System Analysis and Design

Eighth Edition

Alan Dennis, Barbara Wixom, Roberta M. Roth

Chapter 13

Agile Development Methods

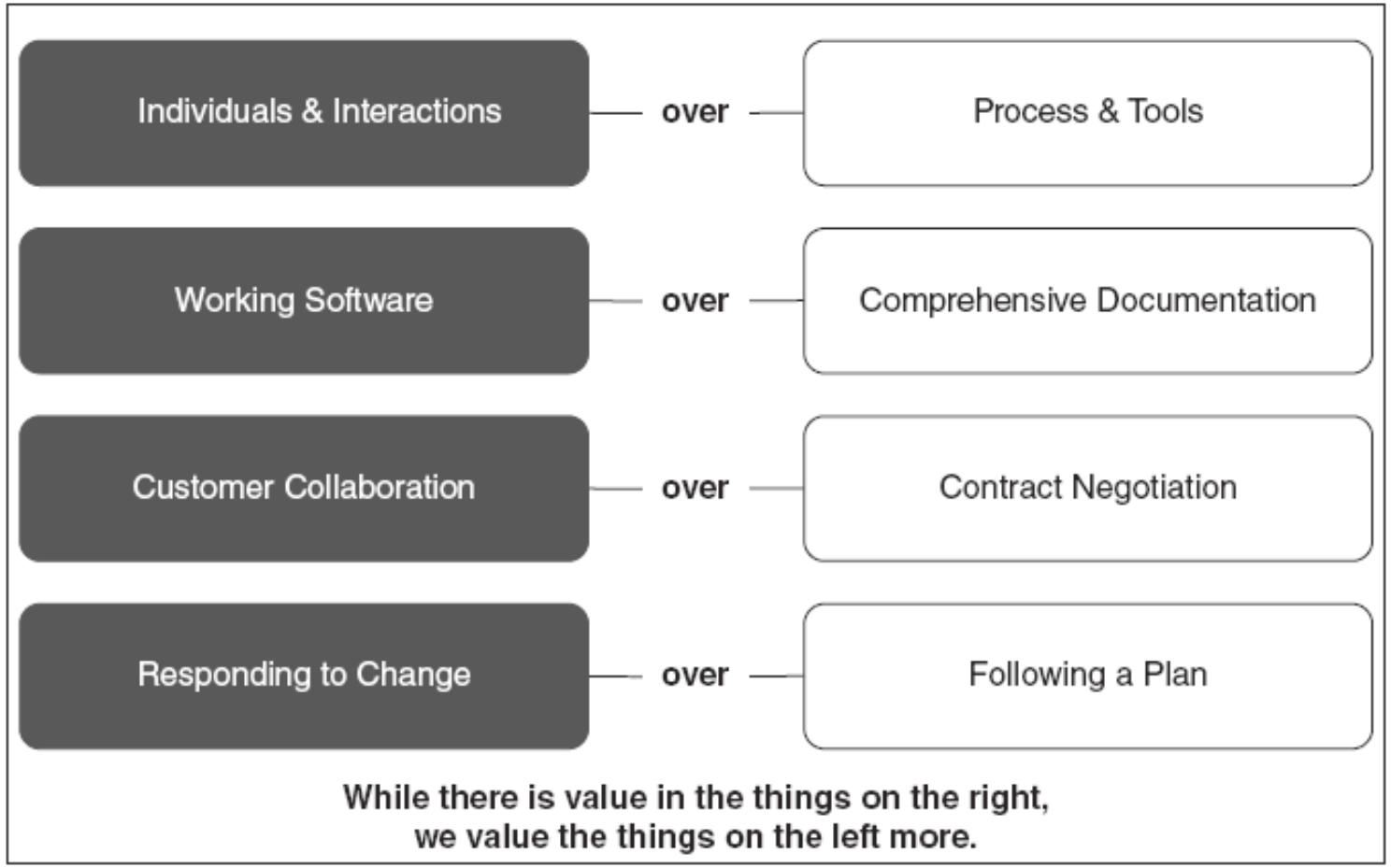
Objectives

- Be able to describe the Agile values and principles expressed in the Agile Manifesto
- Be able to explain the benefits organizations gain by using Agile development approaches
- Be able to describe the overall structure of the Scrum development approach
- Be able to list and explain four key characteristics of Scrum
- Be able to describe the roles of product owner, ScrumMaster, and the team in Scrum
- Be able to discuss the key unique features of Scrum: sprints, user stories, acceptance criteria, story points, and team velocity

Objectives Continued

- Be able to explain the sprint planning process
- Be able to explain the product backlog grooming process
- Be able to discuss the purpose and contribution of Scrum's six distinctive meeting types
- Be able to briefly describe other common Agile approaches
- Be able to describe the factors that limit the adoption of Agile development approaches in organizations today

Agile Manifesto Values



Agile Manifesto 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. Businesspeople 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.

