

Unit 2.
Software Process Models

Fall 2019

Soo Dong Kim, Ph.D.

Professor, School of Software

Soongsil University

Office 02-820-0909 Mobile 010-7392-2220

sdkim777@gmail.com <http://soft.ssu.ac.kr>

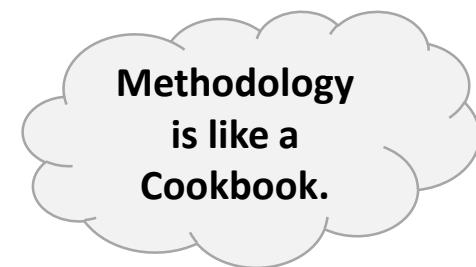
Process in Software Engineering

● **Definition**

- A methodology consists of a process, a vocabulary, and guidelines.
- A process defines a set of activities that together accomplish all the goals of the methodology.

● **Key Elements of a Process**

- Life-Cycle Model
 - Phases, Activities, Steps
- For each Activity
 - Input, Output Artifacts, Instruction, Quality Guidelines



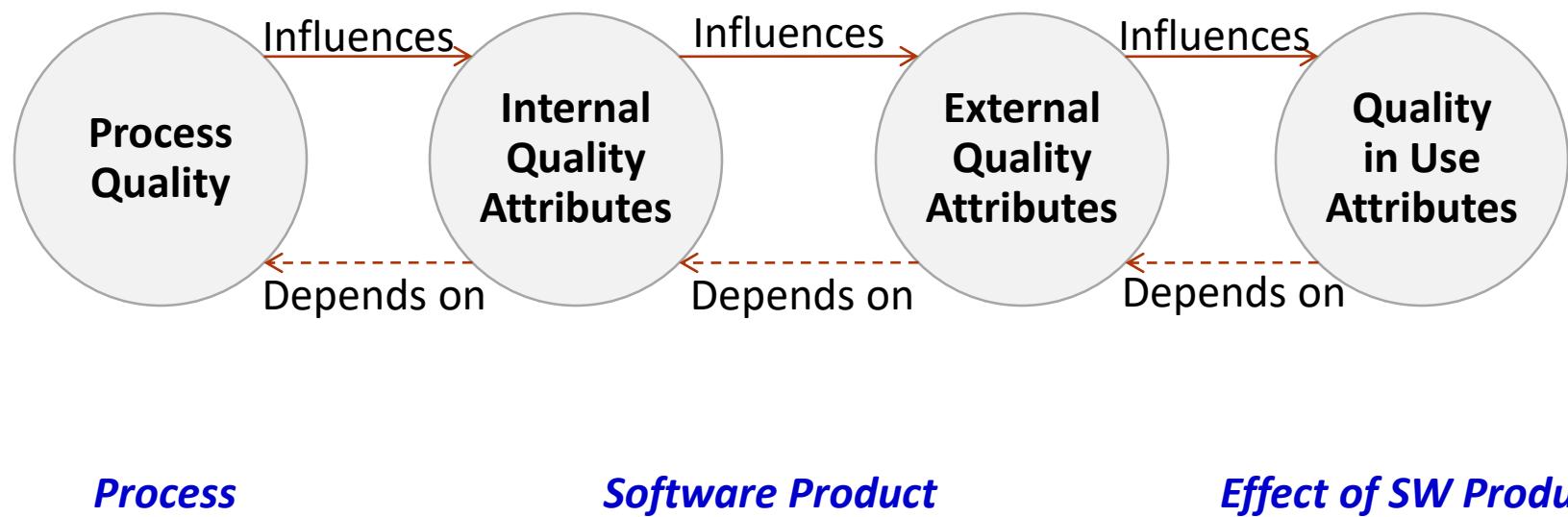
Why is 'process' essential in SE?

- ISO/IEC 9126
 - Revision in 2001
 - ISO/IEC 9126-1 to 9126-4
- Part 1
 - Quality Model
- Part 2
 - External Metrics
- Part 3
 - Internal Metrics
- Part 4
 - Quality In Use Metrics



Quality Model Framework

- **Quality in the Lifecycle**



Quality Model Framework

● **Process Quality**

- Quality of Life-cycle Process
- Process quality contributes to improving product quality.

● **Product Quality**

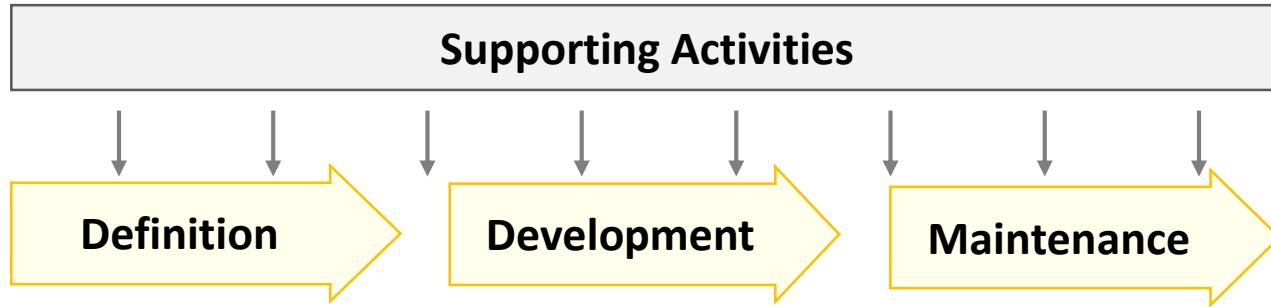
- Can be evaluated by measuring internal attributes or measuring external attributes.
- Internal quality
 - is evaluated by the static measure of intermediate products.
 - View at Technical Level
- External quality
 - is evaluated by measuring the behavior of the code when executed.
 - View of User/Management
- Product quality contributes to improving *quality in use*.

Quality Model Framework

● **Quality In Use**

- User's view of the quality of an environment containing software, and is measured from the results of using the software in the environment.
 - Rather than properties of the software itself.
- User's environment may be different from development environment.

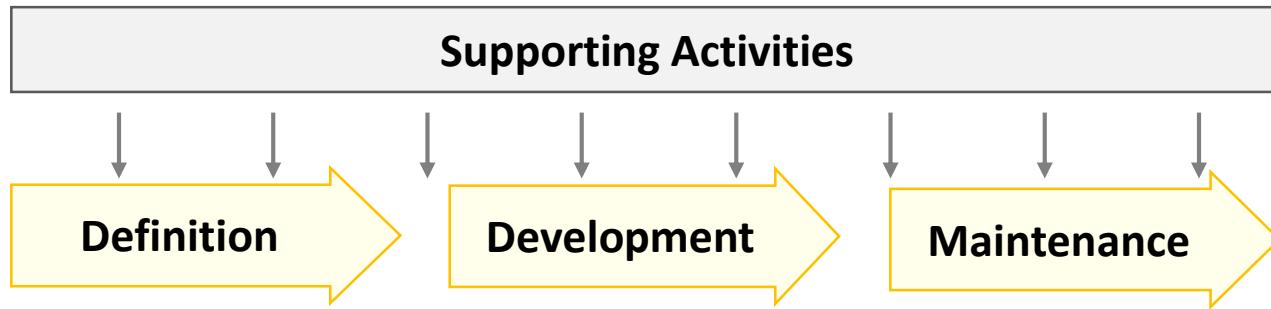
Framework for Processes (1)



● **Supporting Activities**

- Project Management
- Reviews (FTR) and Inspection
- Quality Assurance
- Configuration Management
- Measurement
- Risk Management
- Others

