W02-2:
Agile Deep Dive Part 1

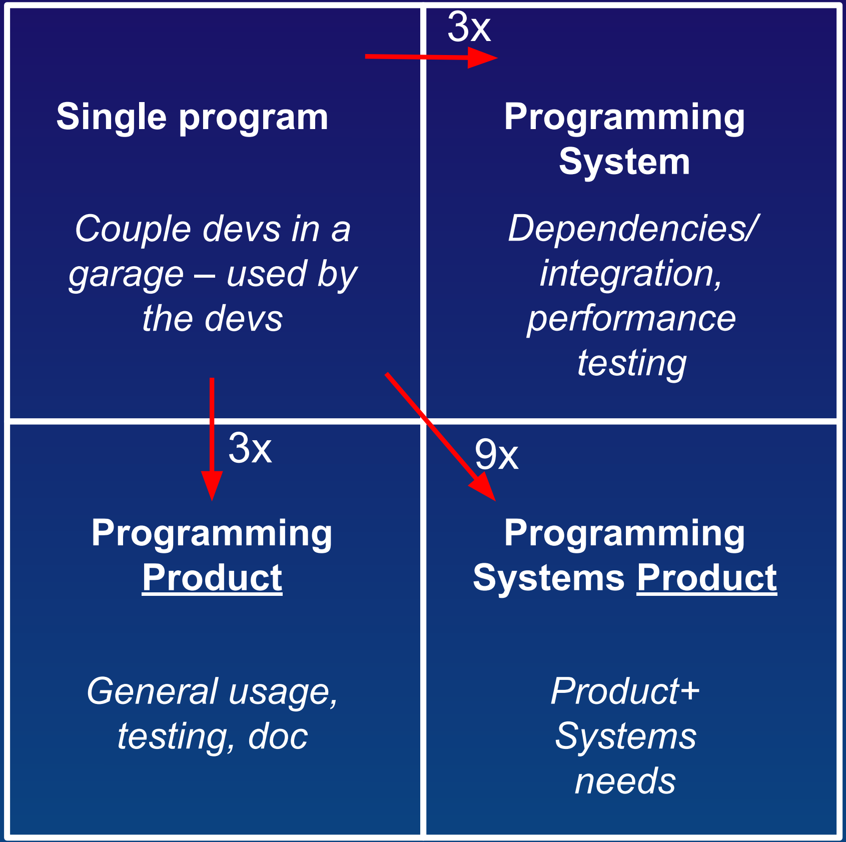
CPSC 3720 Software Engineering
Dr. Alex Adkins

Today’s Objectives

- Review last lesson

- Gain a deeper understanding of Agile & Scrum

The Tar Pit: Complexity of a Program vs. Product



3x

Single program

Couple devs in a garage – used by the devs

Programming System

Dependencies/ integration, performance testing

3x

9x

Programming <u>Product</u>

General usage, testing, doc

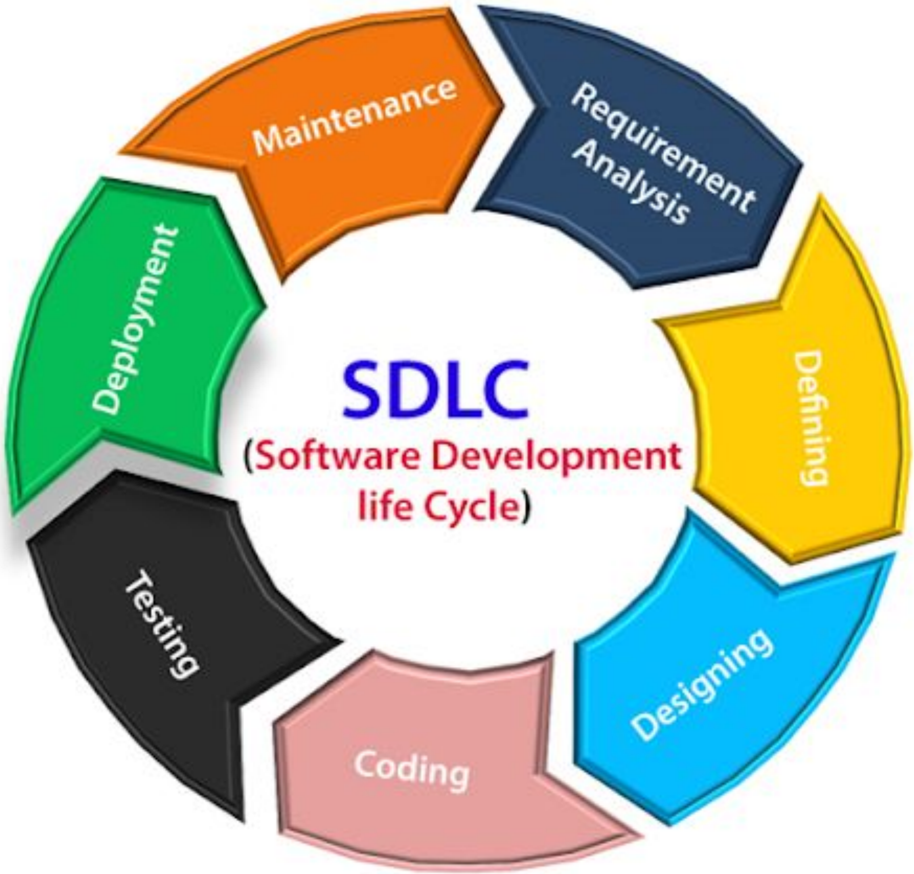
Programming Systems <u>Product</u>

Product+ Systems needs

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

How do we manage this complexity?

Software Development Process Steps



The Agile Manifesto

Individuals and interactions over Process and tools

Working software over Comprehensive documentation

Customer collaboration over Contract negotiation

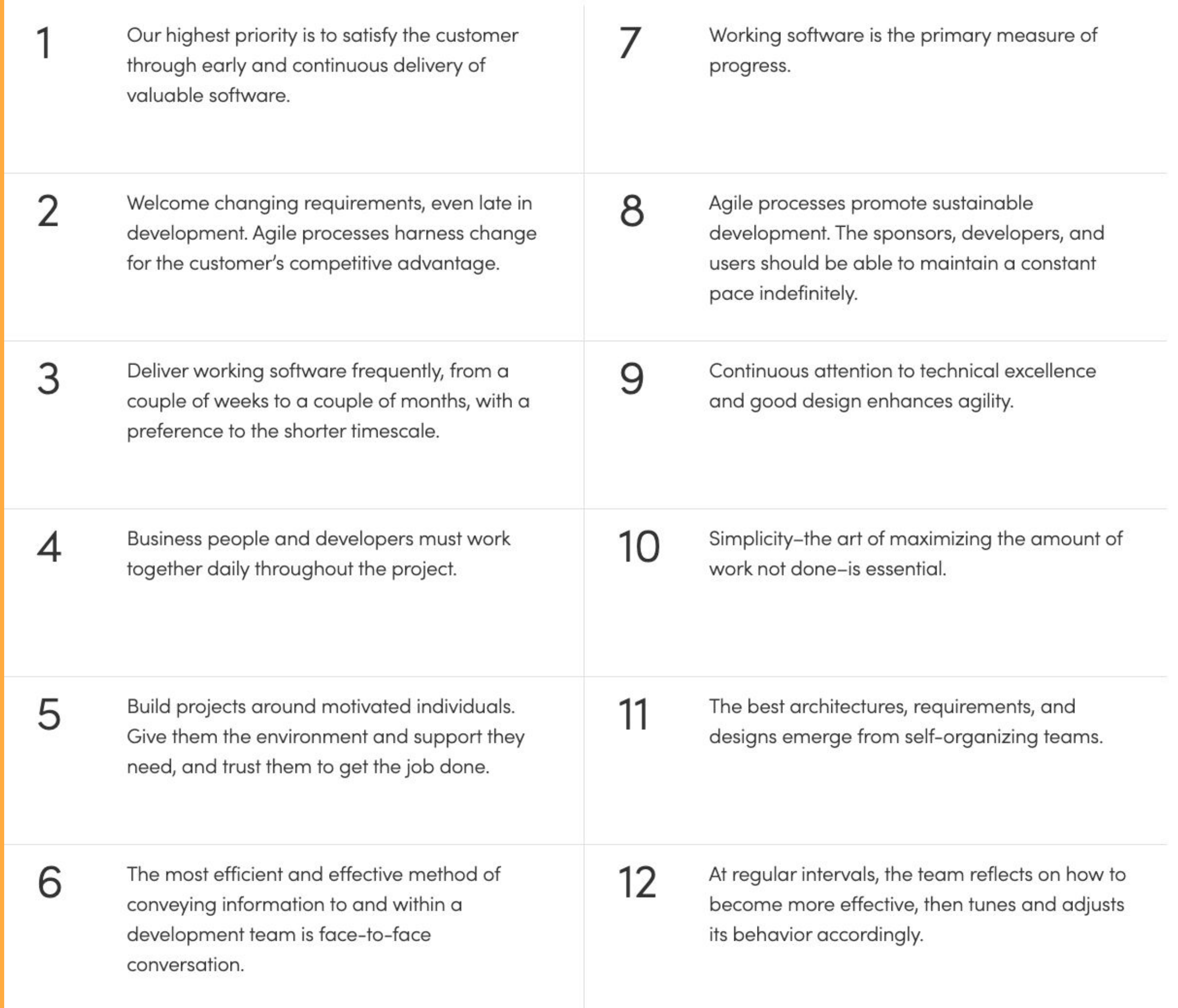
Responding to change over Following a plan

Source: www.agilemanifesto.org

Discussion: 12 Agile Principles

Which do you feel are the most important?