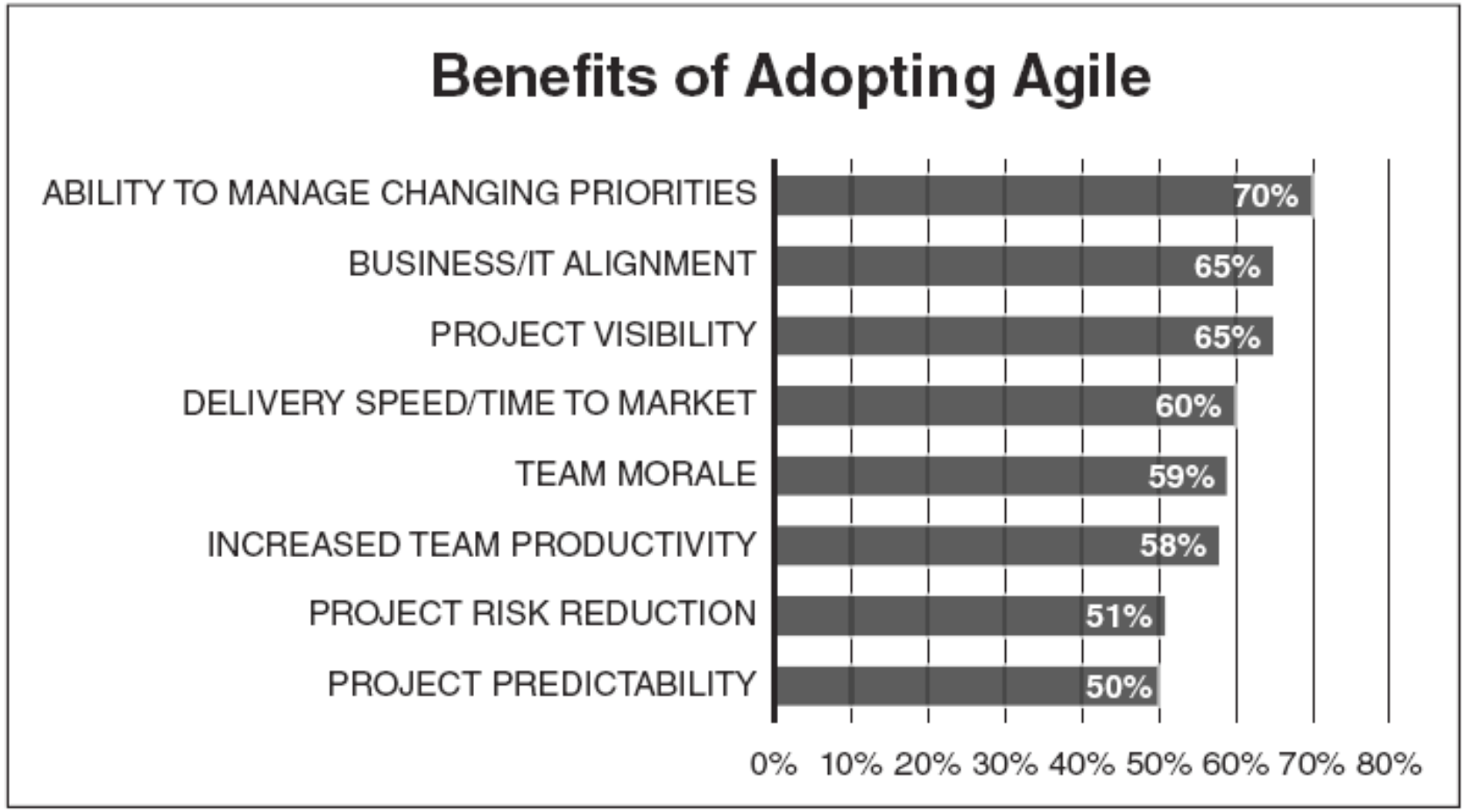
Agile Manifesto Principles Continued

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.