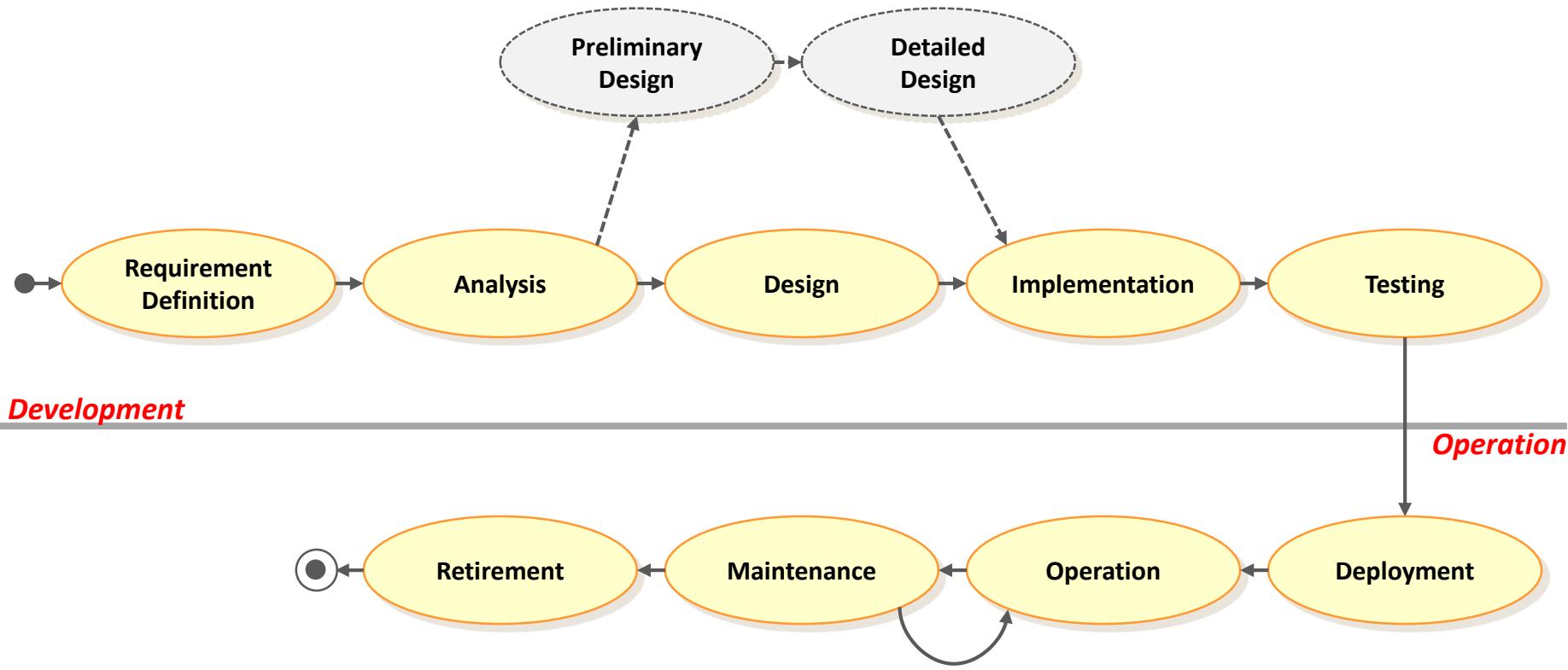
Framework for Processes (2)



● **Engineering Activities**

- **Definition**
 - Requirement Engineering
- **Development**
 - Analysis, Design, Implementation, Testing, Deployment
- **Maintenance**
 - Post-Development Activities
 - Retirement

Engineering Activities



Representative Process Models

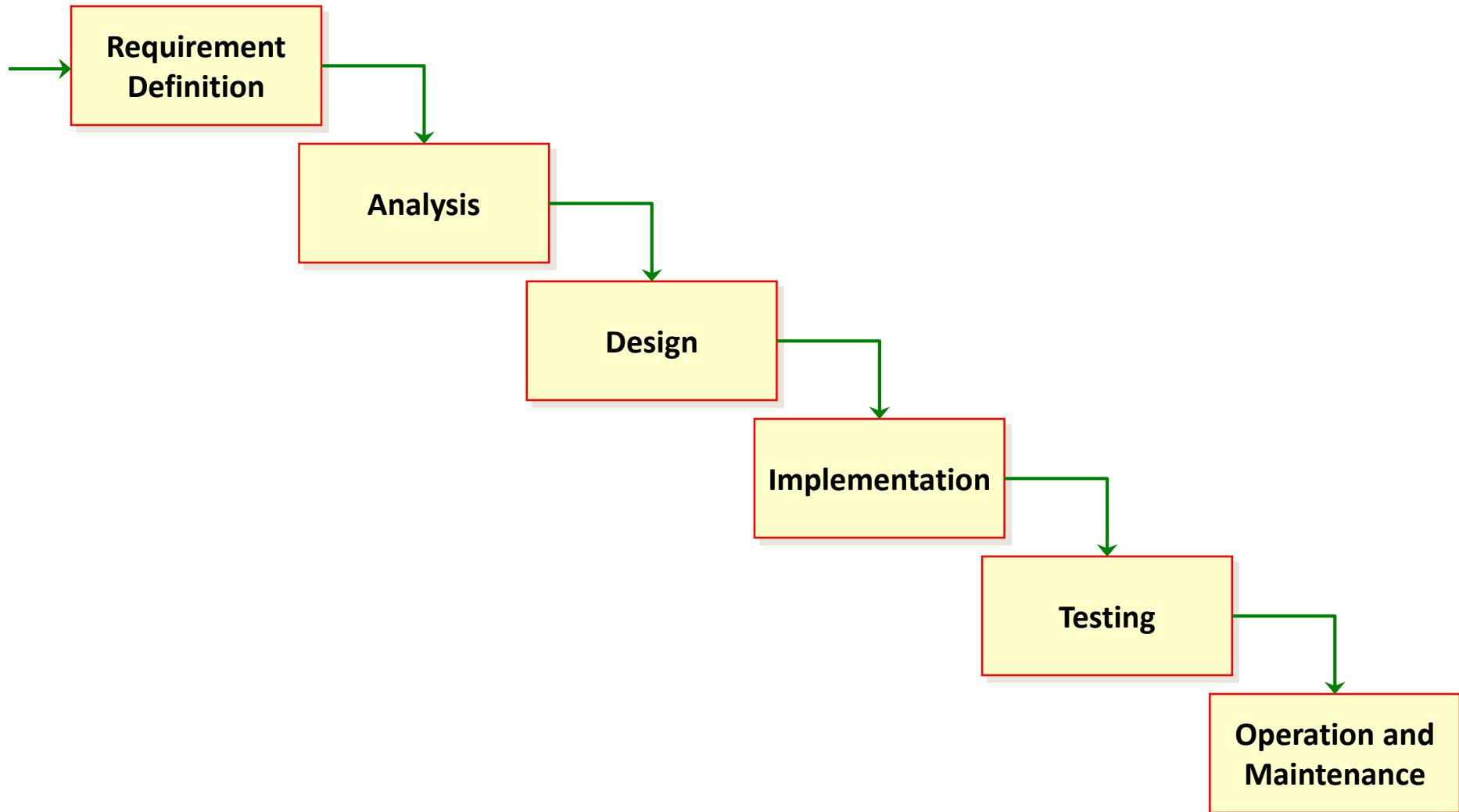
- **Waterfall Model**
- **Rapid Application Development (RAD) Model**
- **Spiral Model**
- **Prototyping Model**
- **Component-Based Development (CBD) Model**
- **Aspect-Oriented Development Model**
- **Agile Process**
 - eXtreme Programming (XP)
 - Scrum
- **Formal Method**

Waterfall Model

Waterfall Model (1)

- **Oldest paradigm in Software Engineering**
- **Sequential approach to S/W development**
- **When to Choose?**
 - When requirements of a problem are reasonably well understood.
 - When work flows from communication through deployment in a reasonably linear fashion.
 - When requirements are well-defined and reasonably stable.

Waterfall Model (2)



Waterfall Model (3)

