# CP317 Software Engineering

week 10-1: Agile, XP, and SCRUM

Shaun Gao, Ph.D., P.Eng.

# Agenda

- RAD cont.
  - Methodology aspect
  - Management aspect
  - Tools aspect
- Agile software development
  - Concept
- Agile manifestos
- Extreme programming (XP)
  - Concept
  - XP values
  - XP roles and practice
- SCRUM
  - Concept
  - Scrum framework
- Summary

# Essential aspects of RAD

- RAD is a software development methodology that enables organizations to develop strategically important software systems faster while reducing development costs and maintaining quality.
- It is also a type of iterative models
- Key elements: (1) proven technologies (2) CASE tools
- RAD has four essential aspects:
  - Methodology
  - People
  - Management
  - Tools

# Methodology aspect

- Fundamentals of RAD methodology:
  - Requirement gathering through workshops, facilitated meetings
  - Use evolutionary prototypes (iterative approach) that are eventually transformed into the final product
  - Requirement validation through iterated prototypes and constant customer testing of designs
  - Constant integration and testing of new code into the software product
  - Select a set of CASE tools to support modeling, prototyping and code reusability, as well as automating many of the combinations of techniques
  - Timeboxing which is for setting a tight delivery schedule for producing the prototype of software products

# People and Management aspect

- In order to be success, management must pay careful attentions to human motivation
- Managers should target professionals who see the value of new methodology and are eager to learn, problem solvers, and meticulous developers
- Team members are excited about the new methodology and they want to make it work well in their environment
- The success of RAD depends on the involvement of people with right skills and talents
- Team members must be carefully selected, highly trained and highly motivated
- They must be able to use the tools and each of them must be a team player

# Tools aspect

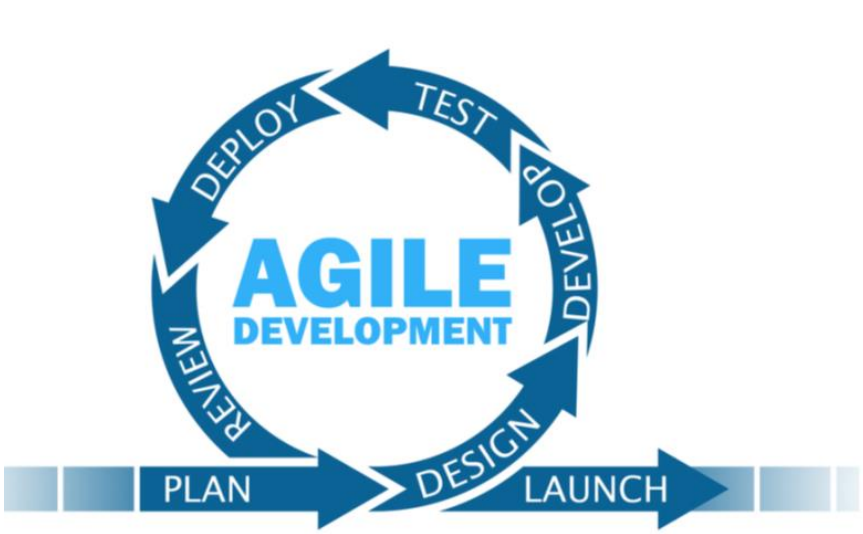
- RAD methodology uses both computerized tools and human techniques to achieve the goals of high-speed and high quality
- Examples of tools that can be used in RAD projects are CASE tools. These tools play a key role
- Examples of tools:
  - AI - ChatGPT
  - Integrated development environment (IDE)
    - Rational Unified Process Model
    - Visual studio, and etc.

