



Software Development Processes

What is a Process?

- 💧 When we provide a service or create a product we always follow a sequence of steps to accomplish a set of tasks
 - 💧 No one would bake a cake before all the ingredients are mixed together
- 💧 We can think of a series of activities as a **process**

What is a Process?

- Any process has the following characteristics
 - It prescribes all of the **major activities**
 - It is designed to use resources and **produce intermediate and final products**
 - It may include **sub-processes**.
 - Constraints may apply to activities
(budget control, availability of resources, time etc.)

Software Development Process

- ◆ A structured set of activities required to develop a software system.
- ◆ Many different software processes but all involve:
 - ◆ Specification – defining what the system should do;
 - ◆ Design - defining the organization of the system
 - ◆ Implementation – implementing/coding the system;
 - ◆ Validation – checking that it does what the customer wants;
 - ◆ Evolution – changing the system in response to changing customer needs.

Software Development Process

- ◆ Process **descriptions** may include:
 - ◆ **Products**, which are the outcomes of a process activity;
 - ◆ **Roles**, which reflect the responsibilities of the people involved in the process;
 - ◆ **Pre- and post-conditions**, which are statements that are true before and after a process activity has been performed or a product produced.

Plan-driven and Agile Processes

- ◆ Plan-driven processes are processes where all of the process activities are **planned in advance** and progress is measured against this plan.
- ◆ In agile processes, **planning is incremental** and it is easier to adapt to reflect changing customer requirements.
- ◆ In practice, most processes include elements of both plan-driven and agile approaches.

Software Process Models

- 💧 The Waterfall model
 - 💧 Move on to the next phase only when the current phase is complete.
- 💧 Incremental development
 - 💧 Add new features to build a more complete system over time.
- 💧 Iterative development
 - 💧 Deliver a series of progressively more complete and refined versions of the system over time.