● **Problems**

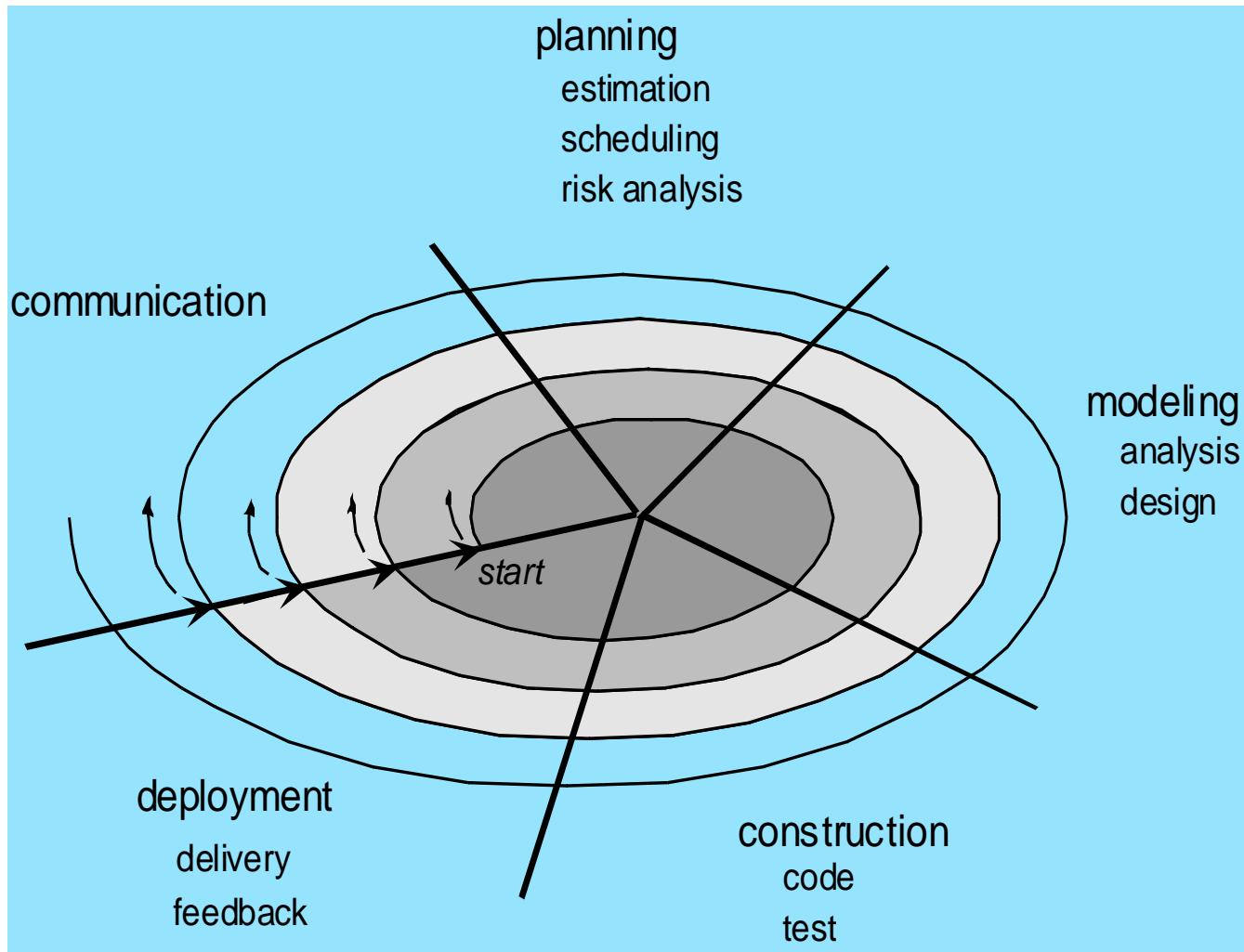
- Real projects rarely follow the sequential flow that the model proposes.
- It is often difficult to for the customer to state all requirements explicitly.
- The customer must have patience.
 - A working version of the program will not be available until late in the project time-span.

Spiral Model

Spiral Model (1)

- **Adapted to apply throughout the entire life cycle of an application, from concept development and maintenance.**
- **Couples the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model.**
- **Software is developed in a series of incremental releases.**
 - **Early iterations**
 - Release might be a paper model prototype.
 - **Later iterations**
 - More complete versions of the engineered system are produced.

Spiral Model (2)



Spiral Model (3)

● **Advantages**

- Effectively Gathering and Refining Requirements
- Risk Reduction through Clients' Feedback
- Realistic Approach for Large-scaled systems

● **Drawbacks**

- Hard to Control
 - Iterations and Increments
 - Progress Monitoring

Agile Process

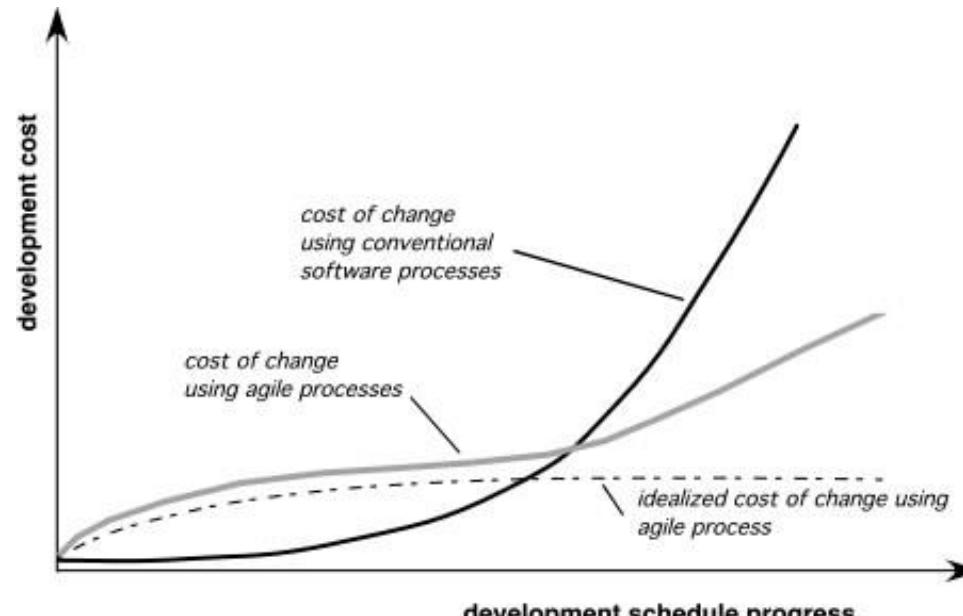
Agile Process (1)

● What is “Agility”?

- Effective (rapid and adaptive) response to change
- Effective communication among all stakeholders
- Drawing the customer onto the team
- Organizing a team so that it is in control of the work performed

Yielding rapid, incremental delivery of software

● Agility and Cost of Change



Agile Process (2)

- A group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams [Wiki]
- To promote;
 - Adaptive Planning
 - Evolutionary development and delivery
 - A time-boxed iterative approach,
 - Rapid and flexible response to change
- Introduced by Agile Manifesto in 2001

12 Agile Principles (1)

- **Principle 1. Customer satisfaction by rapid delivery of useful software**
 - The highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- **Principle 2. Welcome changing requirements, even late in development**
 - Agile processes harness change for the customer's competitive advantage.
 - Agile teams work to keep the software structure flexible, so requirement change impact is minimal.
- **Principle 3. Working software is delivered frequently (weeks rather than months)**
 - From a couple of weeks to a couple of months with a preference to the shorter time scale

12 Agile Principles (2)

- **Principle 4. Working software is the principal measure of progress**
 - Agile projects measure their progress by measuring the amount of working software.
- **Principle 5. Sustainable development, able to maintain a constant pace**
 - Agile Processes promote sustainable development.
 - The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- **Principle 6. Close, daily cooperation between business people and developers**
 - For agile projects, there must be significant and frequent interaction between the
 - customers,
 - developers, and
 - stakeholders.

12 Agile Principles (3)

- **Principle 7. Face-to-face conversation is the best form of communication (co-location)**
 - In agile projects, developers talk to each other.
 - The primary mode of communication is conversation.
 - Documents may be created, but there is no attempt to capture all project information in writing.
- **Principle 8. Projects are built around motivated individuals, who should be trusted**
 - Give them the environment and support they need, and trust them to get the job done.
- **Principle 9. Continuous attention to technical excellence and good design**
 - The way to go fast is to keep the software as clean and robust as possible.
 - Thus, all agile team-members are committed to producing only the highest quality code they can.