Which are the most challenging to accomplish?



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.

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.

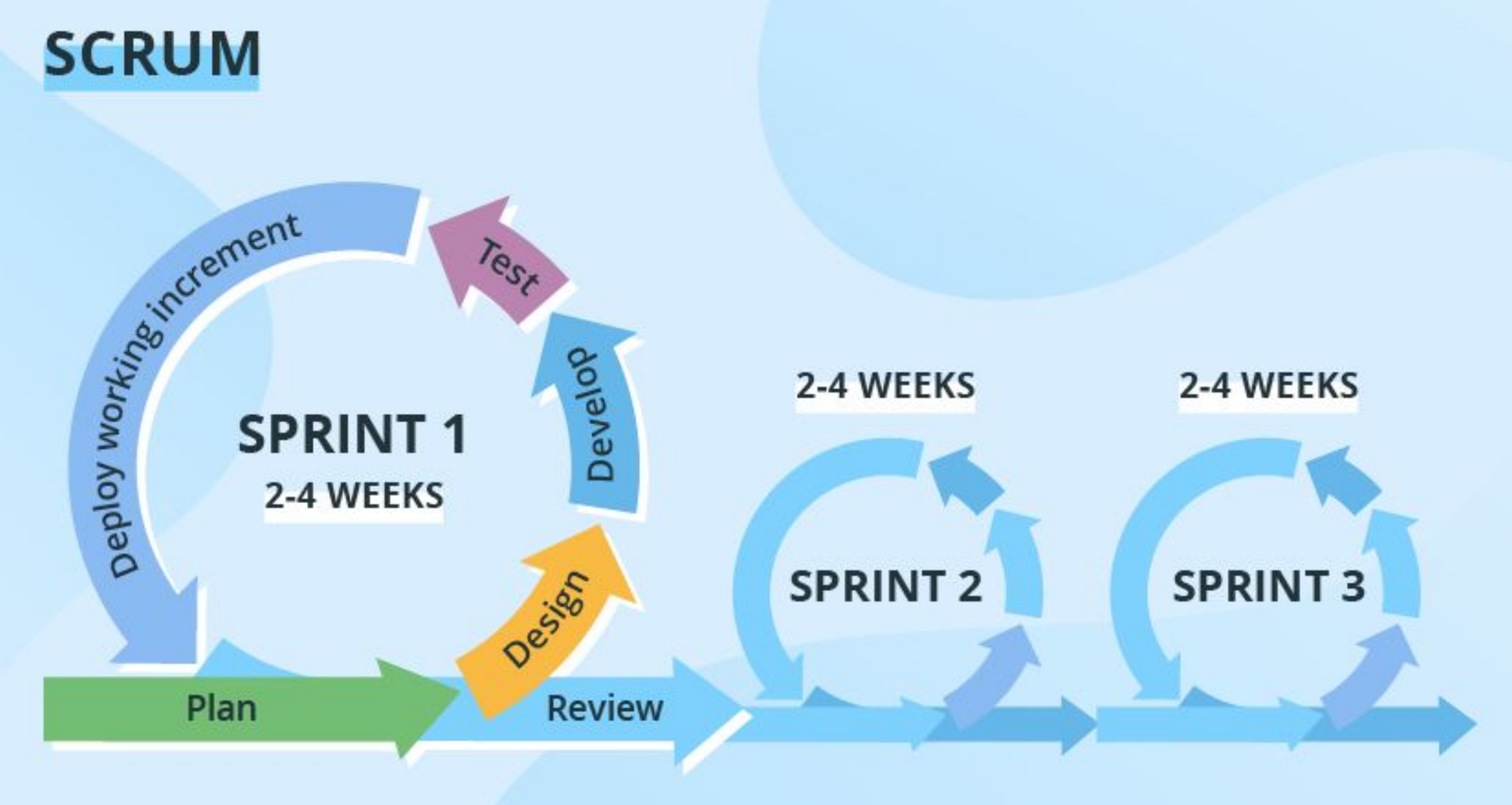
9 Continuous attention to technical excellence and good design enhances agility.

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.

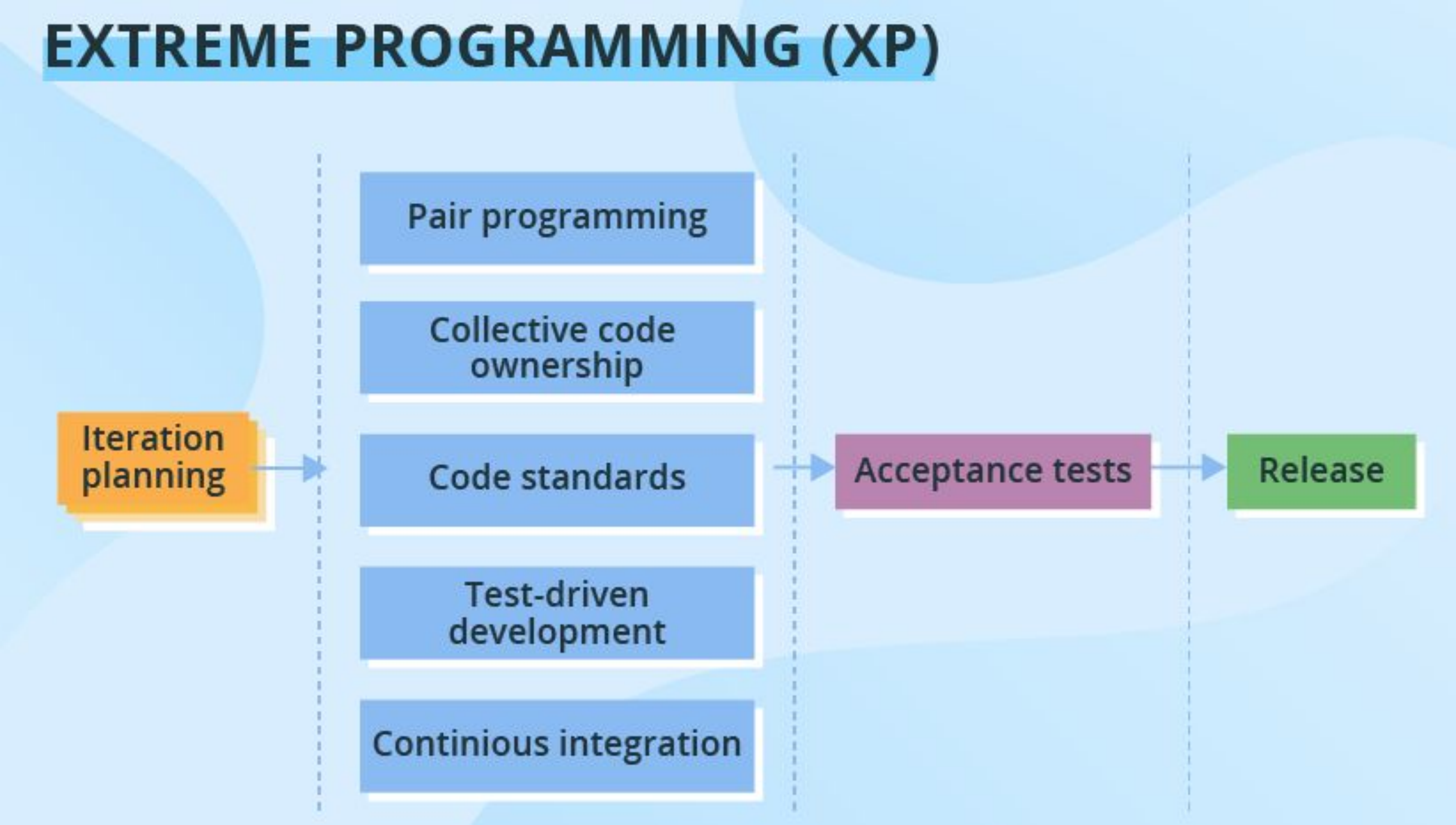
12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Process Model: Scrum