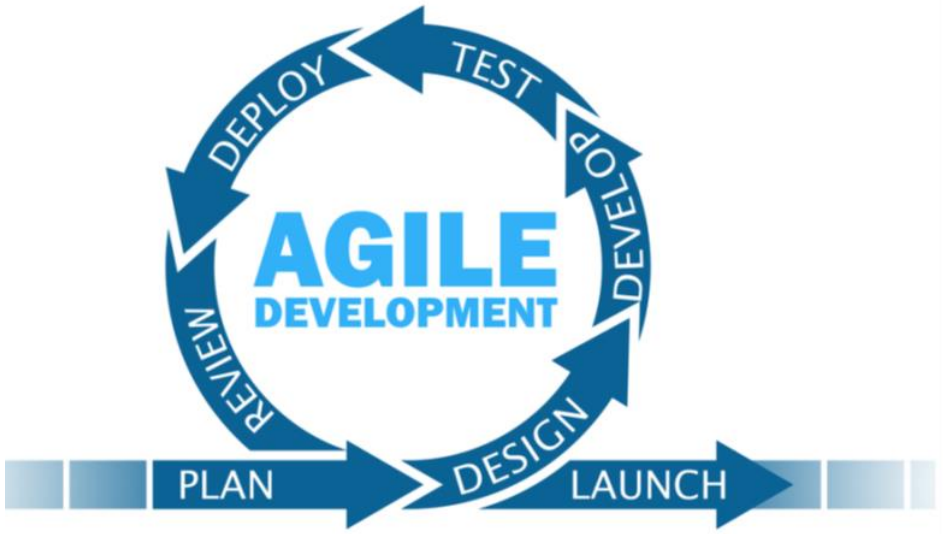
# Advantages and disadvantages of RAD

- Advantages
  - Shorten project time
  - Risk mitigation. Before each iteration, look for potential risks, handle them
  - Greater chance of success
- Disadvantages
  - CASE tools can be expensive, need effort for training developers
  - Require sufficient number of Human Resources
  - Loosely documented
  - Lack of design review and code review activities
  - Software may contains more vulnerabilities
  - Not suitable for safety critical systems

# Agile software development

- Agile --readiness for motion, nimbleness,  activity, dexterity in motion
- **Agile software development** is a **conceptual** framework for software engineering  that promotes development iterations throughout the life-cycle of the project.
- Proposed by a group of software developers/engineers in late 1990s



# Manifesto of Agile Software Development

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
   (working software may not enough now)
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

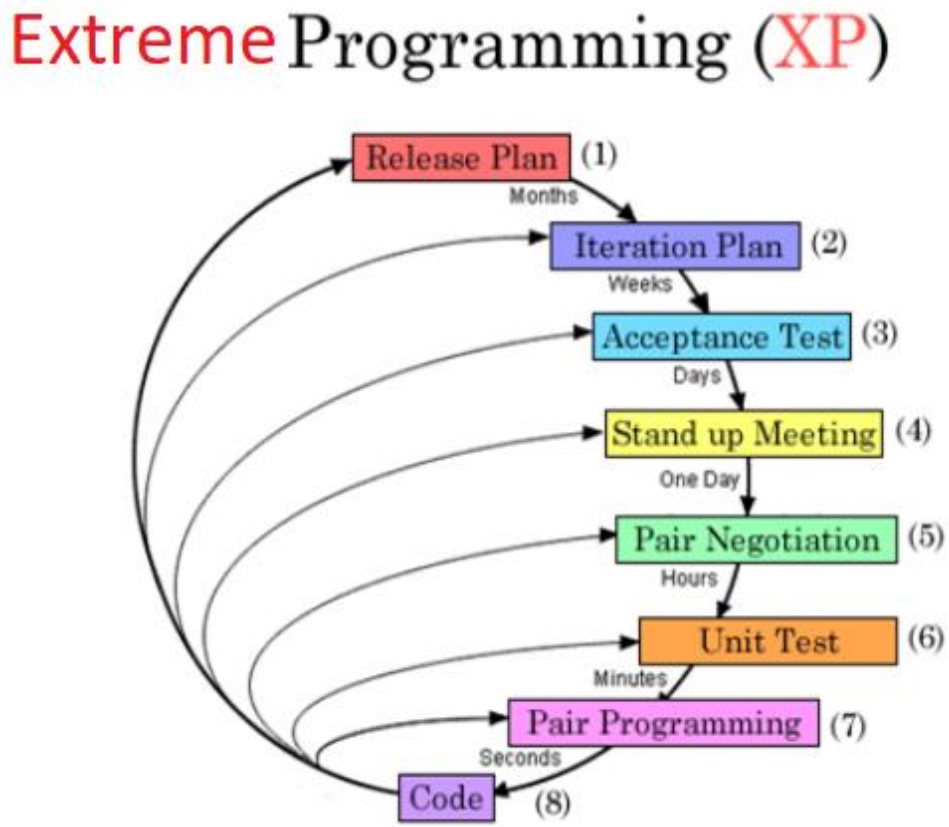
Agile is more a set of guidelines than an actual development model

Reference: https://www.agilealliance.org/

**Two concrete examples of Agile: XP and Scrum**

# Extreme programming (XP)

- Extreme programming(XP) is a software development methodology that intends to improve software quality and responsiveness to changing customer requirements.
- It is a type of Agile software development, proposed in 1999 by Kent Beck, Ward Cunningham and Ron Jeffries.