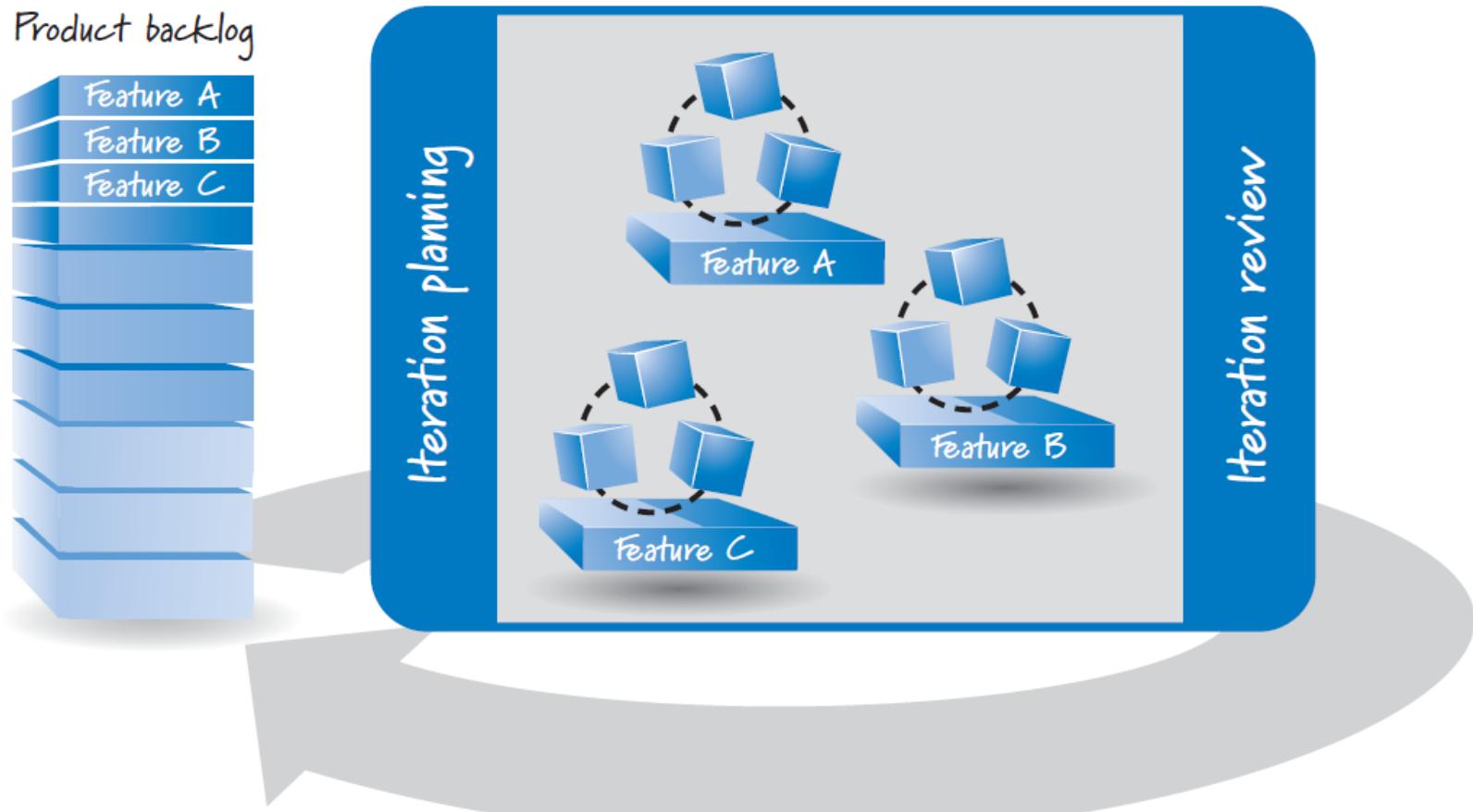
12 Agile Principles (4)

- **Principle 10. Simplicity is essential**
 - Agile teams take the simplest path that is consistent with their goals.
- **Principle 11. Self-organizing teams**
 - The best architectures, requirements, and designs emerge from Self-Organizing Teams.
 - Responsibilities are not handed to individual team members from the outside.
 - Responsibilities are communicated to the team as a whole, and the team determines the best way to fulfill them.
- **Principle 12. Regular adaptation to changing circumstances**
 - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
 - An agile team knows that its environment is continuously changing, and knows that they must change with that environment to remain agile.

Agile Process: Scrum

Overview on Scrum (1)

- Agile approach for developing innovative products and services



Overview on Scrum (2)

- **To begin by creating a product backlog**
 - A prioritized list of the product backlog items (PBIs).
- **To work on the most important or highest priority items first, guided by the product backlog**
- **To perform each work in short, time-boxed iterations**
 - Ranging from a week to a calendar month in length
 - By a self-organizing, cross-functional team
 - Covering the much greater amount of work than can be completed by a team in one short-duration iteration

Overview on Scrum (3)

- **At the start of each iteration**

- To plan which high-priority subset of the product backlog to create in the upcoming iteration

- **At the end of the iteration**

- To produce a potentially shippable product or increment of the product
 - To review the completed features with the stakeholders to get their feedback
 - Based on the feedback, to alter both what they plan to work on next and how the team plans to do the work

Scrum Roles

- **Scrum Team**

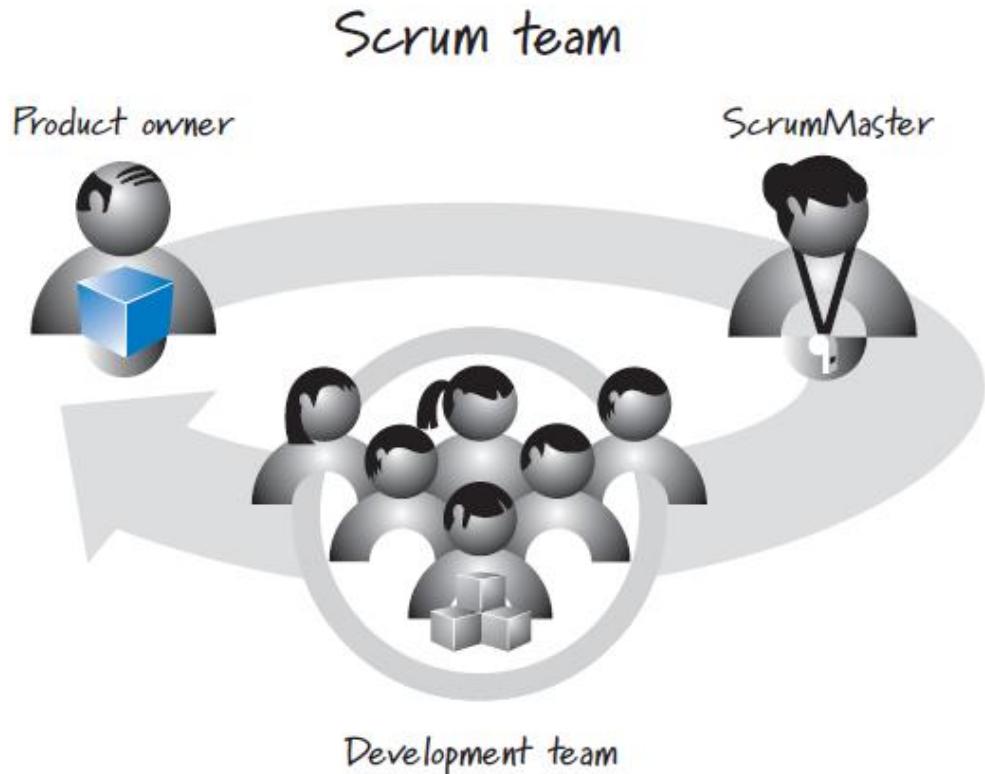
- Consists of one or more members.

- **In each Scrum Team**

- To make up of essential three scrum roles;

- Product Owner
 - ScrumMaster
 - Development Team

- Can include other roles



Scrum Roles – Product Owner

- **The single authority responsible for deciding what features and functionalities will be developed and in what order**
- **To maintain and communicate to all other participants a clear vision of what the Scrum team is trying to achieve**
 - To actively collaborate with the ScrumMaster and development team
 - Must be available to answer questions soon after other participants are posed
- **Responsible for the overall success of the solution being developed or maintained**