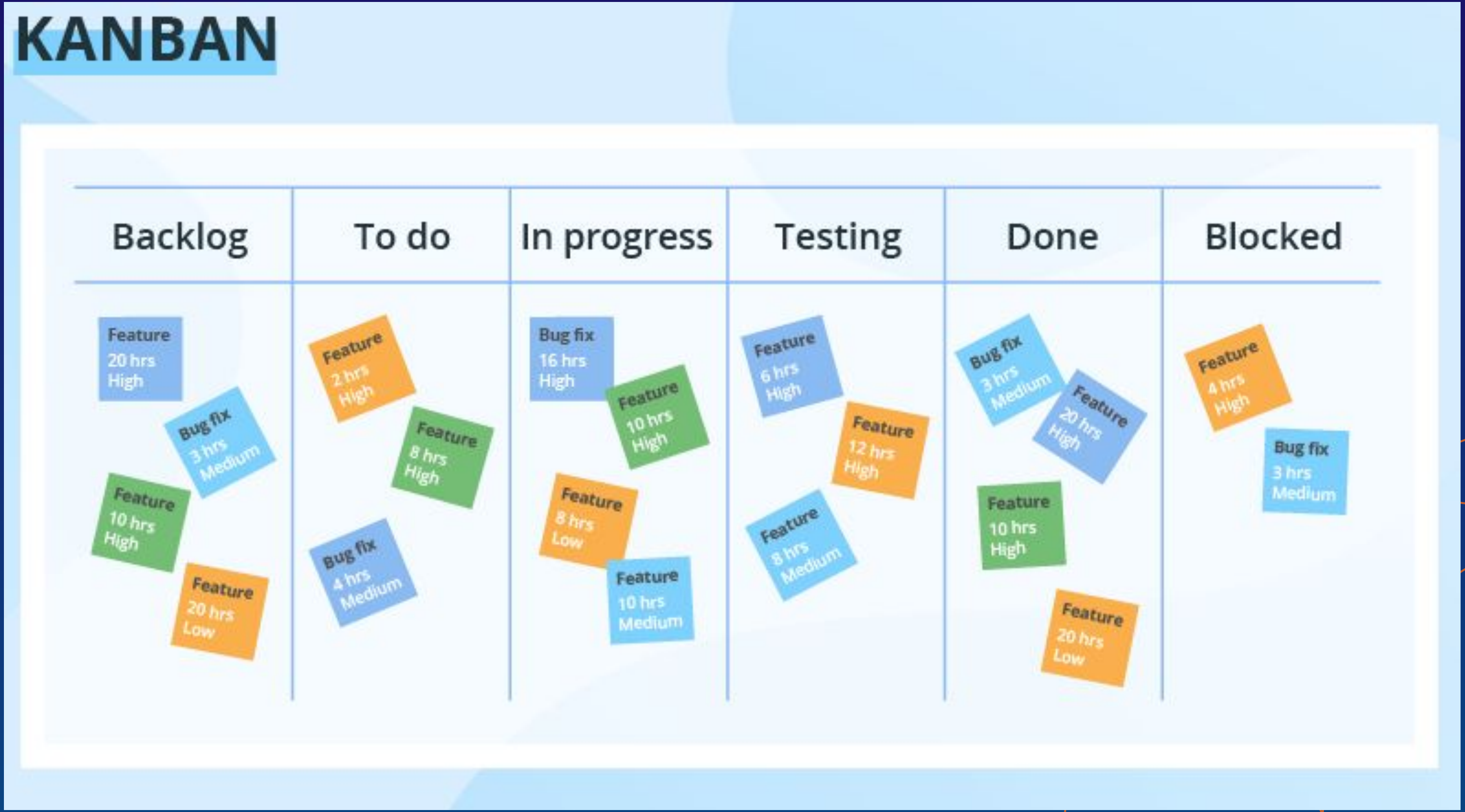
Source: https://www.scnsoft.com/blog/software-development-models

Agile Process Model: XP



Source: https://www.scnsoft.com/blog/software-development-models

Agile Process Model: Kanban



KANBAN

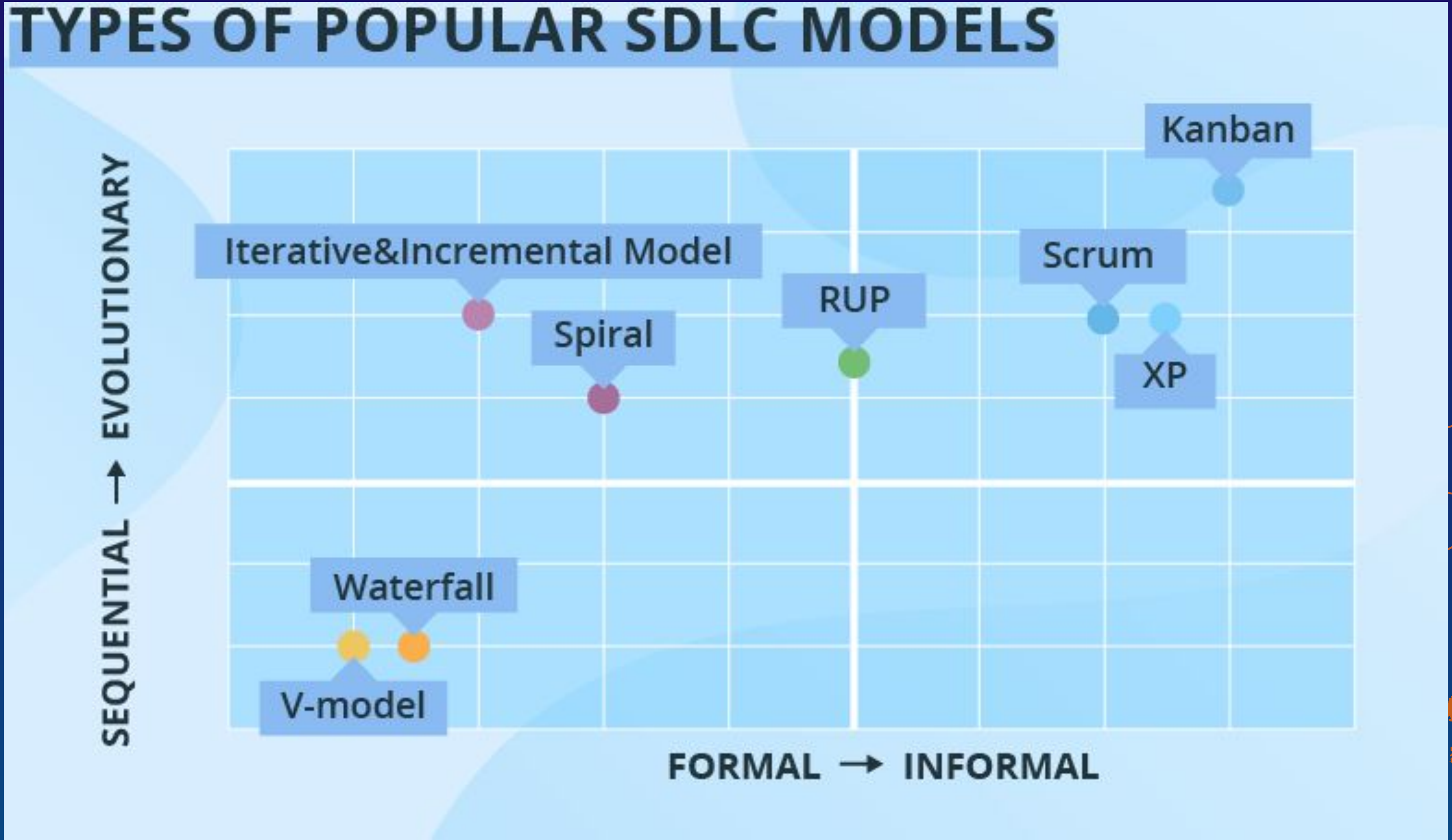
Backlog	To do	In progress	Testing	Done	Blocked
Feature 20 hrs High	Feature 2 hrs High	Bug fix 16 hrs High	Feature 6 hrs High	Bug fix 3 hrs Medium	Feature 4 hrs High
Bug fix 3 hrs Medium	Feature 8 hrs High	Feature 10 hrs High	Feature 12 hrs High	Feature 20 hrs High	Bug fix 3 hrs Medium
Feature 10 hrs High	Bug fix 4 hrs Medium	Feature 8 hrs Low	Feature 8 hrs Medium	Feature 10 hrs High	
Feature 20 hrs Low		Feature 10 hrs Medium		Feature 20 hrs Low	