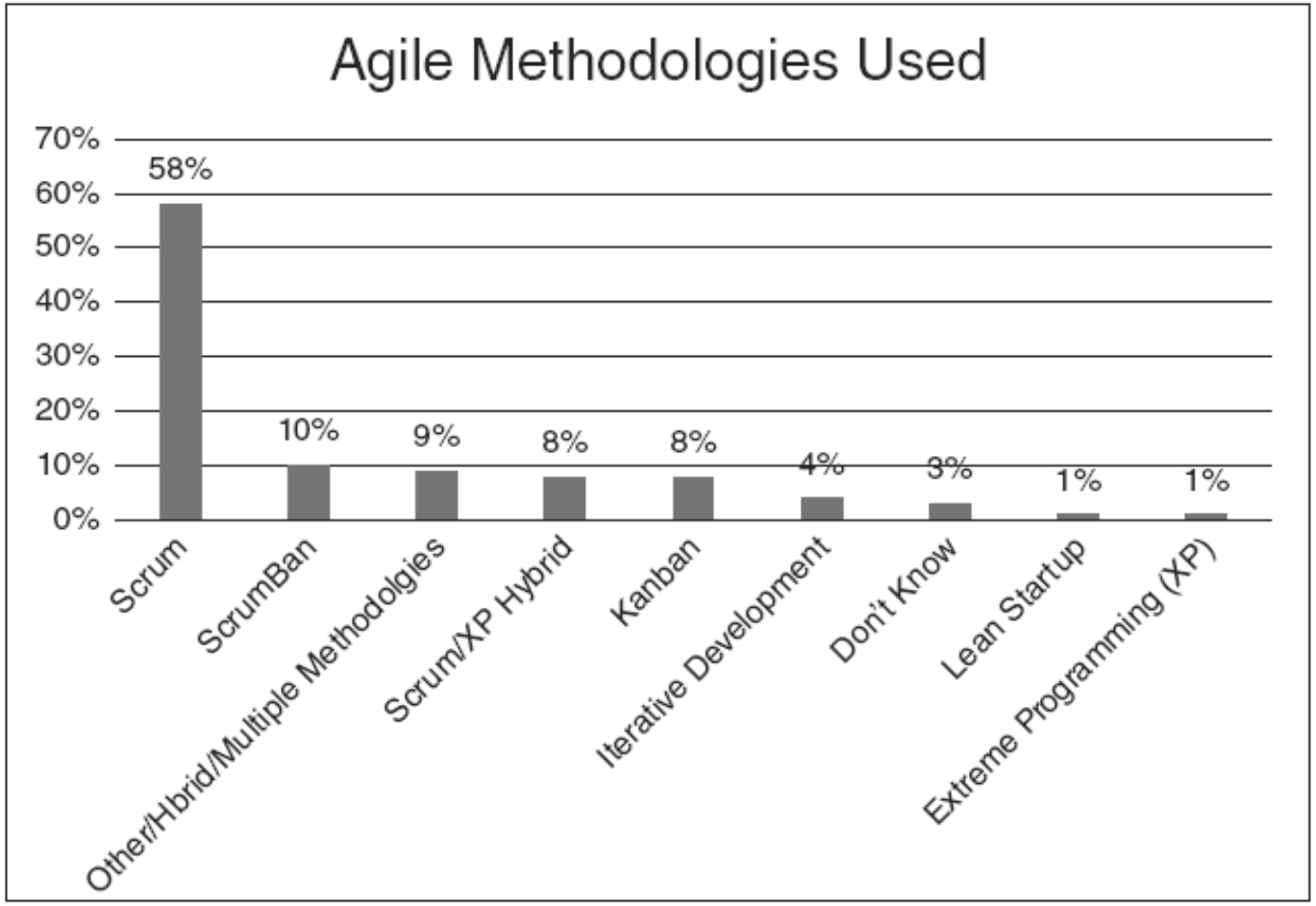
Agile Characteristics

- Close collaboration between the project team and business experts
- Face-to-face communication
- Frequent delivery of new, deployable business value
- Tight, self-organizing teams
- Reduced impact of changes in requirements