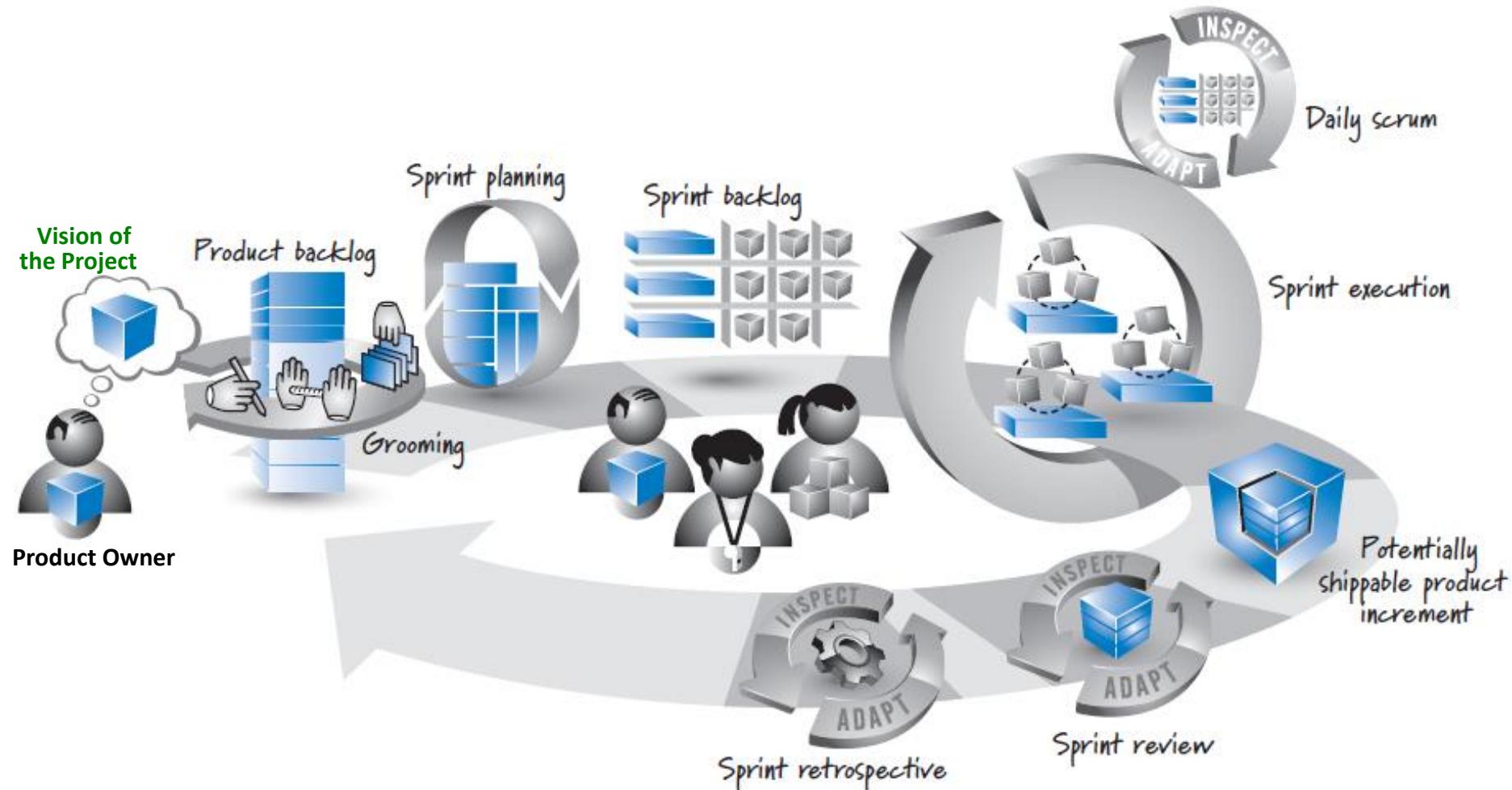
Scrum Roles – ScrumMaster

- **Responsible for guiding the team in creating and following its own process based on the broader Scrum framework**
 - To help everyone involved understand and embrace the Scrum values, principles, and practices
- **To act as a coach**
 - To help the Scrum team and the rest of the organization develop their own high performance, organization-specific Scrum approach
- **To act as a facilitator**
 - To help the team resolve issues and make improvements to its use of Scrum
 - To protect the team from outside interference
- **To act as a leader**
 - To take a leadership role in removing impediments that inhibit team productivity

Scrum Roles – Development Team

- **Responsible for determining how to deliver what the product owner has asked for**
- **To self-organize to determine the best way to accomplish the goal set out by the product owner**
- **Typically five to nine people in size**
 - Its members must collectively have all of the skills needed to produce good quality, working software.

Scrum Activities and Artifacts (1)



Scrum Activities and Artifacts (2)

- **To perform grooming**

- Done by a product owner
- To break a project vision of the product owner down into a set of features that are collected into a prioritized list (i.e. product backlog)
- Artifact
 - A product backlog

- **To perform sprint planning**

- Mainly done by a development team
- To determine a subset of the product backlog items it believes it can complete
- Artifact
 - A sprint backlog
 - Which describes how the team plans to design, build, integrate, and test the selected subset of features from the product backlog during that particular sprint

Scrum Activities and Artifacts (3)

- **To perform the sprint execution**

- Done by the development team
- To perform the tasks necessary to realize the selected features
- To perform the daily scrum everyday
 - The team members help manage the flow of work by conducting a synchronization, inspection, and adaptive planning activity.
- Artifact
 - A potentially shippable product increment
 - Which represents some, but not all, of the product owner's vision, at the end of sprint execution

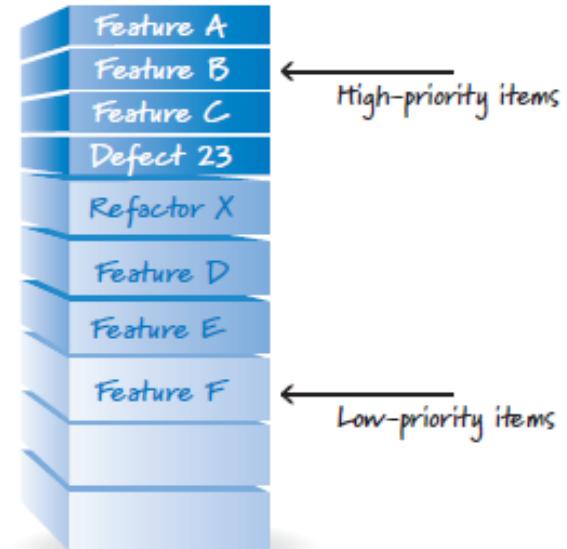
Scrum Activities and Artifacts (4)

- **To perform two inspect-and-adapt activities**
 - **The sprint review**
 - The stakeholders and Scrum team inspect the product being built.
 - **The sprint retrospective**
 - The Scrum team inspects the Scrum process being used to create the product.
 - **Artifact**
 - Adaptations that will make their way into the product backlog or be included as part of the team's development process

Product Backlog

Modified

- A prioritized list of the project work, i.e. Product Backlog Items.
- PBIs are prioritized.
 - Higher-value items appear at the top of the product backlog.
 - Lower-value items appear toward the bottom.
- Initially generated by the product owner
 - With input from the team and stakeholders
- Can be evolved, i.e. Grooming
 - Items can be added, deleted, and revised at the end of each iteration.
 - Mainly due to (1) change on requirement change and (2) team's growing understanding on the product.



Types of Product Backlog Items

● Feature Type

- Represents a functional unit to be developed.
- Example) Develop ‘Rental’ Component.

● Change Type

- Represents a modification work on an existing artifact, mainly due to the change on requirement or business process.
- Example) Modify the ‘Check out car’ procedure.

● Defect Type

- Represents a defect removal work on an existing artifact.
- Example) Fix the defect on *Rental*; computing the fee for a late return.

● Technical Improvement

- Represents an improvement work on an existing artifact.
- Example) Upgrade the DMBS with a latest version.

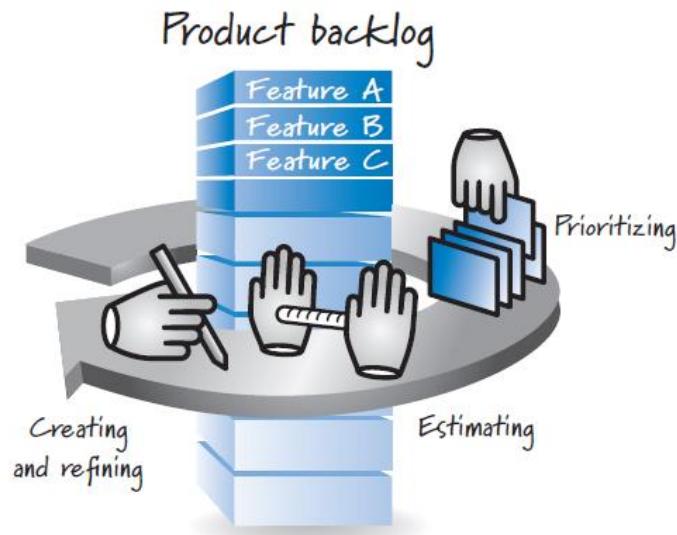
● Knowledge Acquisition Type

- Represents a knowledge acquisition work needed in the project.
- Example) Acquire algorithms for multi-feature machine learning.



Grooming Product Backlog

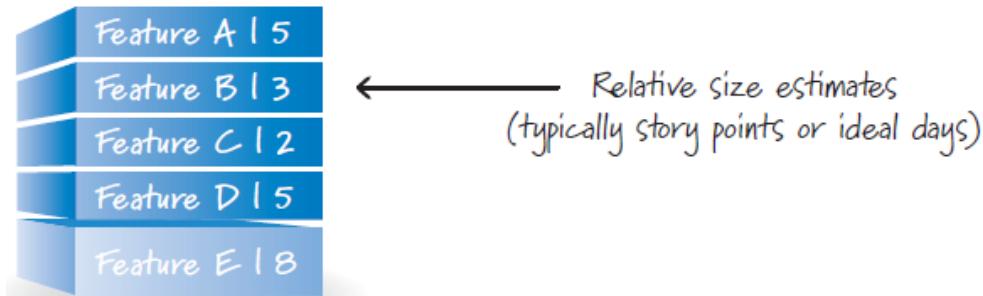
- The activity of creating and refining PBIs, estimating them, and prioritizing them