Source: https://www.scnsoft.com/blog/software-development-models

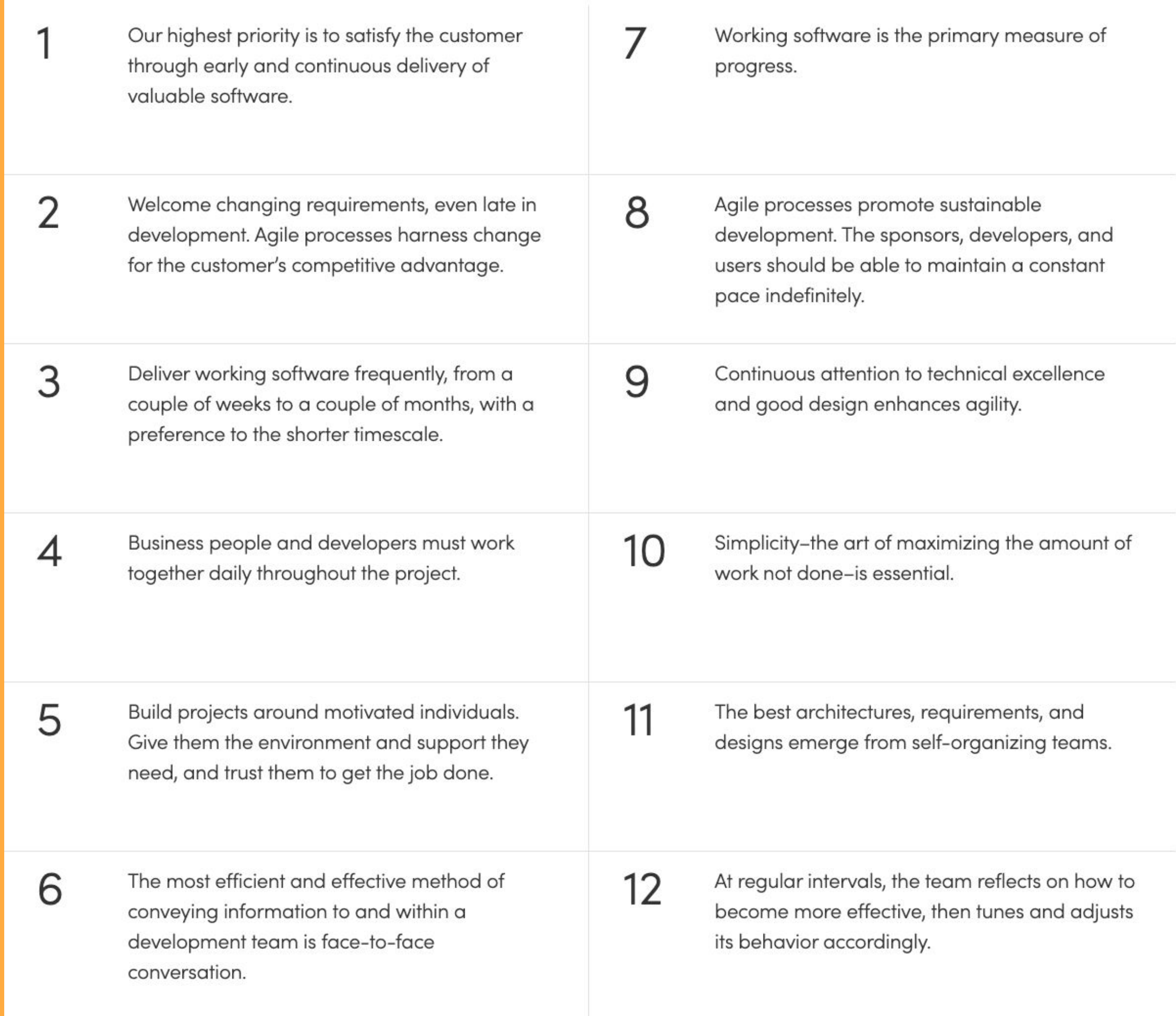
Agile Recap

- Agile methods are considered:
  - Lightweight
  - People-based rather than Plan based
- Multiple Agile Methods:
  - Scrum, XP, Kanban, Lean
- Agile Manifesto closest to a definition
  - Set of principles
  - Developed by Agile Alliance in 2001

Which SDLC Model to Use???



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.

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! The Anti-Agile Principles

- At your tables: Create an opposite principle to the one you are assigned. Make sure it sounds *like* an Agile principle, but is the opposite of your assigned principle.

- Pick someone to report back to the class stating your principle and anti-principle and answer the following questions
    - What does the Agile principle mean?
    - How is your Anti-Agile principle its opposite?

- 10 minutes

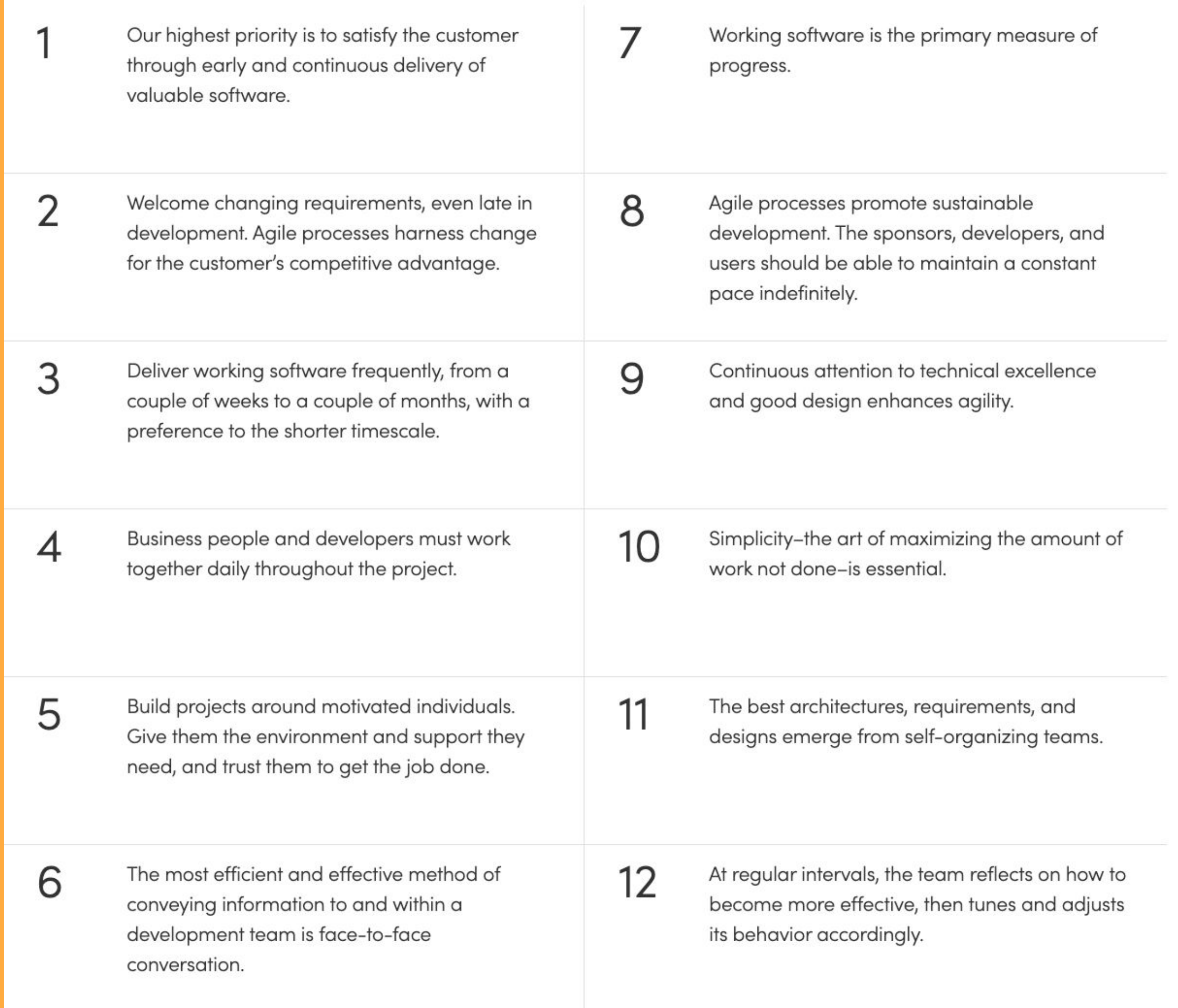
Breakout!
12 Agile Principles

Create an Anti-Agile Principle

Answer these questions:

- What does your Agile principle mean?
- How is your Anti-Agile principle its opposite?

Table	Principle	Table	Principle
1	1	5	8
2	2	6	9
3	4	7	11
4	7	8	12



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.

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.