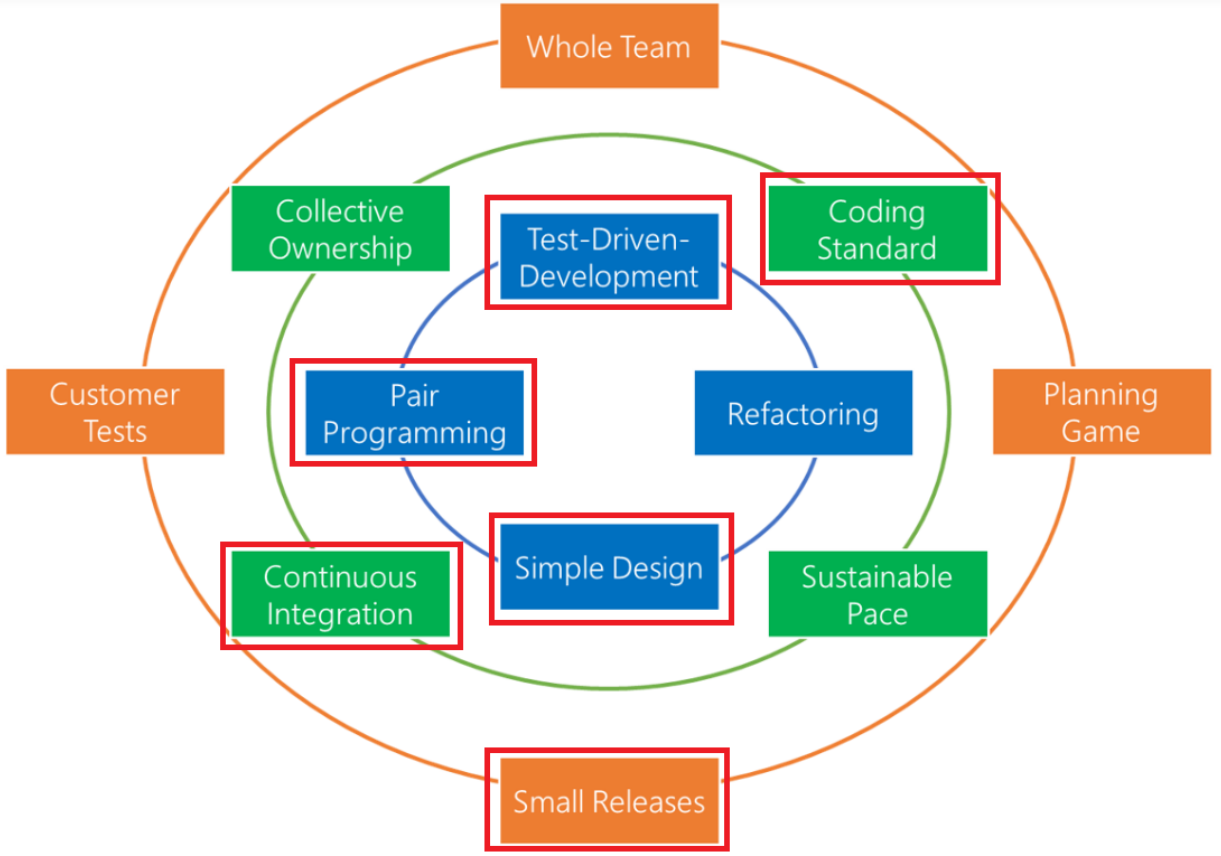


# XP values

- Communication
  - Requirements understanding
- Simplicity
  - XP encourage simple designs
- Feedback
  - From unit and integration testing; from customers
- Courage
  - Start with simple solution; refactor code when necessary
- Respect
  - Respect each other – professionalism