Estimating Cost of PBIs

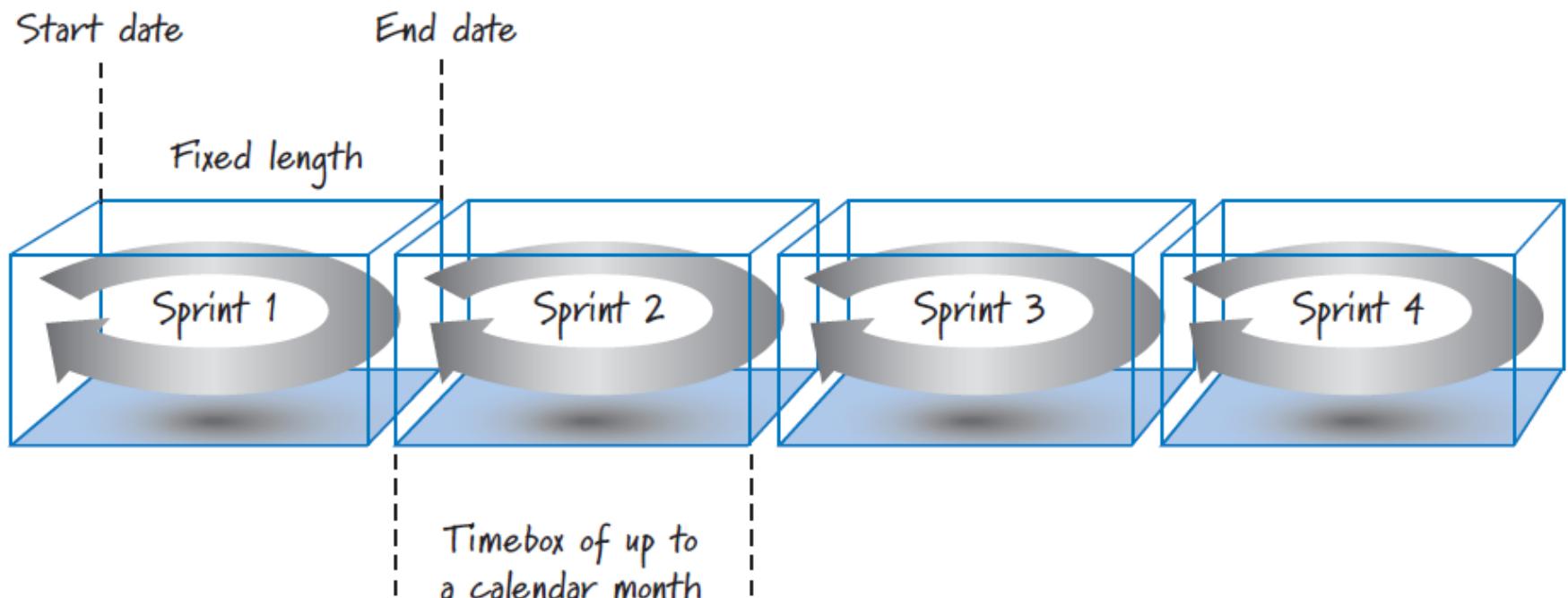
- **Estimating the size/cost of each PBIs**

- Used to properly determine its priority.
- Before finalizing prioritizing, ordering, or otherwise arranging the product backlog
- Which size measures to use
 - Not dictated by Scrum
 - Relative Size Estimates are typically used.



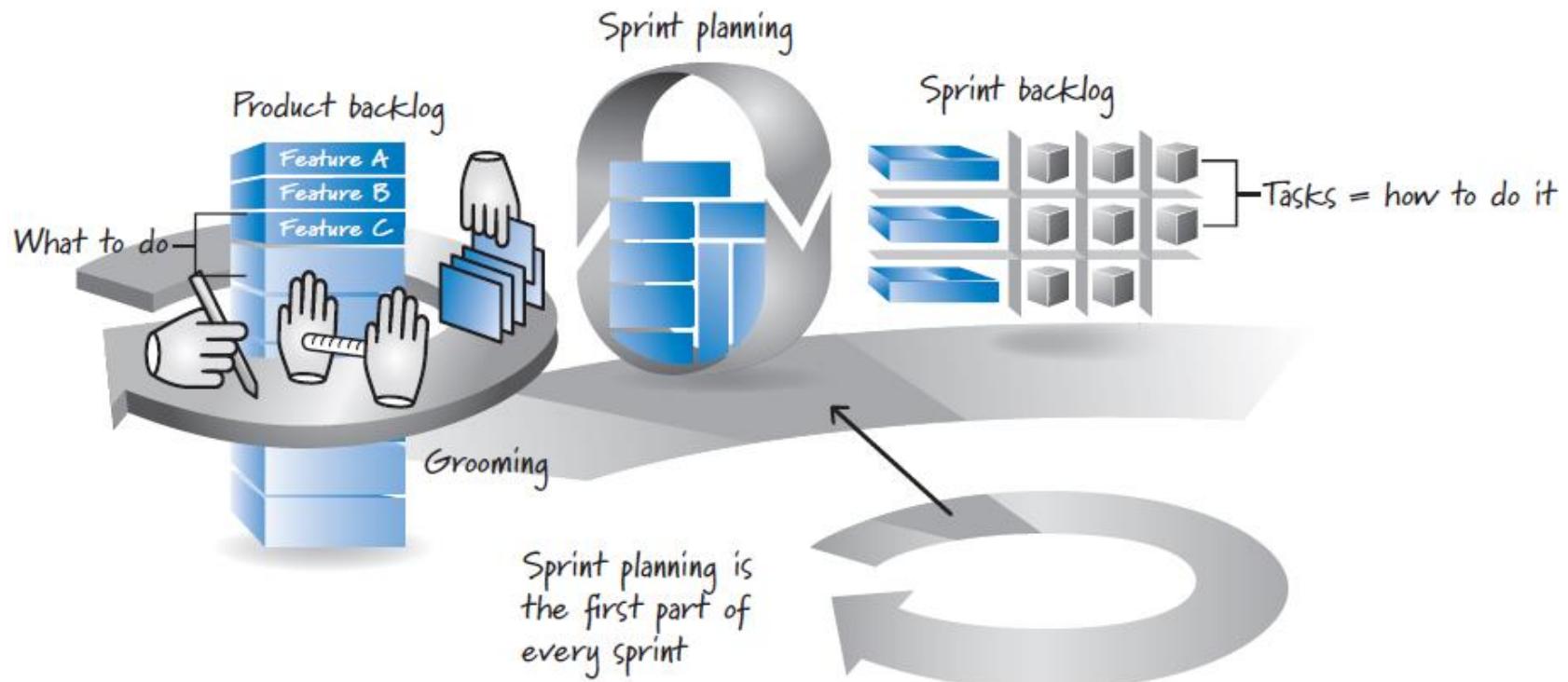
Sprints

- Iterations or cycles of up to a calendar month
- Time-boxed
 - To have a fixed start and end date
 - Generally, to have the same duration



Activity – Sprint Planning (1)

- To determine the most important subset of product backlog items to build in the next sprint
- To spend about four to eight hours in sprint planning
 - For two-week to one-month sprint



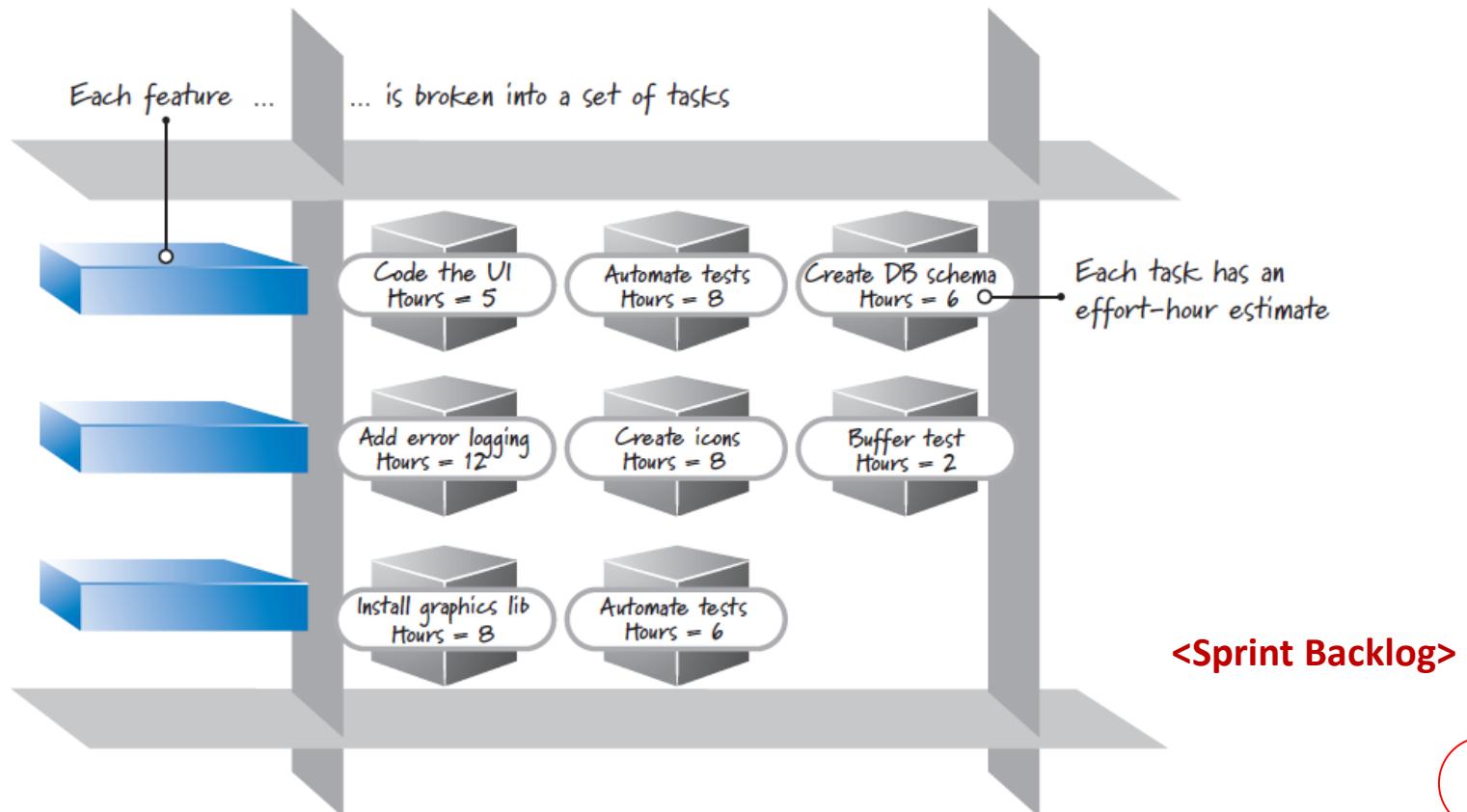
Activity – Sprint Planning (2)