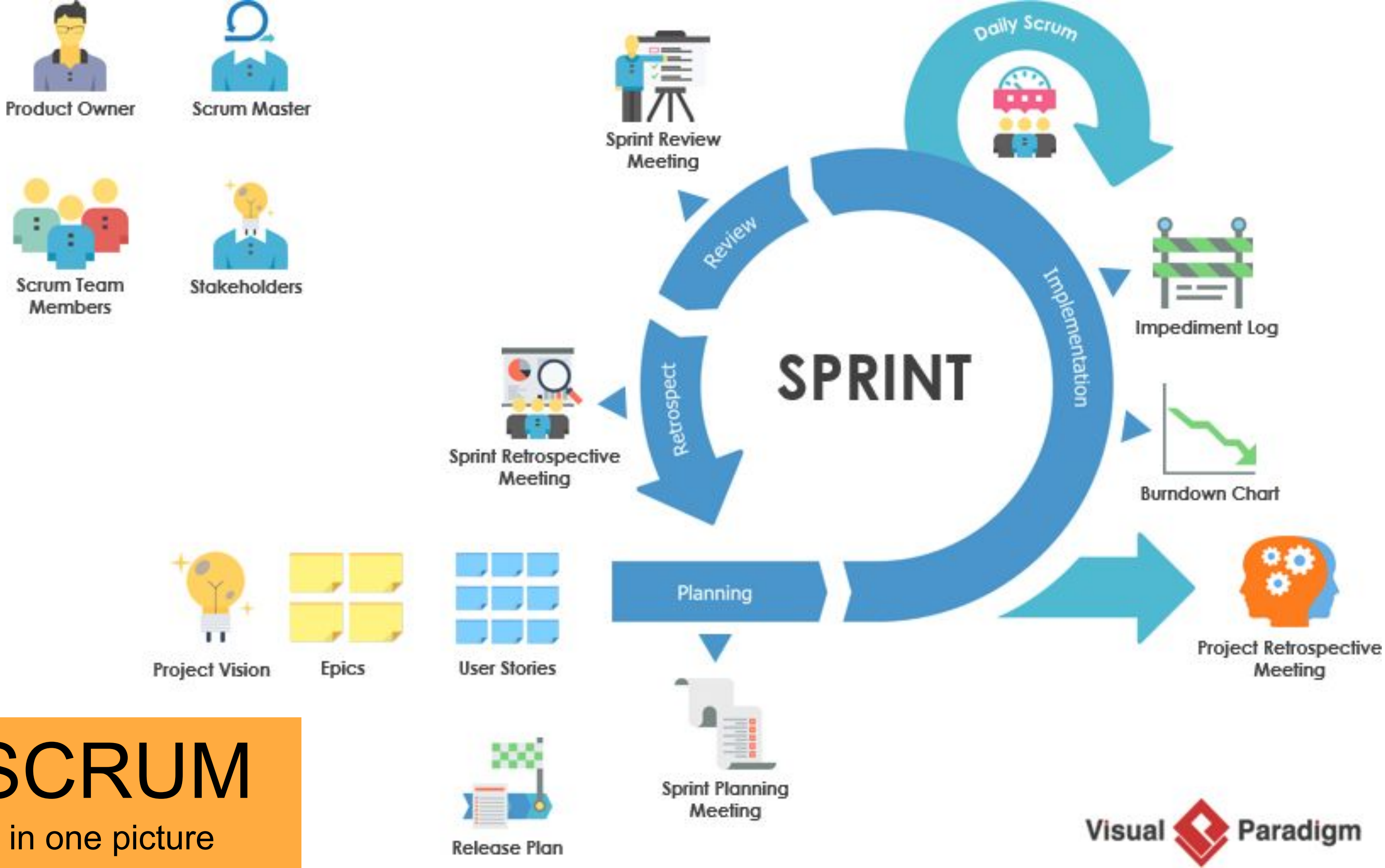
10 Simplicity–the art of maximizing the amount of work not done–is essential.

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.

Scrum

- “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

in one picture

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.
- At the end of each sprint anyone can see real working software and decide to release it "as is" or continue to enhance it for more sprints.

Sprints

- Scrum projects make progress in a series of “sprints” (sometimes called iterations)
- Typical duration is 2-4 weeks or a calendar month at most.
  - The shorter the better.
- A constant duration leads to a better rhythm.
- Product is designed, coded, and tested during the sprint.

No Changes during a Sprint

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

Scrum Pillars

Transparency

Inspection

Adaptation

Scrum Framework

Roles

- Product Owner
- Scrum Master
- Developers/Team

Ceremonies

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

Artifacts

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

Scrum Values

Focus

Respect

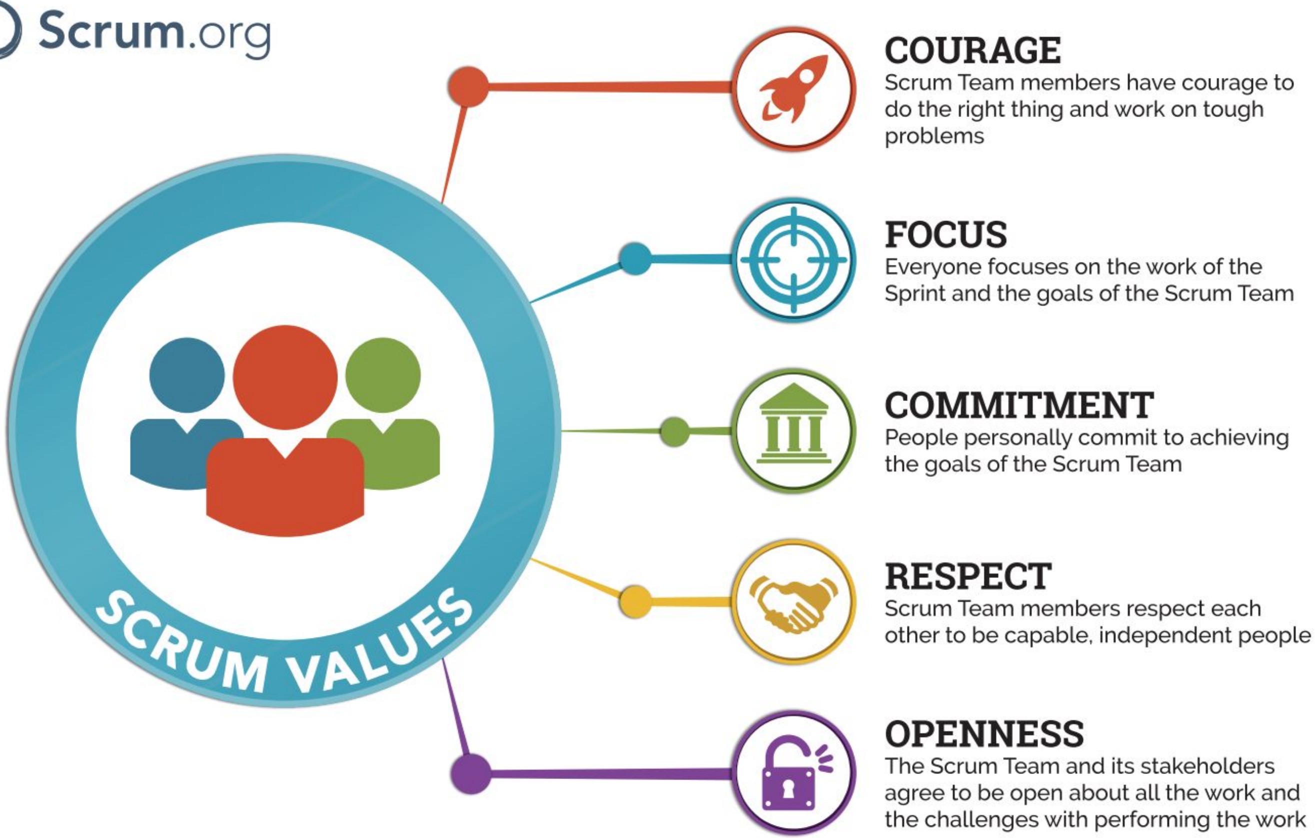
Courage

Commitment

Openness

Scrum Values

Scrum.org



COURAGE
Scrum Team members have courage to do the right thing and work on tough problems

FOCUS
Everyone focuses on the work of the Sprint and the goals of the Scrum Team

COMMITMENT
People personally commit to achieving the goals of the Scrum Team

RESPECT
Scrum Team members respect each other to be capable, independent people

OPENNESS
The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work

Scrum Values

Commitment

Focus

Openness

Respect

Courage

Great teams embrace behaviors that adhere to these values and recognize and eliminate anti-patterns

Let’s play a game:
https://sevawisegames.com/games/scrum-values

Scrum Values

Roles

- Product Owner
- Scrum Master
- Developers/Team

Ceremonies

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

Artifacts

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

Scrum Values

- Courage
- Focus
- Commitment
- Respect
- Openness

Scrum Roles

Roles

- Product Owner
- Scrum Master
- Developers/Team

eremonies

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

Artifacts

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

Scrum Values

- Courage
- Focus
- Commitment
- Respect
- Openness

Roles: Product Owner

- 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
- Sometimes called a Product Manager

Roles: The ScrumMaster

- Project management focus
- 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

Roles: The Team/Developers

- Typically 4-9 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
- Membership should change only between sprints but team consistency is best

Scrum Roles

Developers

Responsible for developing the product. Each Developer is co-equal and contributes in whatever way necessary to complete the iteration.

Scrum Master

Responsible for making sure Scrum Team lives by the values and practices of Scrum. Considered the team coach and helps the team to be successful.

Product Owner

Responsible for the iteration scope. Shares product vision of what is to be built and communicates vision to the Scrum Team.