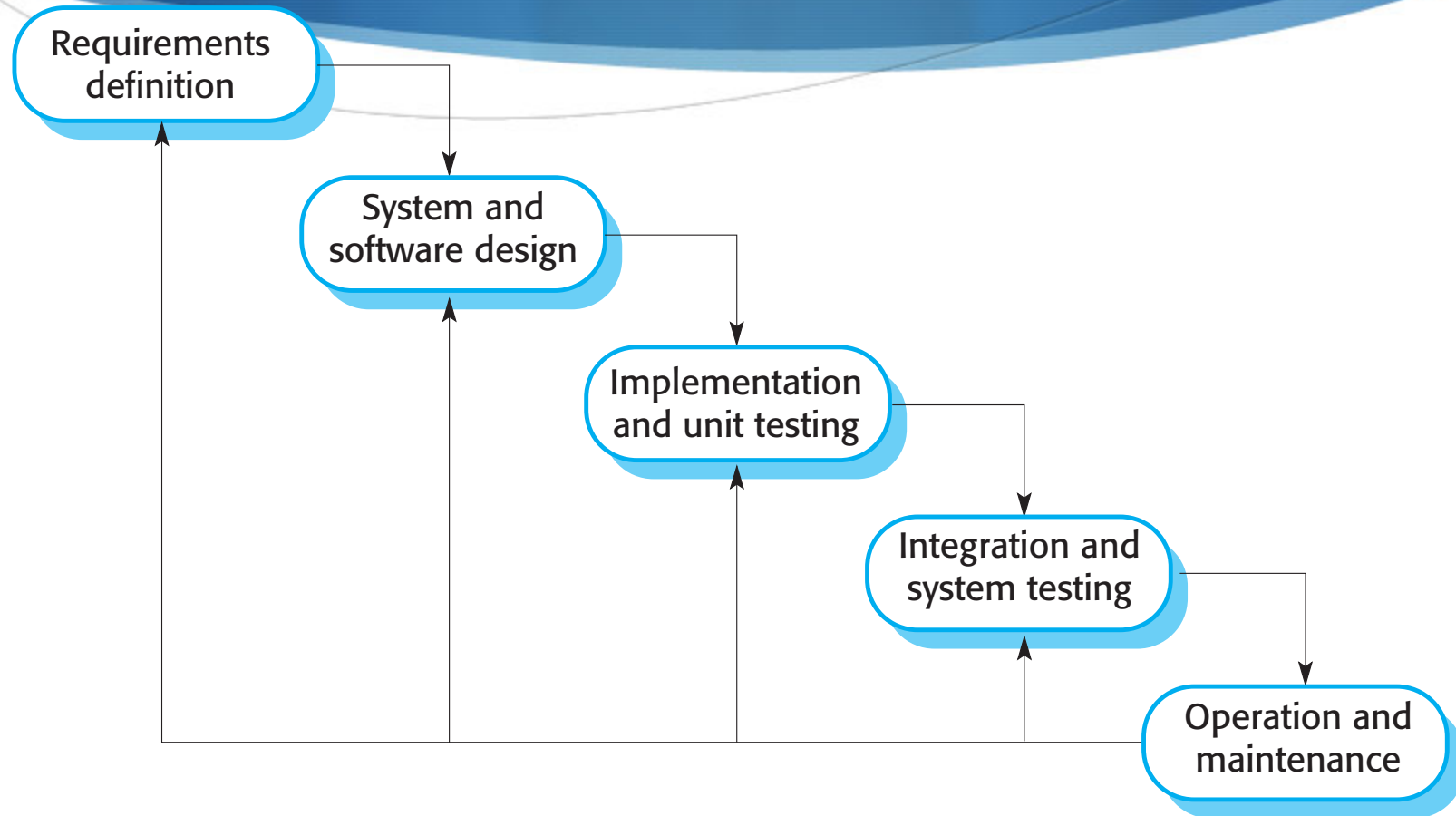
Software Process Models

- ◆ Reuse-oriented software engineering
 - ◆ The system is assembled from existing components. May be plan-driven or agile.
- ◆ In practice, most large systems are developed using a process that incorporates elements from different models.

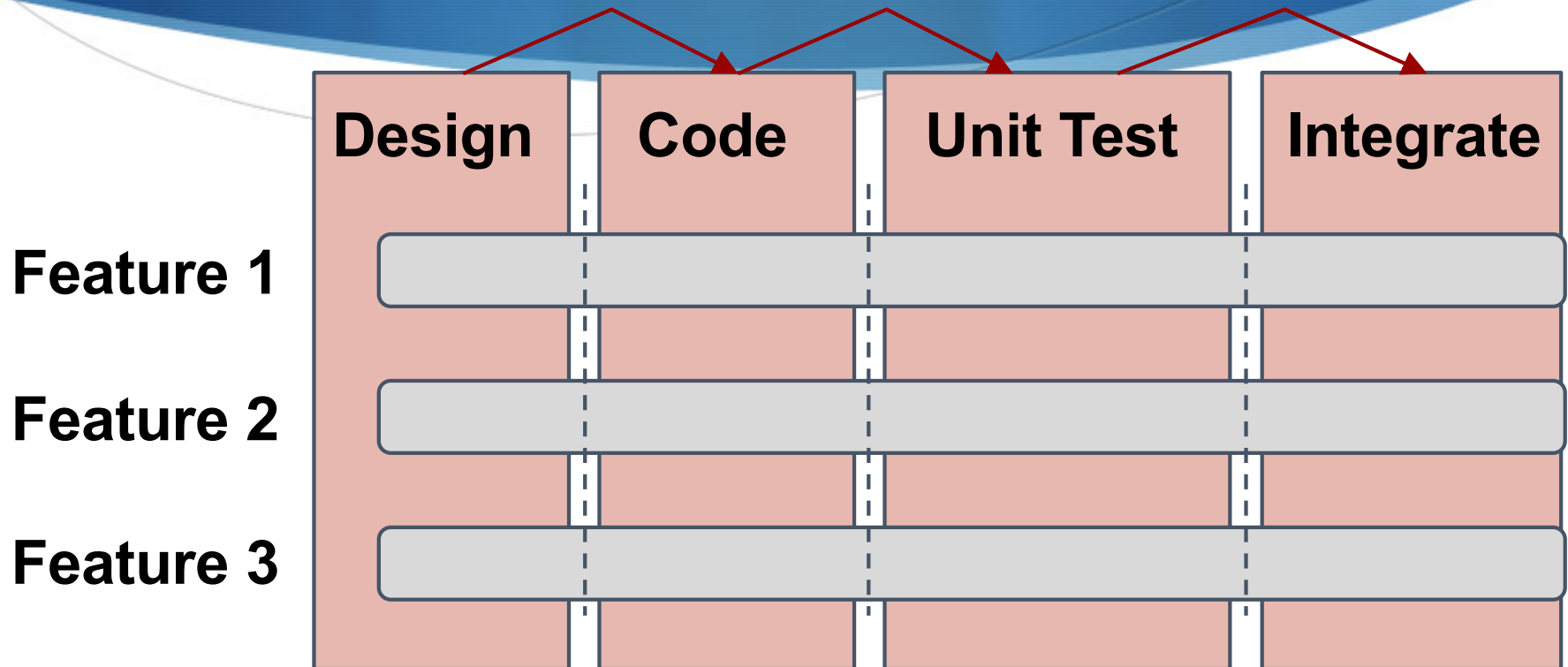


Waterfall Model

The waterfall model



Work Partitioning: Waterfall



Waterfall Model Phases

- ◆ There are distinct phases in the waterfall model:
 - ◆ Requirements analysis and definition
 - ◆ Software design
 - ◆ Implementation and unit testing
 - ◆ Integration and system testing
 - ◆ Operation and maintenance

Waterfall Model Phases and Outputs

Phase	Output Document
Requirements Definition	Requirements Document
System Specification	Functionality specification, acceptance test plan,
Architectural, Interface, and Detailed Design	Architectural Interface, and Detailed Design Specifications. System, Integration, and Unit test plans
Coding	Source Code
Unit, Module, Integration, and System Testing	Testing Reports
Acceptance Testing	Final system and documentation

Applicability

- ◆ Projects where the requirements are stable.
- ◆ Projects where impact of failure is severe.
- ◆ Projects with high turnover or inexperienced developers.