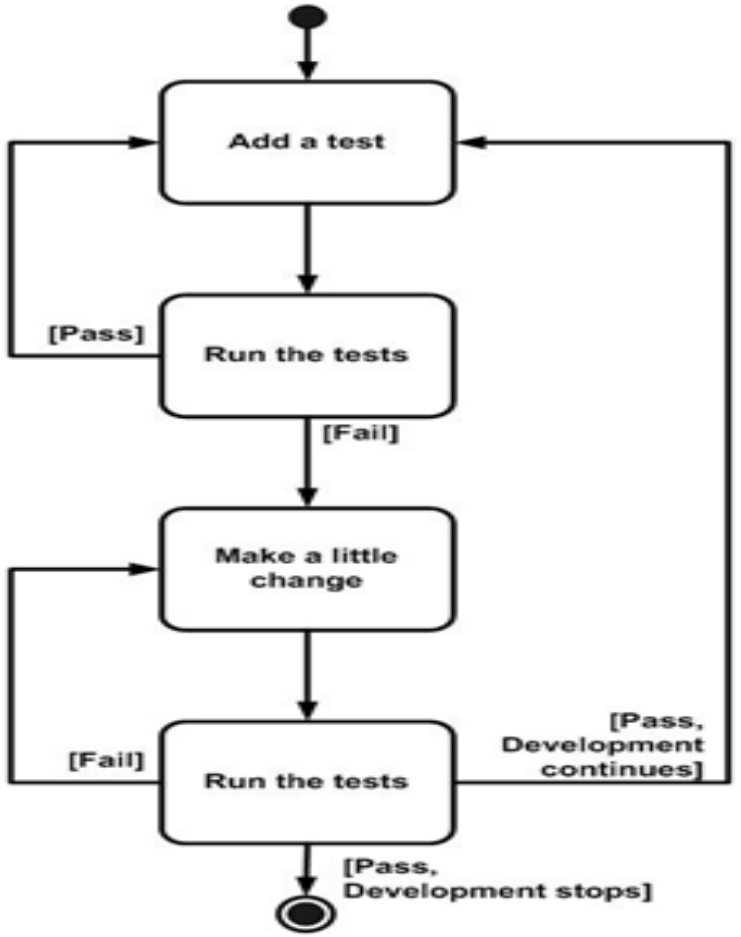
# XP practices

- Pair programming
- Test driven development
- Continuous integration
- Use coding standard
- Defer optimization
- Keep simple design
- Use standup meeting
- Make frequent small releases



# XP practices – cont.

- Pair programming
  - One is the programmer, actively coding; the other is observer, continuously identify mistakes and possible defects = team work
- Test driven development
  - Test driven development is a technique for building software that guides software development by testing the software frequently.
  - The developer writes the unit tests NOT testers



# XP practices – cont.

- Continuous integration (C.I.)
  - Continuous integration is a software development practice where members of a team integrate their work frequently (usually daily). – proposed by Grady Booch in 1991
  - C.I. Benefits:

    (1) Immediate bug detection;

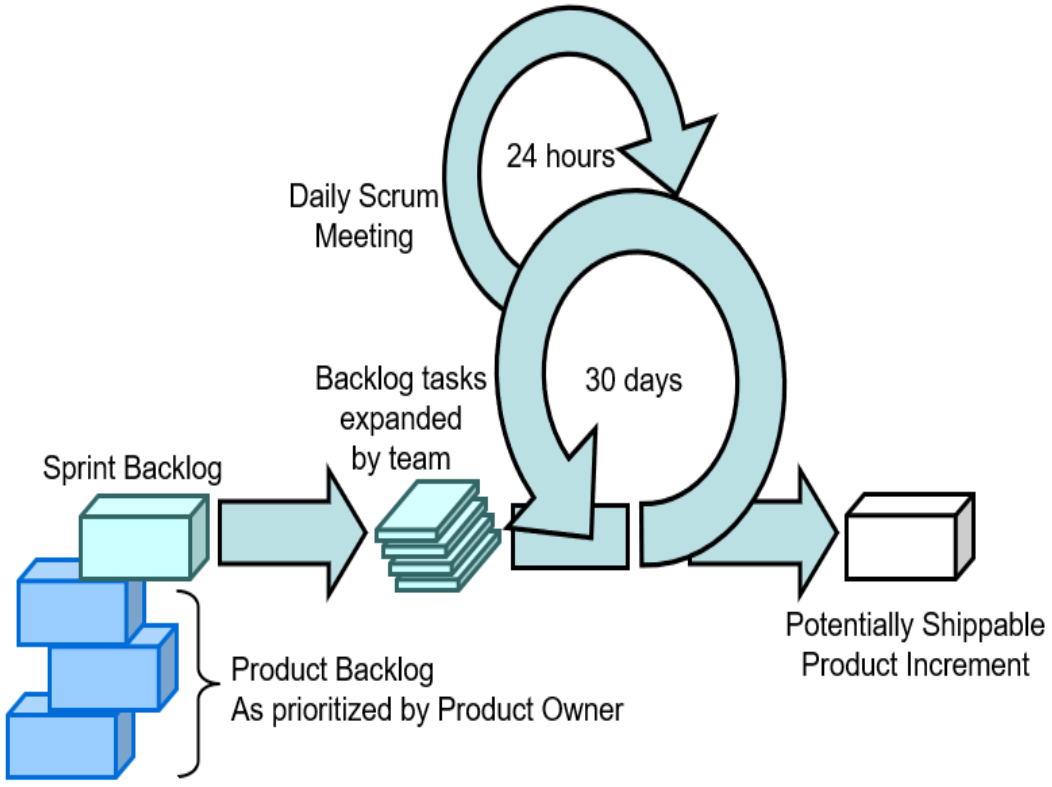
    (2) A deployable system at any given point
- Use coding standard
  - Coding standard helps code review and maintenance
- Defer optimization
  - Make software working correctly first.