Let’s play a game: https://sevawisegames.com/games/scrum-values

Scrum Artifacts

Roles

- Product Owner
- Scrum Master
- Developers/Team

Ceremonies

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

Artifacts

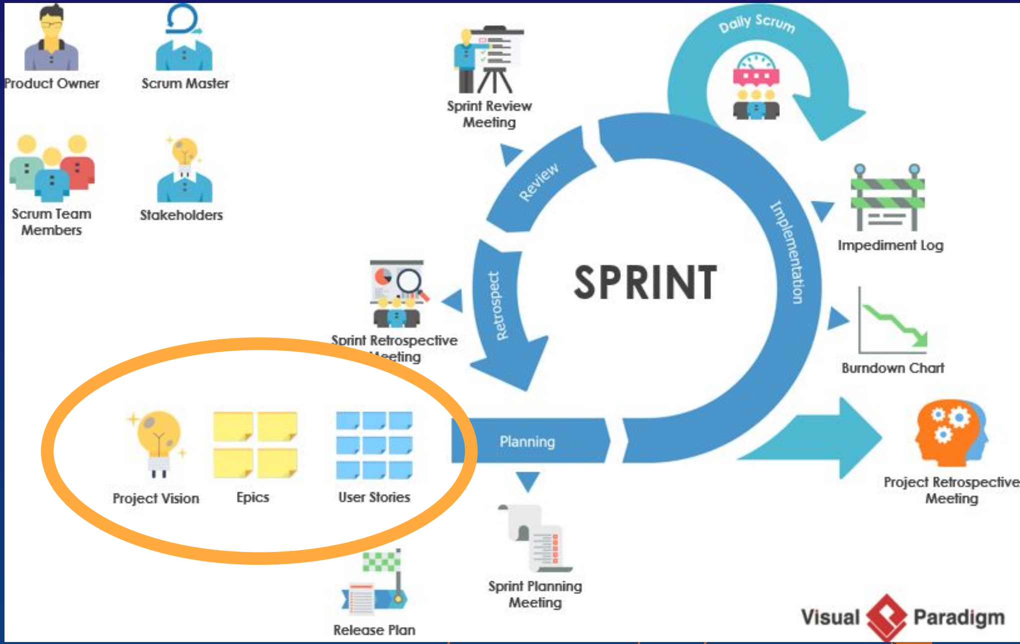
- Product backlog
- Sprint backlog
- Burndown charts
- Impediment Log

Scrum Values

- Courage
- Focus
- Commitment
- Respect
- Openness

Artifact: Product Backlog

- The requirements represented as Epics and Stores desired for 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 and used to create the next sprint backlog

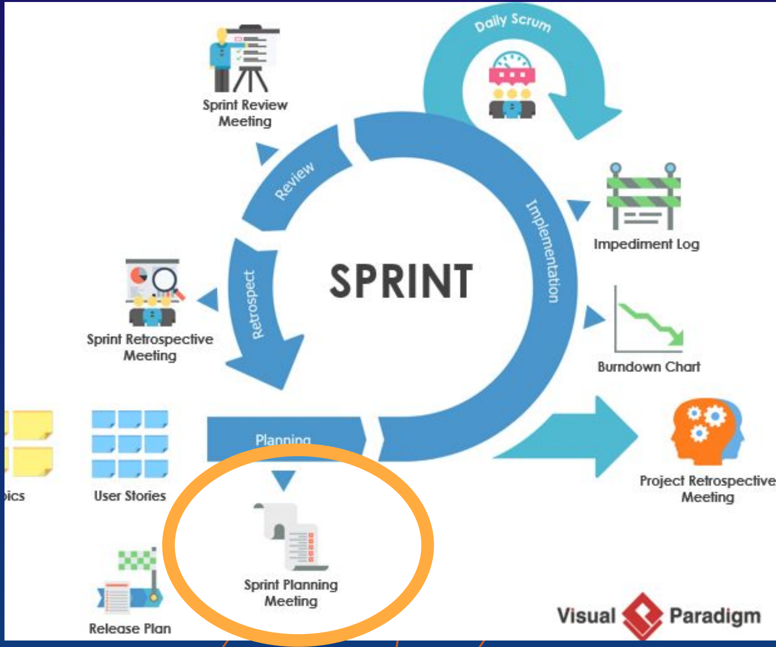


Example Product Backlog

Backlog Items	Storypoint Estimate
As a guest, I can make a reservation	50
As a guest, I want to cancel a reservation	30
As a guest, I want to change the dates of a reservation	15
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	30
…	30

Artifact: Sprint Backlog

- The stories that are chosen to be delivered for a particular Sprint as prioritized by the Product Owner
- Stories for a sprint should be “developer ready”
- The task breakdown for teach story is done in the Sprint Planning ceremony
- Sprints are in storypoints and tasks are estimated in days/hours
- The scrum team will commit to the sprint backlog to be completed in that sprint

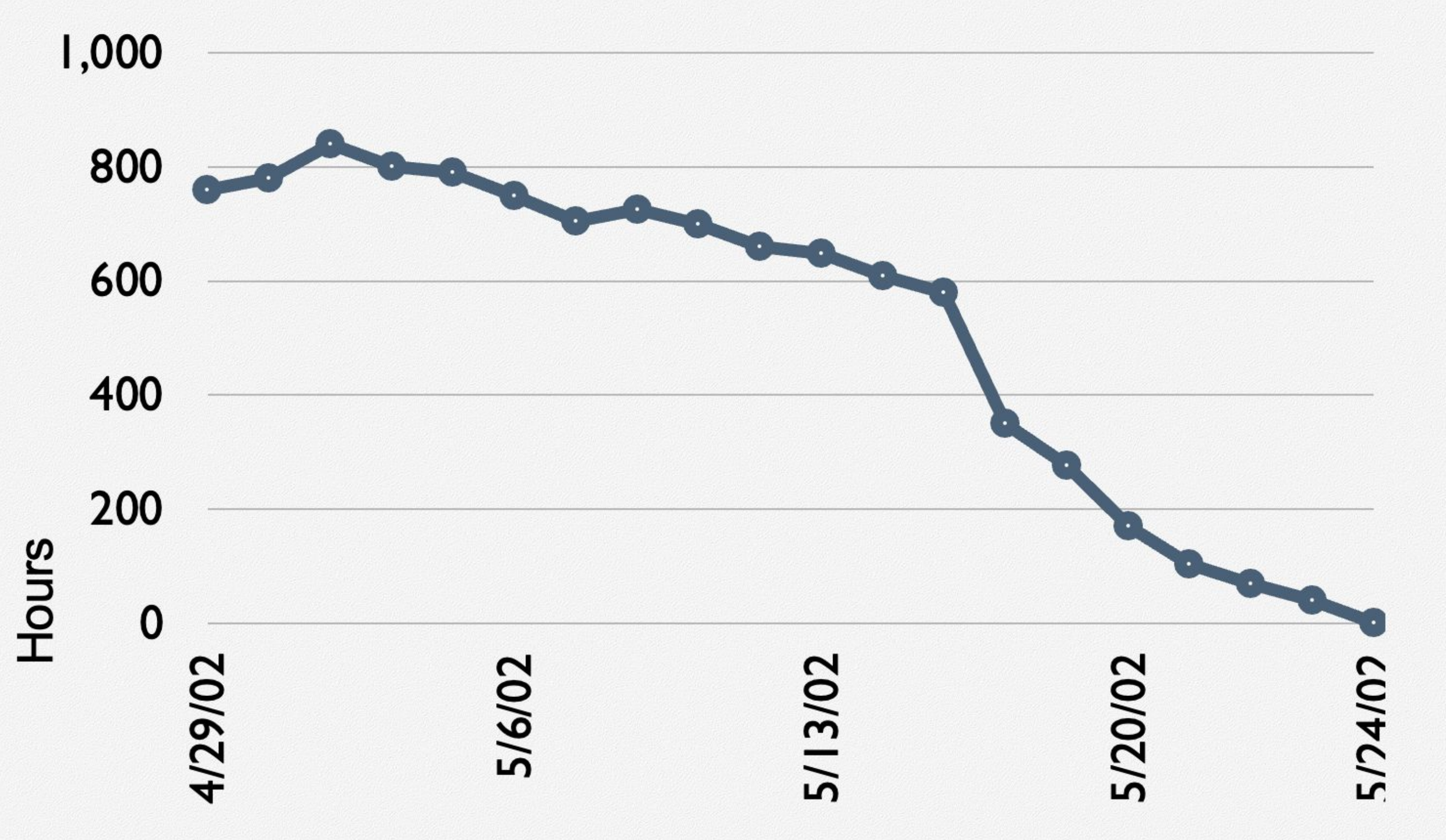


Artifact: Sprint Backlog

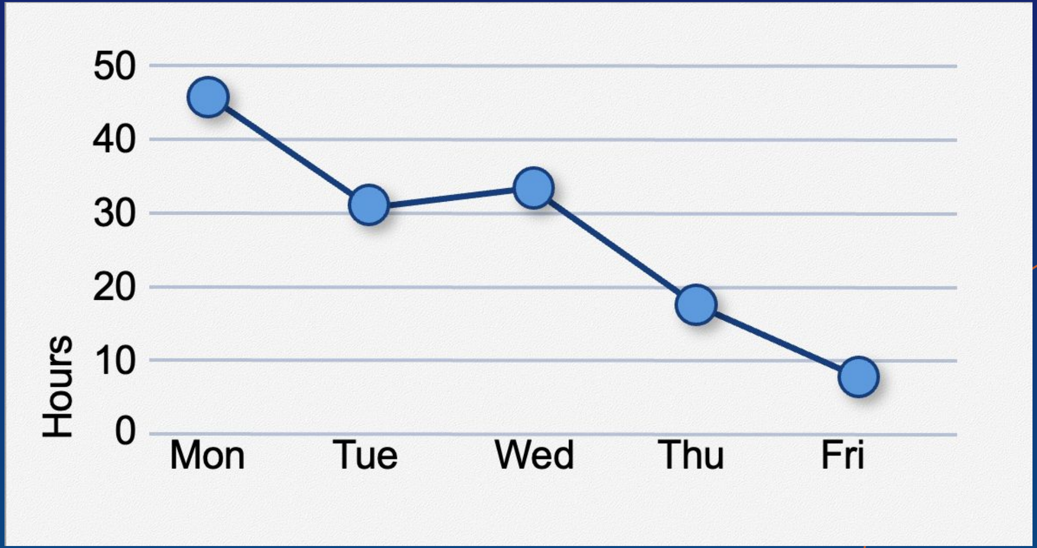
Backlog Items	Storypoint Estimate
As a g	
As a g	
As a g	
As a h (rever	
…	