Benefits of Adopting Agile



Agile Methods in Use



Scrum

- Scrum is an Agile approach that is designed to enable delivery of working software providing the highest business value in the shortest amount of time
- It is structured so that the development team rapidly and repeatedly produces actual working software that is ready for inspection in two-week to four-week cycles
- Priorities established by the business define the development team's "to-do" list
- The team self-organizes to determine the best way to deliver the features in response to those priorities.
- Every two to four weeks, the team demonstrates real working software

Overview of Scrum

- Ideas for features of the new system are provided by end-users, customers, the development team, and other stakeholders
  - These ideas represent the system's requirements
- These items are gathered and managed by the product owner
  - Represents the organization's interests in this project
- The product owner develops and manages a prioritized feature list, also called a product backlog, that serves as the development teams' to-do list
- The development cycles in the Scrum development process are called sprints
  - One to four weeks

Overview of Scrum Continued

- At the beginning of a sprint, each development team assigned to the project conducts a ***sprint planning meeting***
- The feature(s) are refined into a more detailed set of tasks, called the ***sprint backlog***
- A standard feature of Scrum is the daily scrum meeting, called a ***daily standup***
- At the end of the sprint, potentially shippable software should be produced
- As a final aspect of the sprint, the team performs a ***sprint retrospective*** on its performance in the just-completed sprint

Scrum Characteristics

- Uses of dedicated, self-organizing teams
- The software product development is accomplished in a series of short work cycles, called sprints
- The system’s requirements are captured from end-users, customers, and other interested stakeholders in a list called a product backlog
- No specific software engineering practices are prescribed in the Scrum methodology