# XP practices – cont.

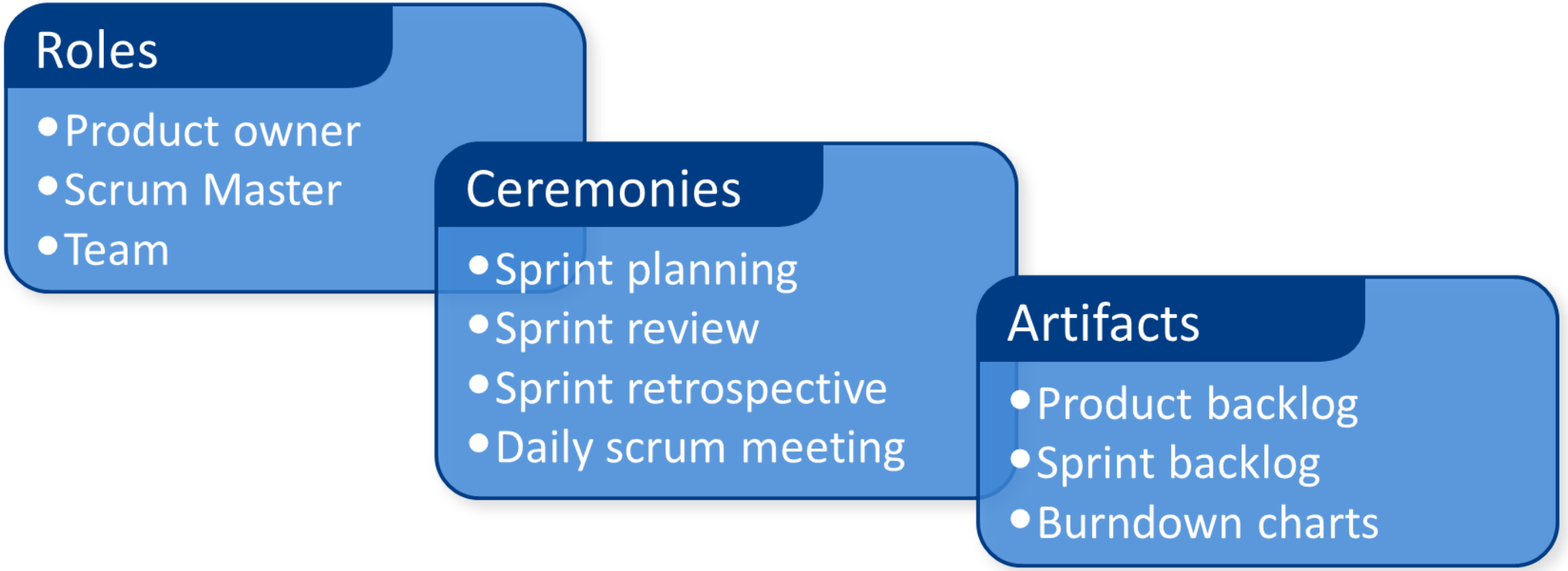
- Keep simple design
  - If there are multiple solutions, select the simplest solution.
- Use standup meeting
  - Usually once a day to communicate in a team environment
- Make frequent small releases
  - Iterative/incremental process

# Scrum

- Definition: Scrum is a process framework used to manage the development of software products and other knowledge work.
- It is a type of Agile software development.
- It uses iterative, incremental practices to increase productivity and reduce project times.
- Proposed by Ken Schwaber and Jeff Sutherland in 1995.