Task	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	

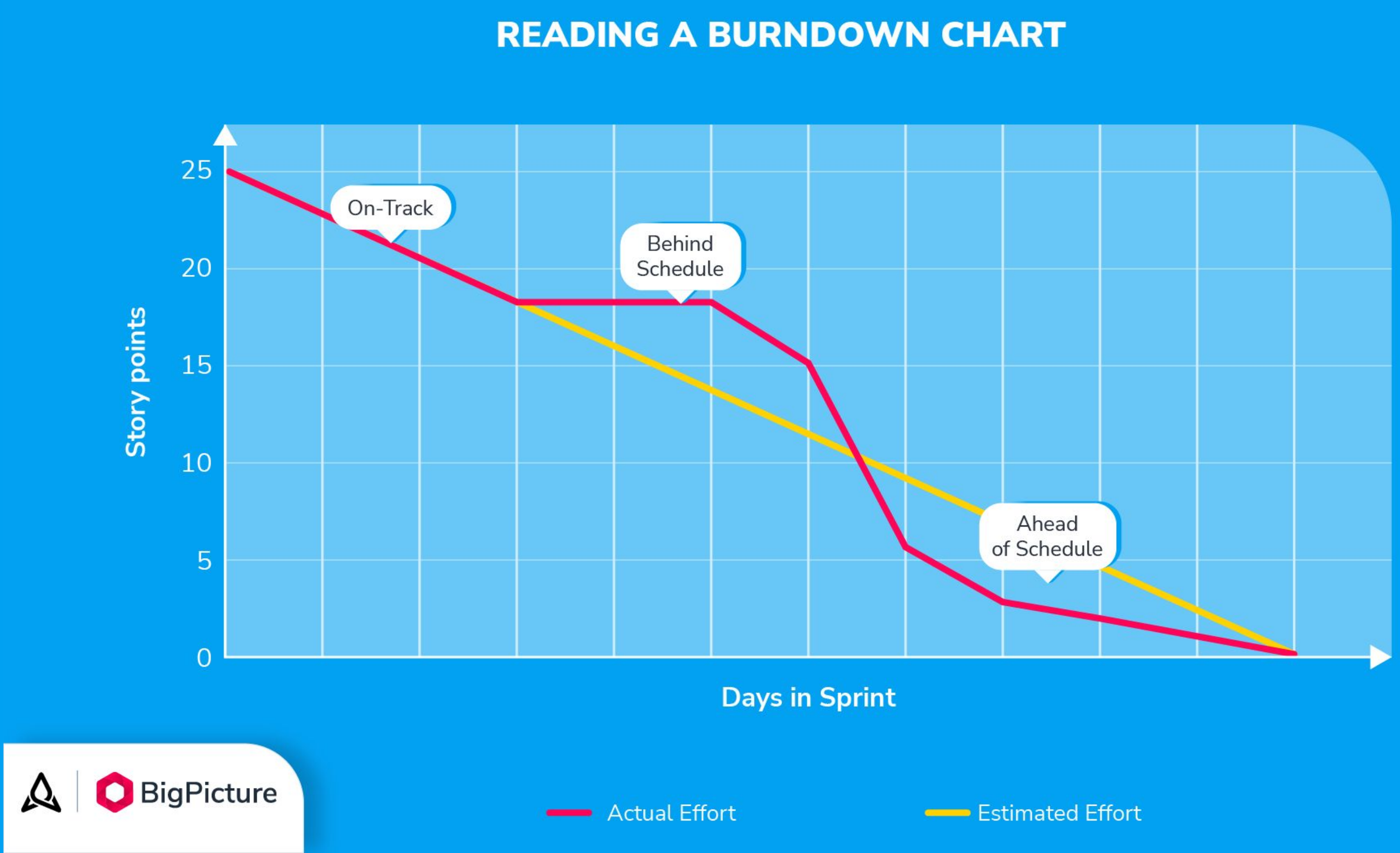
Artifact: Sprint Burndown Chart



Task	M	Tu	W	Th	F
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	



Reading a Sprint Burndown Chart



Artifact: 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. Dev keeps getting pulled into code reviews for other teams
    - A team member is not showing up to daily standup

Scrum Ceremonies

Roles

- Product Owner
- Scrum Master
- Developers/Team

Ceremonies

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

Artifacts

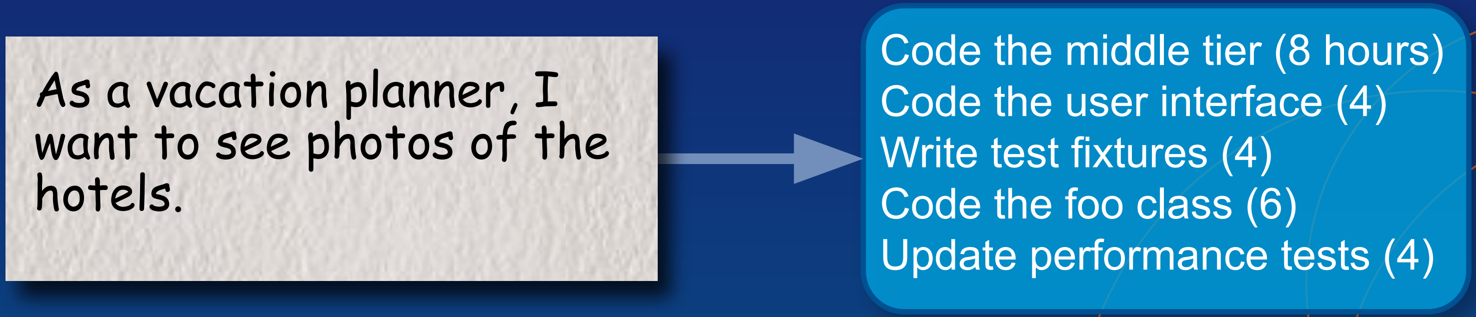
- Product backlog
- Sprint backlog
- Burndown charts
- Impediment Log

Scrum Values

- Courage
- Focus
- Commitment
- Respect
- Openness

Ceremony: Sprint Planning

- Teams select 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 scrum Master



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)

Product Backlog Item

Sprint Backlog

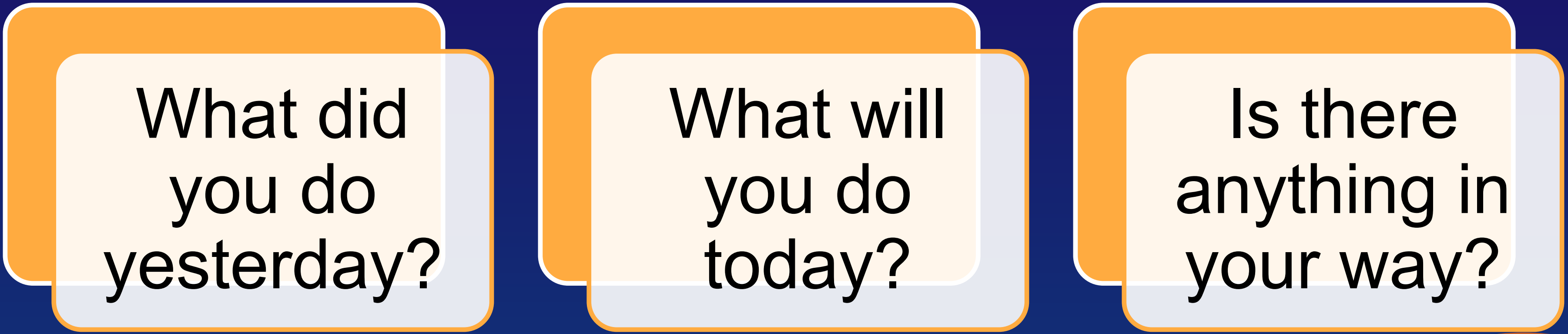
Ceremony: The Daily Scrum (Stand-Up)

- Parameters:
  - Daily
  - 15 minutes
  - Stand-Up

- Not for problem solving
  - Whole world can be invited, BUT:
  - Only team members, scrum master, product owner can talk

- Helps avoid unnecessary meetings

Daily Scrum: Everyone Answers 3 Questions



What did you do yesterday?