● How to Perform

- To agree on a sprint goal that defines what the upcoming sprint is supposed to achieve
 - Done by the product owner and development team
- To review the product backlog and determine the high priority items
 - That the team can realistically accomplish in the upcoming sprint while working at a sustainable pace
 - Done by the development team

Activity – Sprint Planning (3)

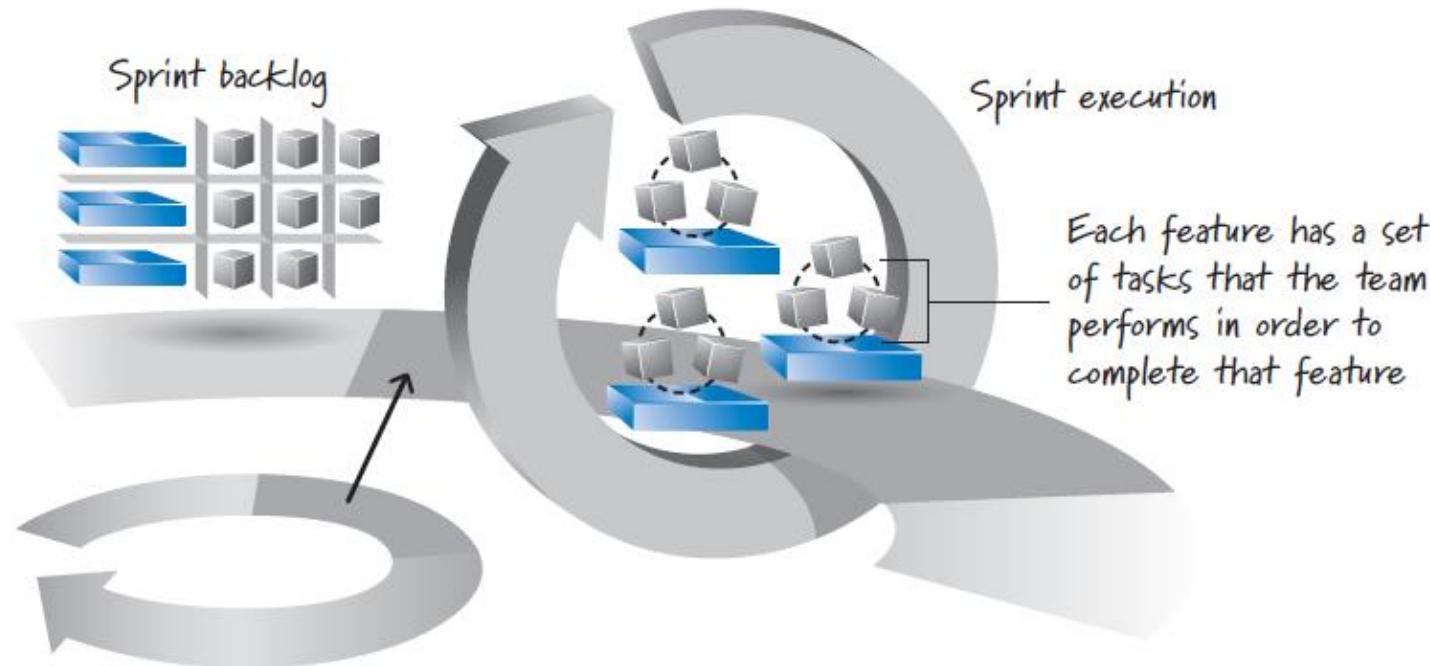
- To break down each targeted feature into a set of tasks, i.e. sprint backlog
- To provide an estimate (typically in hours) of the effort required to complete each task



Activity – Sprint Execution (1)

- To perform all of the task-level work necessary to get the features done
 - Done by the development team
 - Guided by the ScrumMaster's coaching

Sprint execution takes up the majority of time spent in each sprint



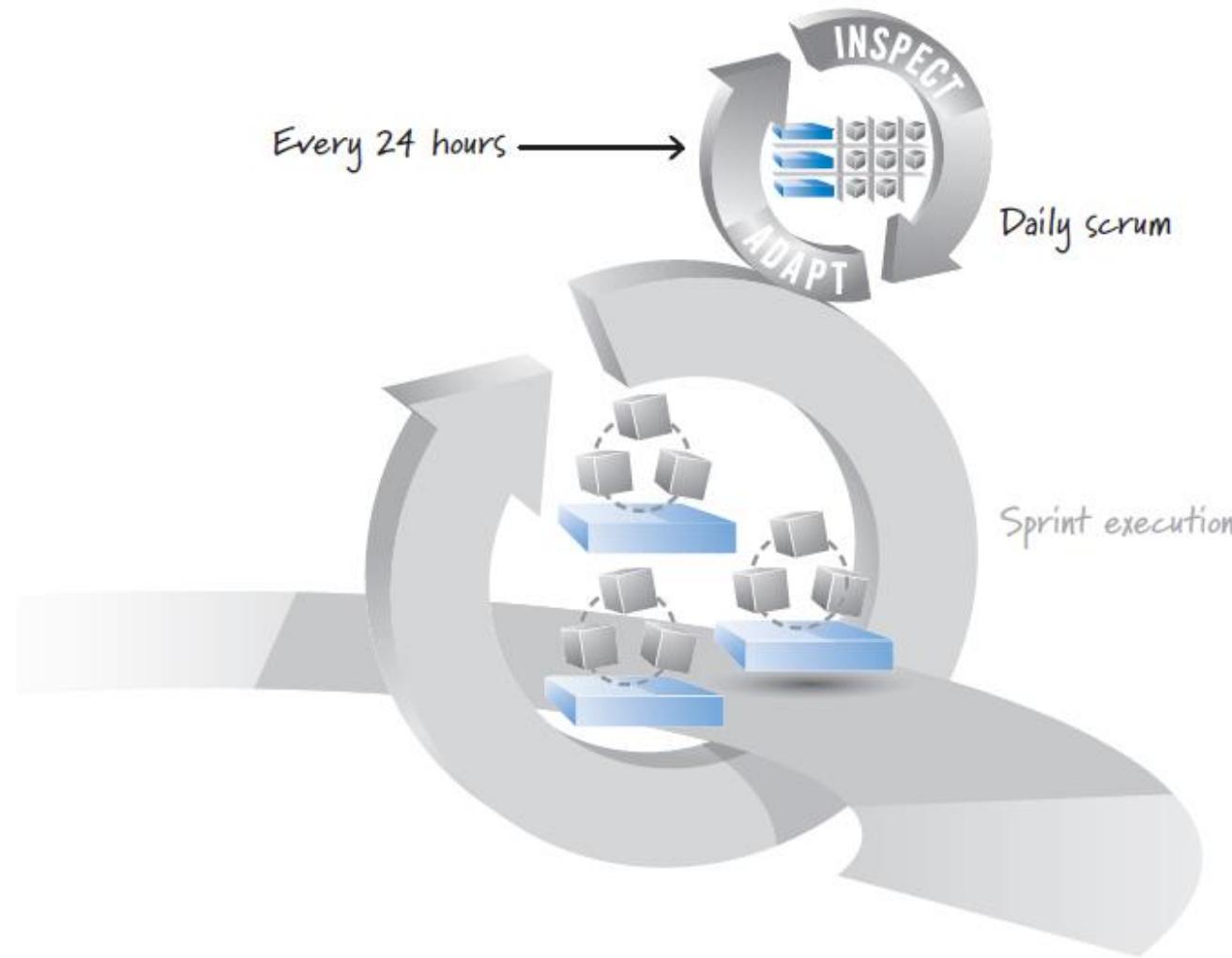
Activity – Sprint Execution (2)

- To define team's task-level works by themselves
- To self-organize in any manner they feel is best for achieving the sprint goal

Activity – Daily Scrum (1)

- **To hold a time-boxed (15 minutes or less) inspect-and adapt activity, each day of the sprint**
 - An inspection, synchronization, and adaptive daily planning activity that helps a self-organizing team do its job better
- **Referred as the daily stand-up**
 - Everyone stands up during the meeting to help promote brevity.