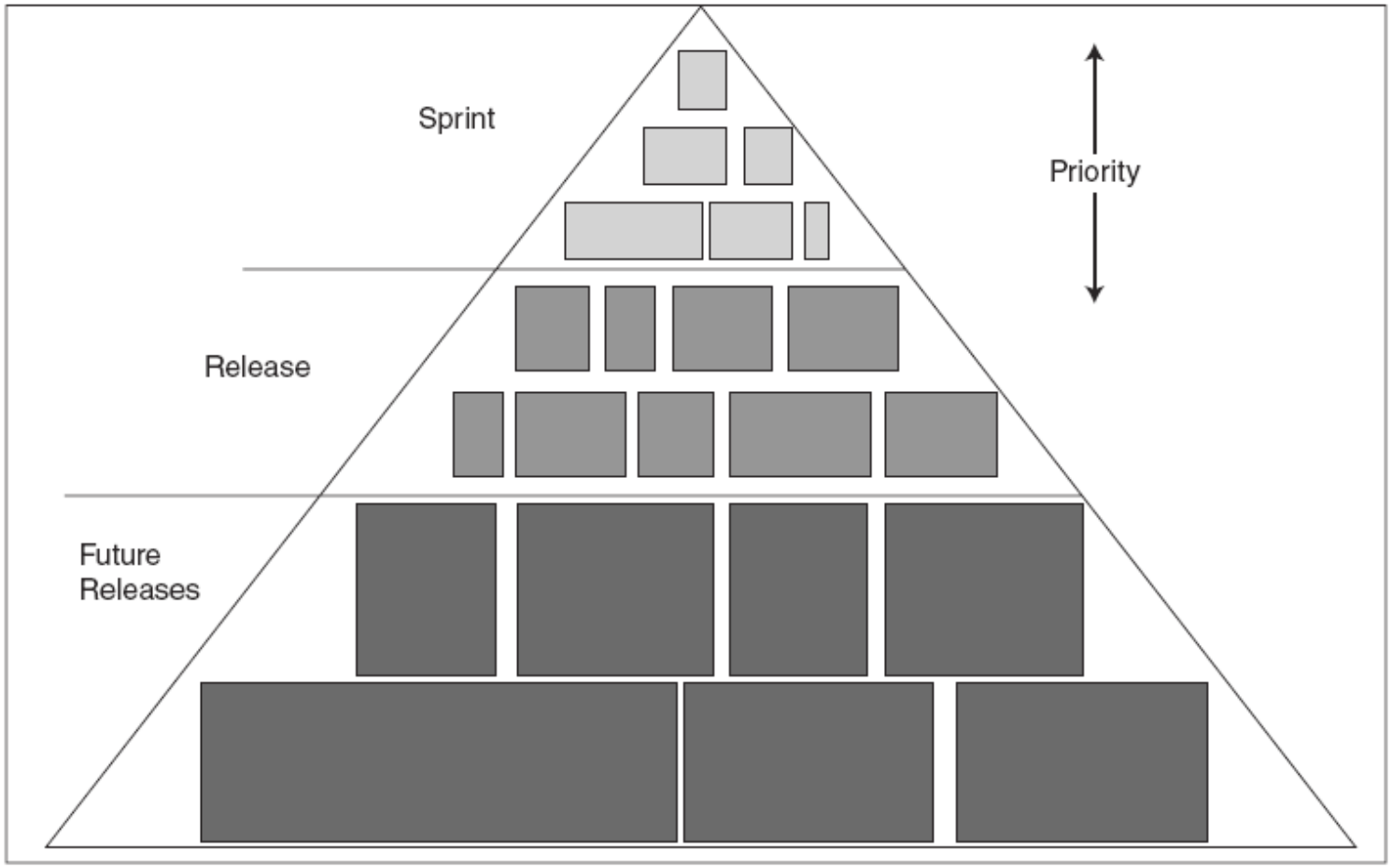
Scrum Roles

- The person designated as ***product owner*** is typically a representative of the business area for which the system is being developed
  - Holder of business value
  - Instrumental in defining the features of the product that will be included in the product backlog
- The ***ScrumMaster*** is seen as a servant leader, providing guidance in the use of Scrum by the team
  - They ensure that the team is fully functional and productive
- The ***development team*** typically consists of 5–9 people
  - The team is free to organize itself as it sees fit and to take on and deliver chunks of work in frequent increments

Scrum Features

- In Scrum, projects make progress through a series of work cycles called ***sprints***
- In Scrum, requirements are expressed through ***user stories***
  - A large story, termed an ***epic***, is one that may take many weeks or more to implement
  - An ***implementation size story*** will take days or less to implement
- User story refinement is expected as larger stories are broken into smaller sized stories that add more details

User Stories in the Product Backlog



Acceptance Criteria

- Even implementation size user stories may be lacking in detail
- Scrum includes the acceptance criteria feature to provide more detailed requirements
- Acceptance criteria, or “conditions of satisfaction” help the team understand the story and set expectations as to when the team can consider something “done”
- Good acceptance criteria help the team clarify what should be built before the work starts
- Anyone on the Scrum team can write acceptance criteria using input from the team

User Story and Acceptance Criteria Example

User Story:

As a mutual fund investor, I want a strong password, so that my investment account information is secure.

Acceptance Criteria:

- The password must be at least 10 characters
- The password must contain at least 1 character from each of the following groups:
  - Lower case alphabet
  - Upper case alphabet
  - Numeric characters
  - Special characters (!, @, #, $, %, ^, &, *)

Story Points

- A feature of Scrum is the ***story point***, which is used to measure the size of a story
- A story point does not have a precise meaning
- The usual practice is to use a range of story points based on a modified Fibonacci sequence
  - 1, 2, 3, 5, 8, 13, 20, 40, 100
- User stories with large story point values are considered ***epics***
  - Will need to be broken down into smaller, more detailed user stories over time
- The number of story points that a team can successfully complete during a sprint is termed the ***team velocity***

Definition of Done

- Feature is complete
- Code is complete
- Fully tested
- No known defects—fully documented