What will you do today?

Is there anything in your way?

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

The Daily Scrum

https://youtu.be/oHcmLKroPqw

The Daily Scrum

https://youtu.be/oLmDe8pAc6I

checkin.cs.clemson.edu:

Stand-Up!

At your tables
Groups of 3-4
Answer:

- What did you do yesterday?
- What will you do today?
- Is there anything in your way?

Ceremony: 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

Ceremony: 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.

Sprint Retrospective: The Three L's

The whole team gathers and discusses what they'd:

Loved

Loathed

Learned

This is what we'll use for the class project

Scrum Management

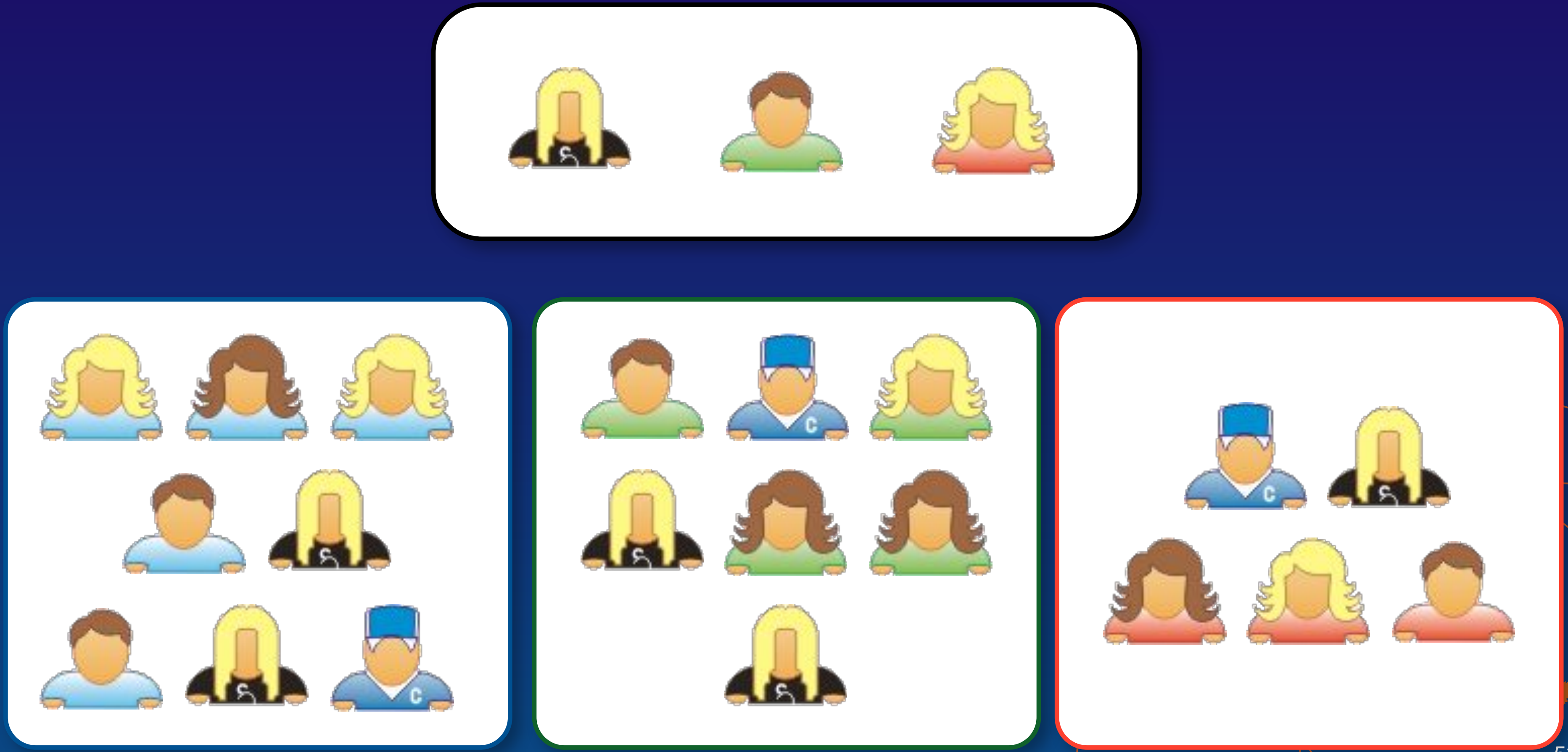
- The Scrum process is typically managed using Agile product management tools such as:
    - Atlassian products – Jira, Confluence, etc.
    - Trello
    - Microsoft Azure DevOps
    - Monday.com

These tools will integrate with Git and a chat tool such as Microsoft Teams or Slack

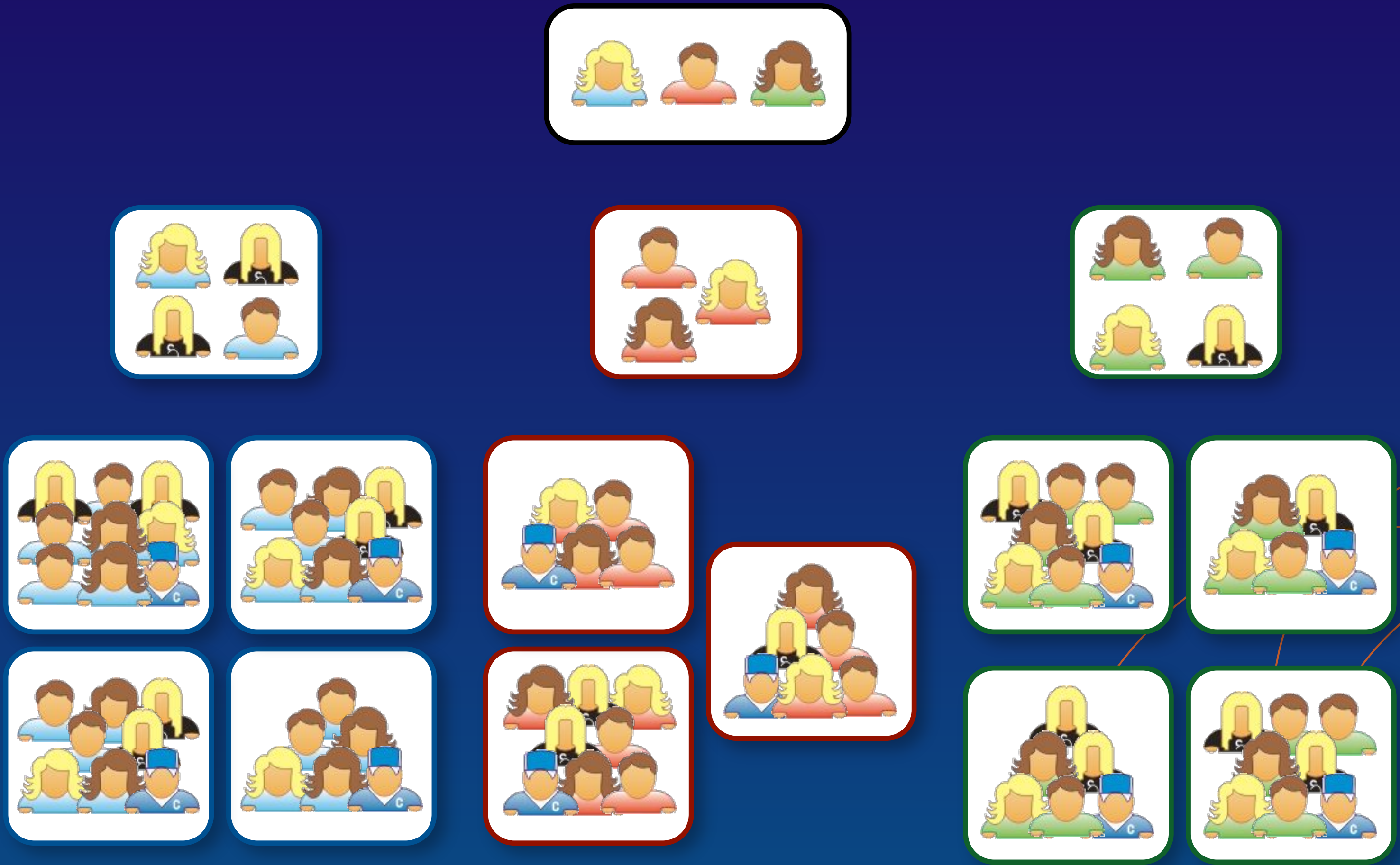
Scrum Scalability

- 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



Think…

Take a couple of minutes to think…

- What about software engineering do you enjoy? What interests you?
- What skills do you want to develop?
- What is your ideal career/job in the field?

Next Time

- More Agile and Scrum!
- Games!!
- Quiz 1 (Lessons 1-4) on Tuesday Jan 28
    - ~15 minutes, closed note, 35 points
    - Will be on your computer – take the tech test quiz

Sources

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