# Scrum framework




**Sprint** is one timeboxed iteration of a continuous development cycle.

# Scrum roles

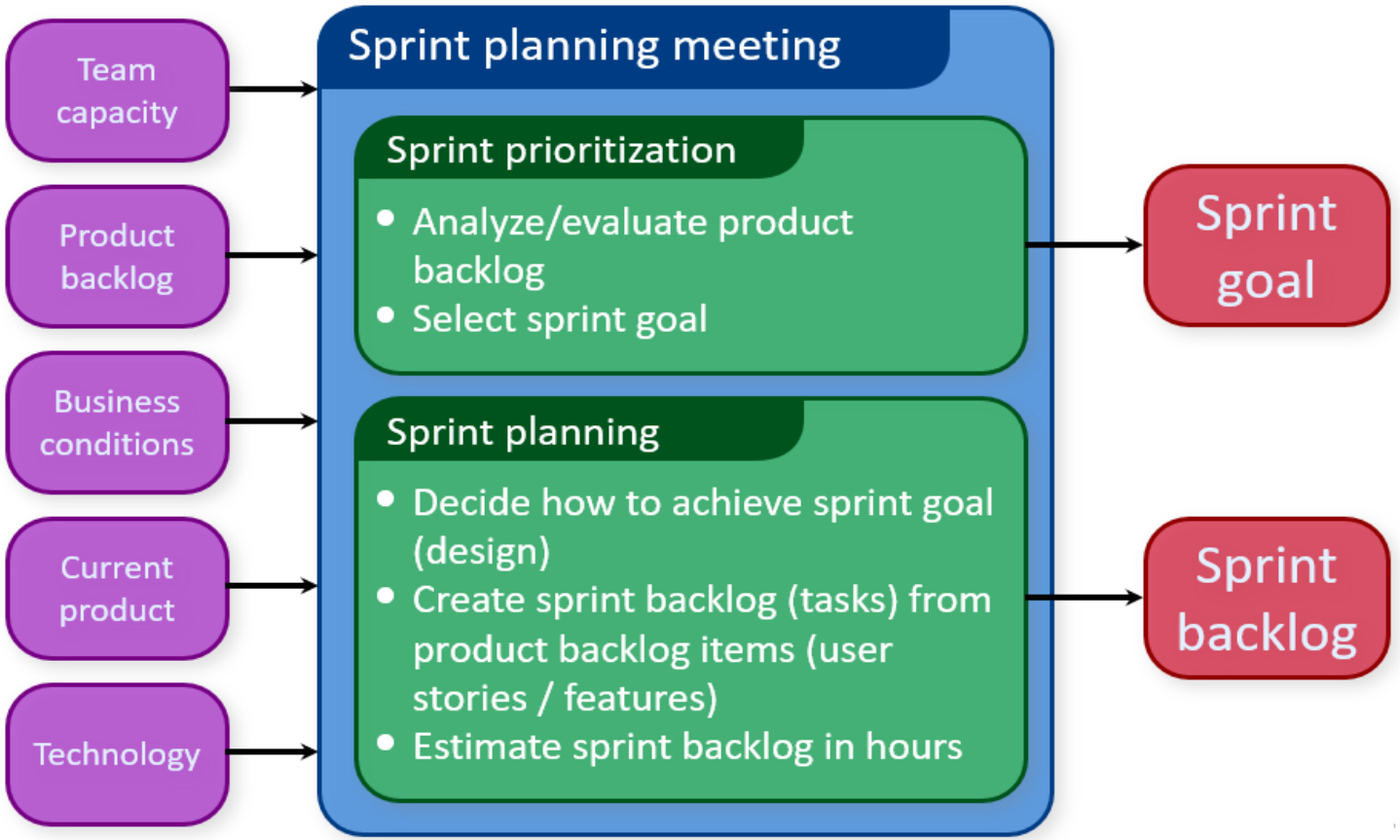
- Product Owner
  - Possibly a Product Manager or Project Sponsor
  - Decides features, release date, prioritization, $$$

- Scrum Master
  - Typically a Project Manager or Team Leader
  - Responsible for enacting Scrum values and practices
  - Remove impediments / politics, keeps everyone productive

- Team members
  - 5-10 members;  Teams are self-organizing
  - Cross-functional: QA, Programmers, UI Designers, etc.
  - Membership should change only between sprints

# Ceremonies

- Sprint planning meeting



# Ceremonies – cont.

- **Sprint review**
- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Product owner and whole team participates
- **Sprint retrospective**
- Review the process and team working
- Discuss what went well and what could be improved
- Whole team participates