Sprint Planning Process

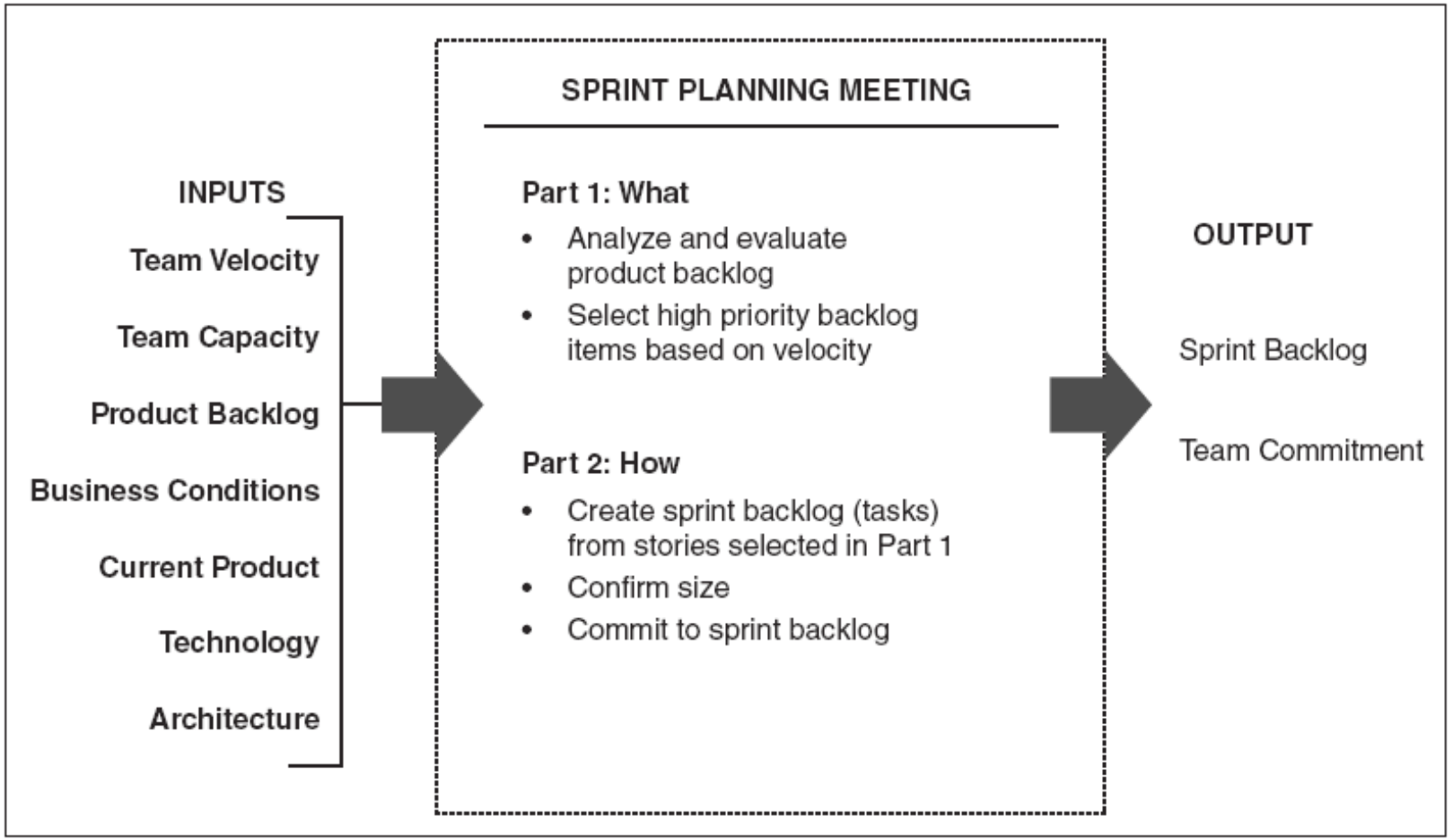


Illustration of Sprint Planning

Velocity—20 points

Product Backlog

Cancels	5pts
Returns	8pts
Gift Wrap	3pts
Coupons	2pts
Wish List	13pts
Product Reviews	8pts
Tracking	20pts
Recommendations	5pts

Sprint Commitment

Cancels	5pts
Returns	8pts
Gift Wrap	3pts
Coupons	2pts

Sprint Backlog

Cancels	• Code the User Interface • Create cancel transaction • Add confirmation e-mail • Update Inventory
Returns	• Code the User Interface • Create Return Transaction • Add Confirmation E-mail • Update Inventory
Gift Wrap	• Allow Gift Wrap Selection • Add to Invoice • Notify Gift Wrap Group
Coupons	• Allow Coupon Entry • Update Invoice

Product Backlog Grooming

- Another important and unique process of Scrum is ***product backlog grooming***
- During this process, the participants review the product backlog with the intent to refine and improve it
- The epics in the backlog may be broken down into smaller, more focused user stories
- Another aspect of this process is consideration of ***technical debt***
  - This represents a change or improvement to the technical environment of the team that should be done but has been delayed or deferred