Activity – Daily Scrum (2)



Activity – Daily Scrum (3)

- **A Common Approach to Performing the Daily Scrum**

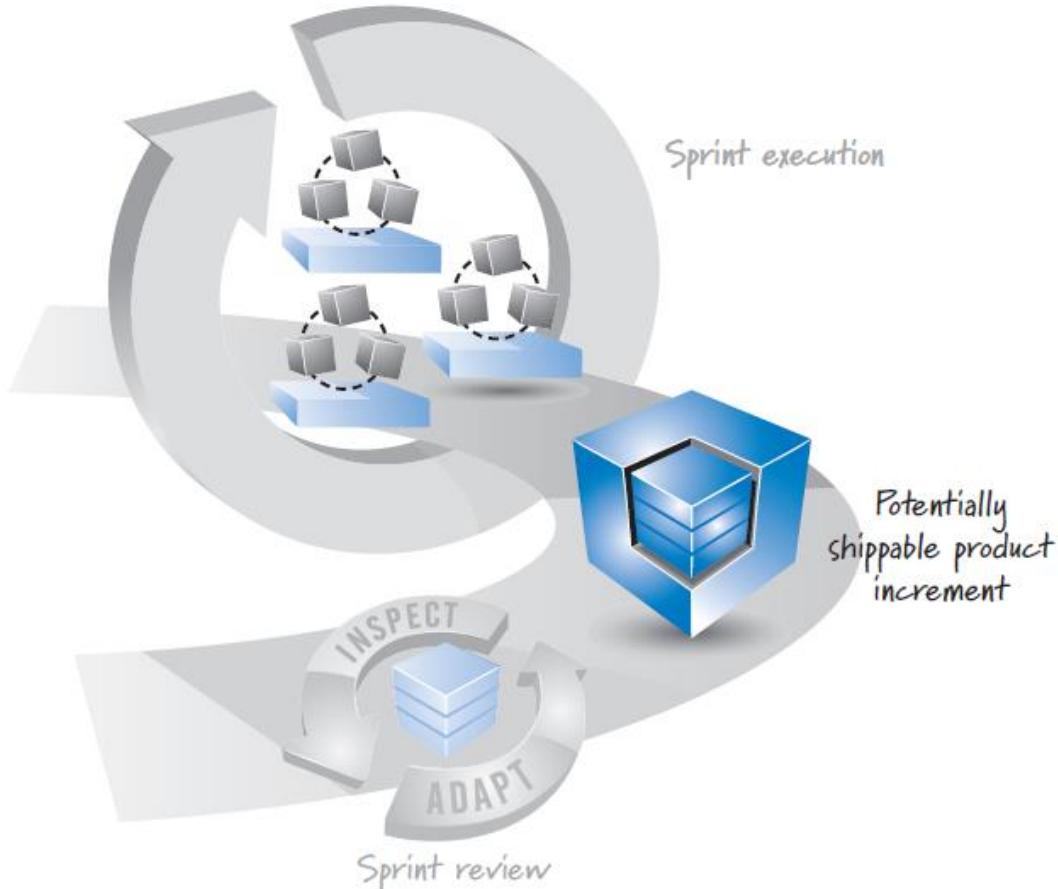
- To have the ScrumMaster facilitating
- To have each team member taking turns answering three questions
 - What did I accomplish since the last daily scrum?
 - What do I plan to work on by the next daily scrum?
 - What are the obstacles or impediments that are preventing me from making progress?

- **Benefits**

- Essential for helping the development team manage the fast, flexible flow of work within a sprint
- Useful to communicate the status of sprint backlog items among the development team members

Activity – Done

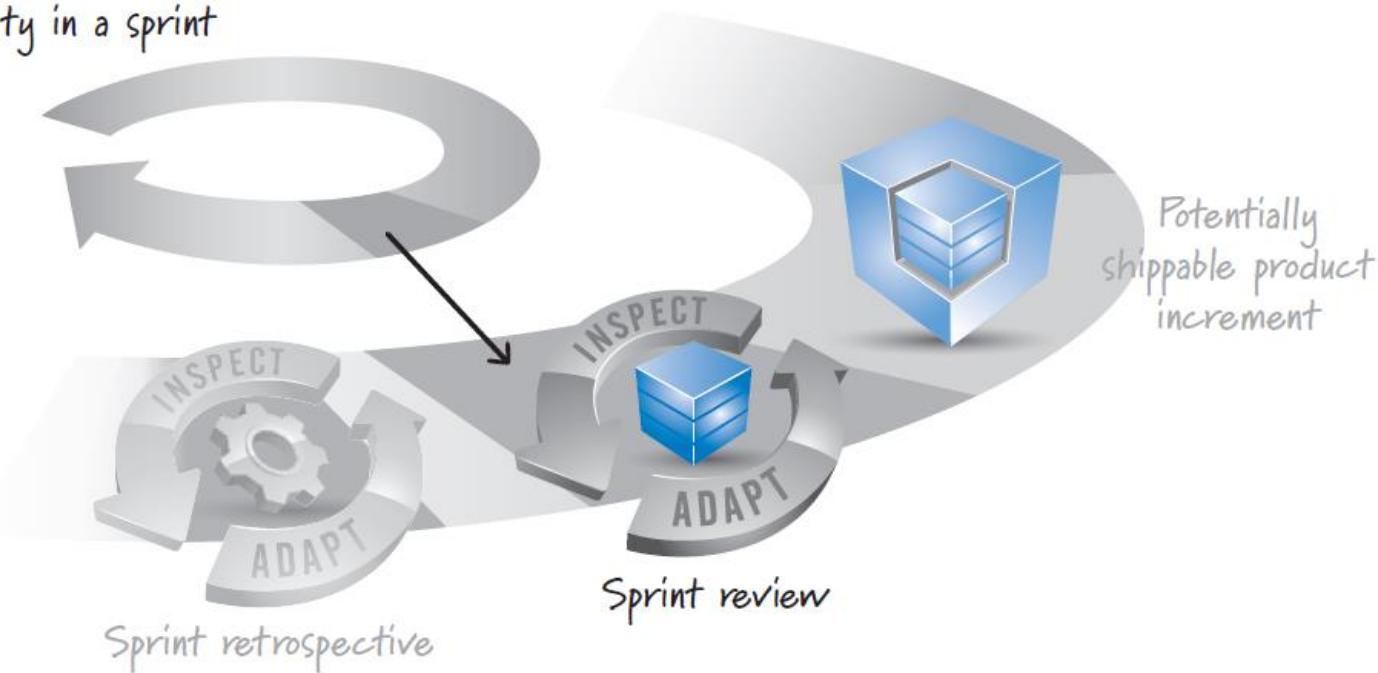
- **To produce potentially shippable product increment**
 - Whatever the Scrum team agreed to do is really done according to its agreed-upon definition of done.
 - There isn't materially important undone work that needs to be completed.



Activity – Sprint Review (1)

- To inspect and adapt the product that is being built

Sprint review is the next-to-last activity in a sprint



Activity – Sprint Review (2)

- **To take place a conversation among its participants**
 - Including the Scrum team, stakeholders, sponsors, customers, and interested members of other teams
 - To review the just-completed features in the context of the overall development effort
- **Objectives**
 - To get clear visibility into what is occurring
 - To have an opportunity to help guide the forthcoming development to ensure that the most business-appropriate solution is created

Activity – Sprint Retrospective

- **To inspect and adapt the process**
 - The development team, ScrumMaster, and product owner discuss what is and is not working with Scrum and associated technical practices.
- **To focus on the continuous process improvement necessary to help a good Scrum team become great**