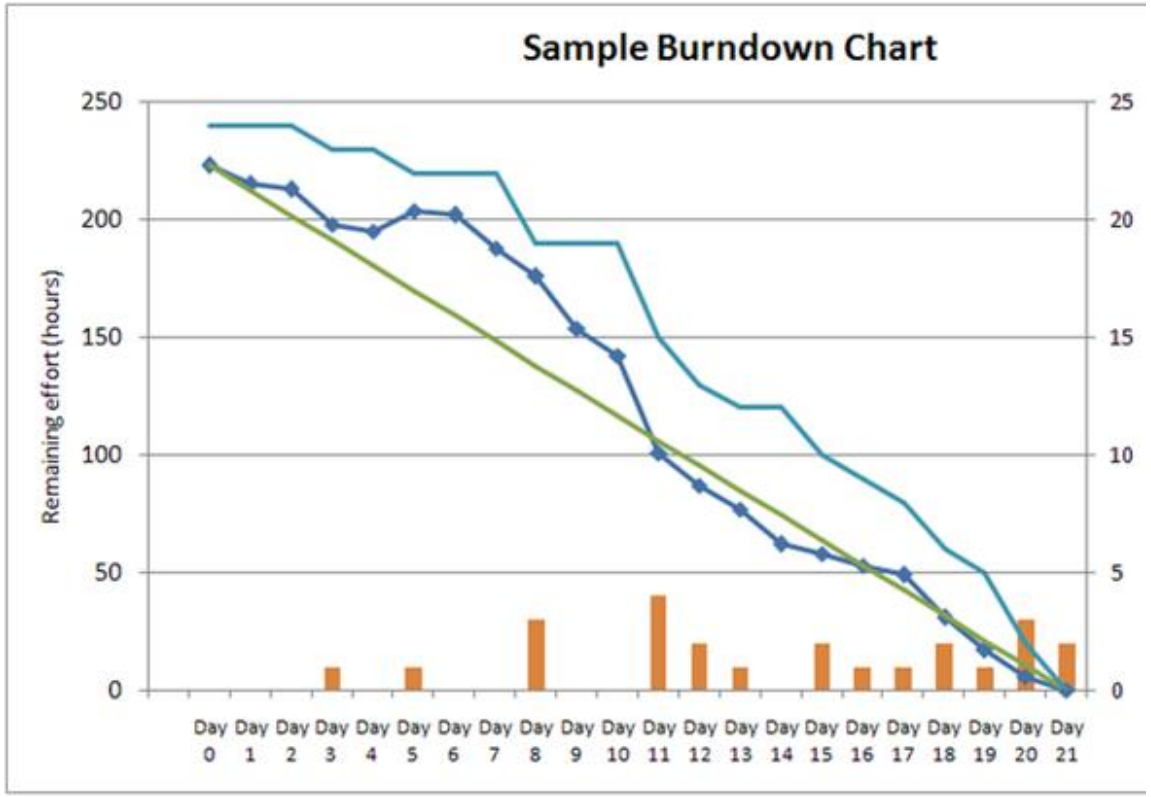
# Ceremonies – cont.

- **Daily Scrum meeting**
  - Stand-up meeting: 15 minutes

- Three questions answered by each team member:
  1. What did you do yesterday?
  2. What will you do today?
  3. What obstacles are in your way?

# Artifacts

- **Sprint burndown chart**
- Burndown chart: it is a graphical representation of work left to do versus time.
- A display of what work has been completed and what is left to complete.
  - one for each developer or work item
  - updated every day
  - (make best guess about hours/points completed each day)



