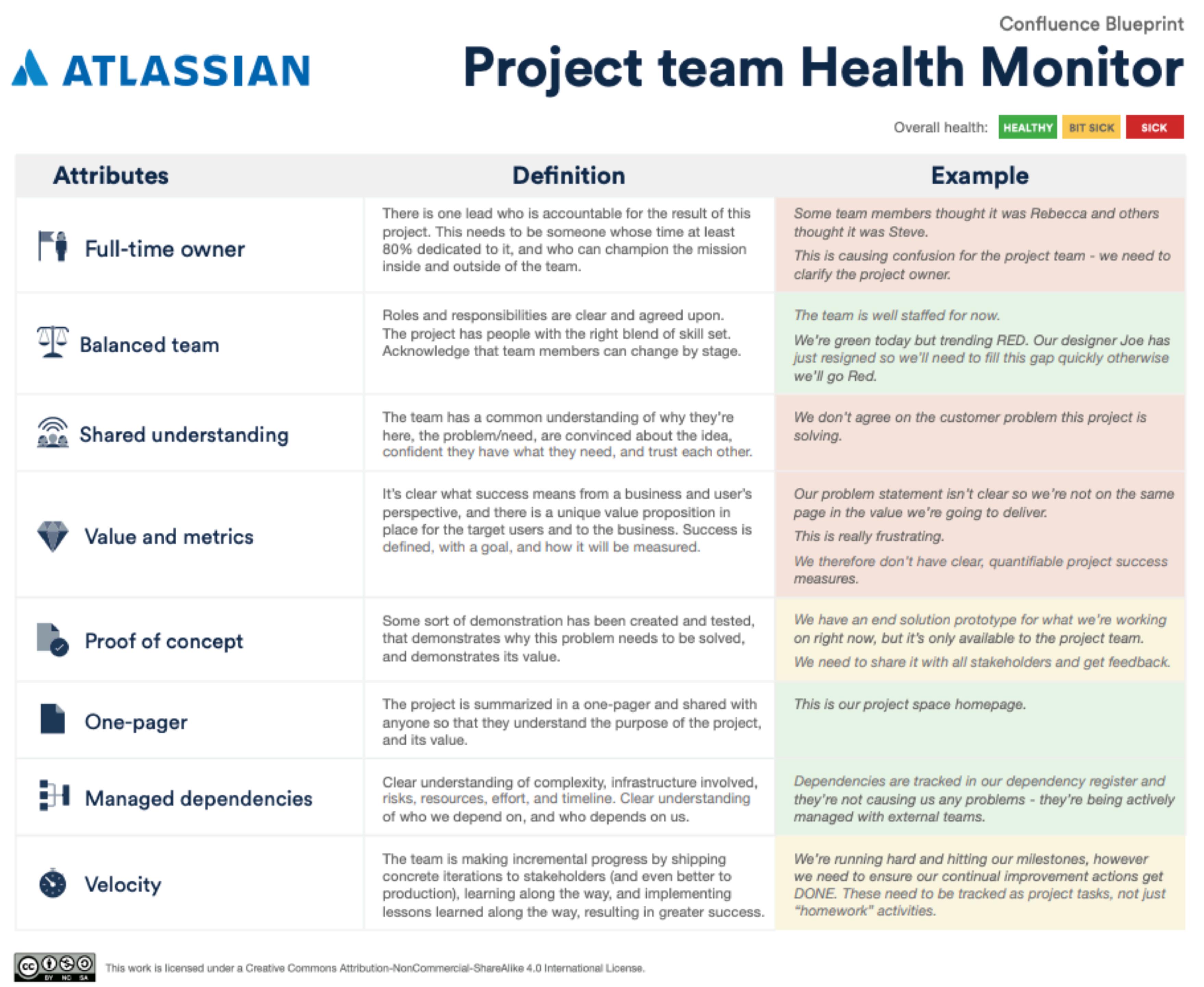
The Importance of Teams in Software Development

- Conway’s Law:
  - ❖ “Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.” — Melvin E. Conway

    https://youtu.be/QSKLm8E6KyE
- Team = Software/Product
  - ❖ “You can't have great software without a great team, and most software teams behave like dysfunctional families” - Jim McCarthy

Don’t Flip the Bozo Bit

https://youtu.be/QSKLm8E6KyE



Confluence Blueprint

ATLASSIAN

Project team Health Monitor

Overall health: 🟩 HEALTHY 🟨 BIT SICK 🟥 SICK

Attributes	Definition	Example
Full-time owner	There is one lead who is accountable for the result of this project. This needs to be someone whose time at least 80% dedicated to it, and who can champion the mission inside and outside of the team.	🟥 *Some team members thought it was Rebecca and others thought it was Steve.* *This is causing confusion for the project team - we need to clarify the project owner.*
Balanced team	Roles and responsibilities are clear and agreed upon. The project has people with the right blend of skill set. Acknowledge that team members can change by stage.	🟩 *The team is well staffed for now.* *We're green today but trending RED. Our designer Joe has just resigned so we'll need to fill this gap quickly otherwise we'll go Red.*
Shared understanding	The team has a common understanding of why they're here, the problem/need, are convinced about the idea, confident they have what they need, and trust each other.	🟥 *We don't agree on the customer problem this project is solving.*
Value and metrics	It's clear what success means from a business and user's perspective, and there is a unique value proposition in place for the target users and to the business. Success is defined, with a goal, and how it will be measured.	🟥 *Our problem statement isn't clear so we're not on the same page in the value we're going to deliver.* *This is really frustrating.* *We therefore don't have clear, quantifiable project success measures.*
Proof of concept	Some sort of demonstration has been created and tested, that demonstrates why this problem needs to be solved, and demonstrates its value.	🟨 *We have an end solution prototype for what we're working on right now, but it's only available to the project team.* *We need to share it with all stakeholders and get feedback.*
One-pager	The project is summarized in a one-pager and shared with anyone so that they understand the purpose of the project, and its value.	🟩 *This is our project space homepage.*
Managed dependencies	Clear understanding of complexity, infrastructure involved, risks, resources, effort, and timeline. Clear understanding of who we depend on, and who depends on us.	🟩 *Dependencies are tracked in our dependency register and they're not causing us any problems - they're being actively managed with external teams.*
Velocity	The team is making incremental progress by shipping concrete iterations to stakeholders (and even better to production), learning along the way, and implementing lessons learned along the way, resulting in greater success.	🟨 *We're running hard and hitting our milestones, however we need to ensure our continual improvement actions get DONE. These need to be tracked as project tasks, not just "homework" activities.*

Sources

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

An Agile/Scrum/Team reading list

- *Agile Estimating and Planning* by Mike Cohn
- *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
- *User Stories Applied for Agile Software Development* by Mike Cohn
- *Dynamics of Software Development by Jim McCarthy*

This Week

- Reading: User Stories Applied by Mike Cohn, Chapters 1-3 (available on the O'Reilly site)