Scrum Meetings

Session	Purpose	Timing/Duration	Participants
Release Planning	Determine what a release should include and when it should be delivered	Start of release 2–4 hours	Product owner, ScrumMaster, key stakeholders, architect, team (optional)
Sprint Planning	Elaborate, estimate, and prioritize highest-value product backlog items for a sprint	Start of each sprint 2–4 hours	Team, ScrumMaster, product owner
Daily Standup	Facilitate rapid coordination between team member and product owner	Daily 15 minutes	Team, ScrumMaster, product owner
Sprint Review (or Demo)	Demonstrate completed functionality to interested stakeholders and users to show progress and get feedback	End of each sprint 1–1½ hours	Team, ScrumMaster, product owner, interested stakeholders and users
Sprint Retrospective	Reflect on project and process issues within team and act as appropriate	End of each sprint 30–45 minutes	Team, ScrumMaster, product owner
Product Back-log Grooming	Review upcoming user stories to confirm size and clarify team questions and decompose to execution level	Each sprint 1–2 hours	Team, ScrumMaster, product owner

How Does Scrum End?

- One of the challenges of a development approach such as Scrum is determining when the project is finished
- This issue is not as straightforward as it sounds
- The product owner is responsible for managing the product backlog and judging the value of the features in the backlog
- From a practical standpoint, the project sometimes must be terminated because the project budget is exhausted
- In other cases, the organization determines that the team needs to be reassigned to a different, higher value project and the project is ended

Other Types of Agile Methodologies

- Crystal Development Methodology
- Dynamic Systems Development Methodology
- Feature Driven Development
- Lean Software Development

Crystal Development Methodology

- The Crystal Development Methodology4 is a lightweight and flexible approach to develop software
- Crystal incorporates several essential properties:
  - Teamwork is essential to Crystal and team members are encouraged to work on tasks as a team rather than individually
  - Communication is considered the most critical aspect of the project. Communication spans both developer–customer interactions and interactions between team members
  - Simplicity is stressed in terms of product design, requirements, and other project elements
  - Reflection is incorporated so that team members respond, and report as needed; valid reasoning is provided for every action; and work can be revised and reconstructed when necessary
  - Frequent adjustments are expected
  - Process improvements are performed continuously

Dynamic Systems Development Methodology

- Dynamic Systems Development Methodology (DSDM) is an iterative, incremental approach
- It is based on a four-phrase framework:
  1. Feasibility and business study
  2. Functional model/prototype iteration
  3. Design and build iteration
  4. Implementation
- DSDM relies upon direct and frequent collaboration between the developers and users

Feature Driven Development

- Feature Driven Development (FDD) is an Agile framework that organizes software development around completing features
- The FDD framework involves:
  1. Develop an overall model
  2. Build a features list
  3. Plan by feature
  4. Design by feature
  5. Build by feature

Lean Software Development

1. Eliminate everything that is not necessary for completing the project
2. Build quality into the product from the outset
3. Improve team knowledge about the project
4. Commit to rapid development
5. Plan for fast product delivery
6. Treat all team members and stakeholders with respect
7. Optimize the value of the project as a whole

Factors Affecting the Slow Adoption of Agile

- Not enough leadership participation
- Inconsistent processes and practices across teams
- Organizational culture at odds with Agile values
- Inadequate management support and sponsorship
- Lack of skills/experience with Agile methods

Chapter Review

- Describe the Agile values and principles expressed in the Agile Manifesto.
- Explain the benefits organizations gain by using Agile development approaches.
- Describe the overall structure of the Scrum development approach.
- List and explain four key characteristics of Scrum.
- Describe the roles of product owner, ScrumMaster, and the team in Scrum.

Chapter Review Continued

- Discuss the key unique features of Scrum: sprints, user stories, acceptance criteria, story points, and team velocity.
- Explain the sprint planning process.
- Explain the product backlog grooming process.
- Discuss the purpose and contribution of Scrum's six distinctive meeting types.
- Briefly describe other common Agile approaches.
- Describe the factors that limit the adoption of Agile development approaches in organizations today.

Key Terms

- Acceptance criteria
- Agile Manifesto
- Crystal Development Methodology
- Daily standup
- Definition of “done”
- Dynamic Systems
- Development Methodology (DSDM)
- Epic
- Feature Driven Development (FDD)
- Implementation size story
- Lean Software Development (LSD)
- Mob programming
- Paired programming
- Product backlog
- Product backlog grooming
- Product owner
- Release planning meeting
- Scrum
- ScrumMaster
- Self-organizing teams
- Sprint
- Sprint backlog
- Sprint Commitment
- Sprint planning meeting
- Sprint retrospective
- Sprint Review (or Demo) meeting
- Story point
- Team capacity
- Team velocity
- Technical debt
- User stories