# Scrum vs. XP

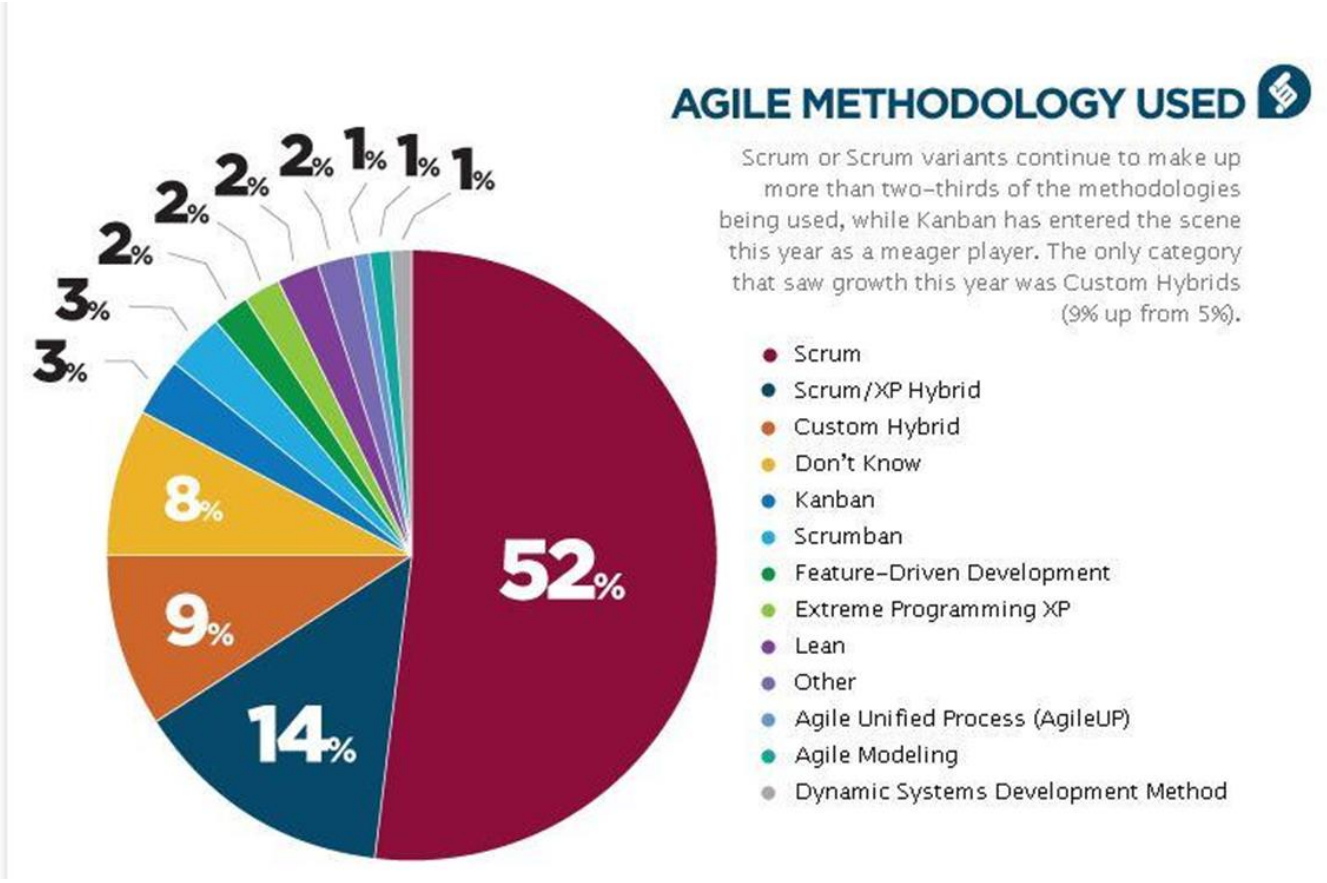
## SCRUM vs. XP

- SCRUM is a little more management and planning focused (Sprint planning, backlog, etc.)
- XP is more “activity” focused (pair programming, test-driven development, etc.)
- Often the practices are mixed

| *Quality Parameter* | *XP* | *Scrum* |
|---|---|---|
| Engineering practices | Yes | No |
| Project management practices | No | Yes |
| Accept changes in iteration at any time | Yes | No |
| Requirement | Yes | No |

# Summary

- Agile development
  - concept
- Extreme programming (XP)
  - Concept
  - XP values
  - XP roles and practice
- Scrum
  - Concept
  - Scrum framework
- Both XP and Scrum are types of agile software development



# Announcement

- **Group project presentation schedule:**
  - Nov. 21 (Thursday) group numbers 9, 13, 18, 2
  - Nov. 26 (Tuesday) group numbers 6, 11, 1, 21, 22, 25
  - Nov. 28 (Thursday) group numbers 10, 8, 5, 20, 7
  - Dec. 3 (Tuesday) group numbers 12, 15, 24, 19, 3
  - Dec. 5 (Thursday) group numbers 16, 23, 14, 17, 4
- Group project
  - Project report due date Nov. 22 at 11:59pm

- **The presentation time is 6 minutes (3 minutes theory and 3 minutes demos).** If your time exceeded more 7 minutes, your grade will be reduced 20%.