Waterfall Model Advantages

- ◆ Spending more time on earlier phases prevents problems from being discovered later.
- ◆ Brings discipline and structure.
- ◆ Clear understanding of project progress.
- ◆ Places emphasis on documentation.

Waterfall Model Disadvantages

- ◆ Partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements.
 - ◆ This model is only appropriate when the requirements are well-understood and changes will be fairly limited during the design process.
 - ◆ Few business systems have stable requirements.
- ◆ The waterfall model is mostly used for large systems engineering projects where a system is developed at several sites.
 - ◆ The plan-driven nature of the waterfall model helps coordinate the work.

Waterfall Model Disadvantages

- 💧 Hard to respond to unexpected risks.
- 💧 Working system emerge late during the development lifecycle.

Agile Development



The Agile Manifesto

Prefer

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

Agile Principles

- ◆ Satisfy the customer through early and continuous delivery of valuable software.
- ◆ Welcome changing requirements, even late in development.
- ◆ Deliver working software frequently with a preference to the shorter timescale.
- ◆ Business people and developers must work together everyday throughout the project.

Agile Principles

- ◆ Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- ◆ The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- ◆ Working software is the primary measure of progress.

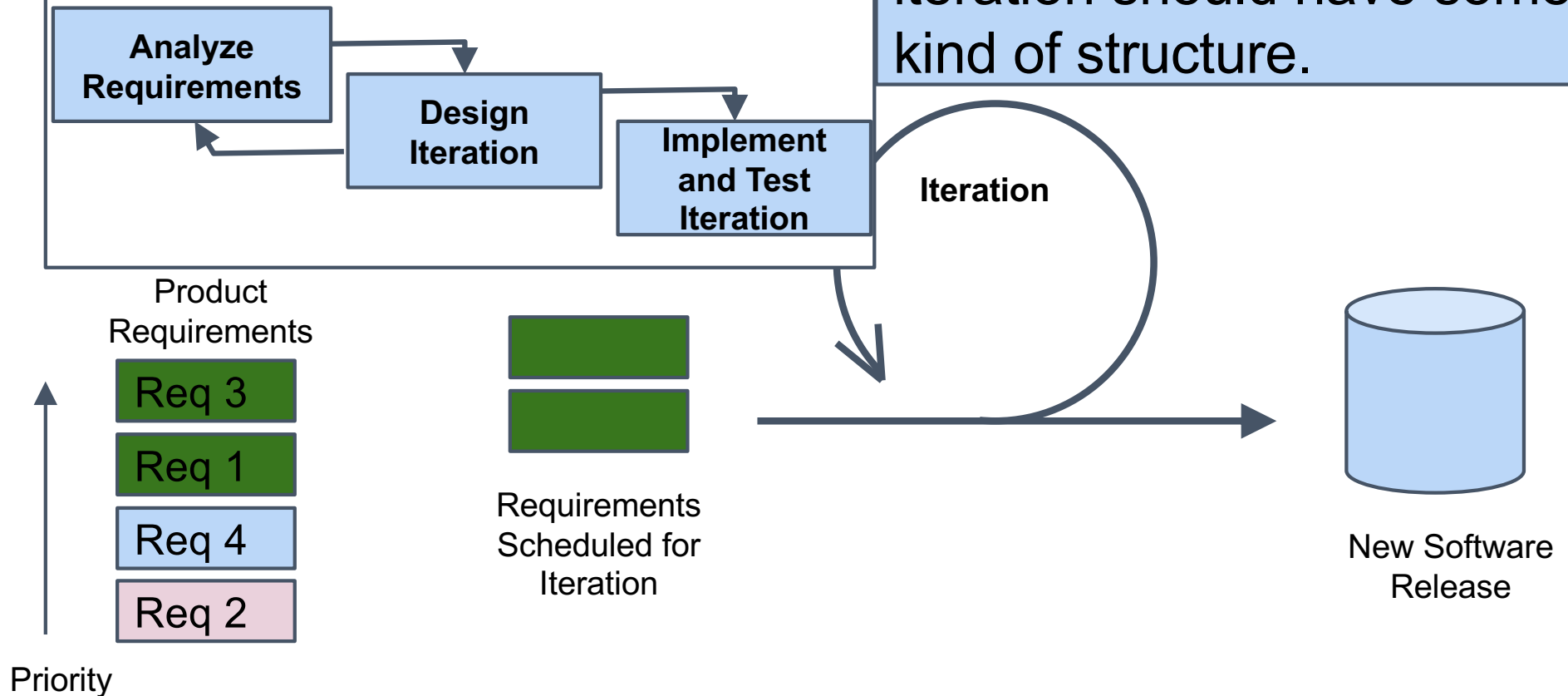
Agile Principles

- ◆ Continuous attention to technical excellence and good design enhances agility.
- ◆ The best architectures, requirements, and designs emerge from self-organizing teams.
- ◆ At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its strategy accordingly.

The Agile Model

An Iteration

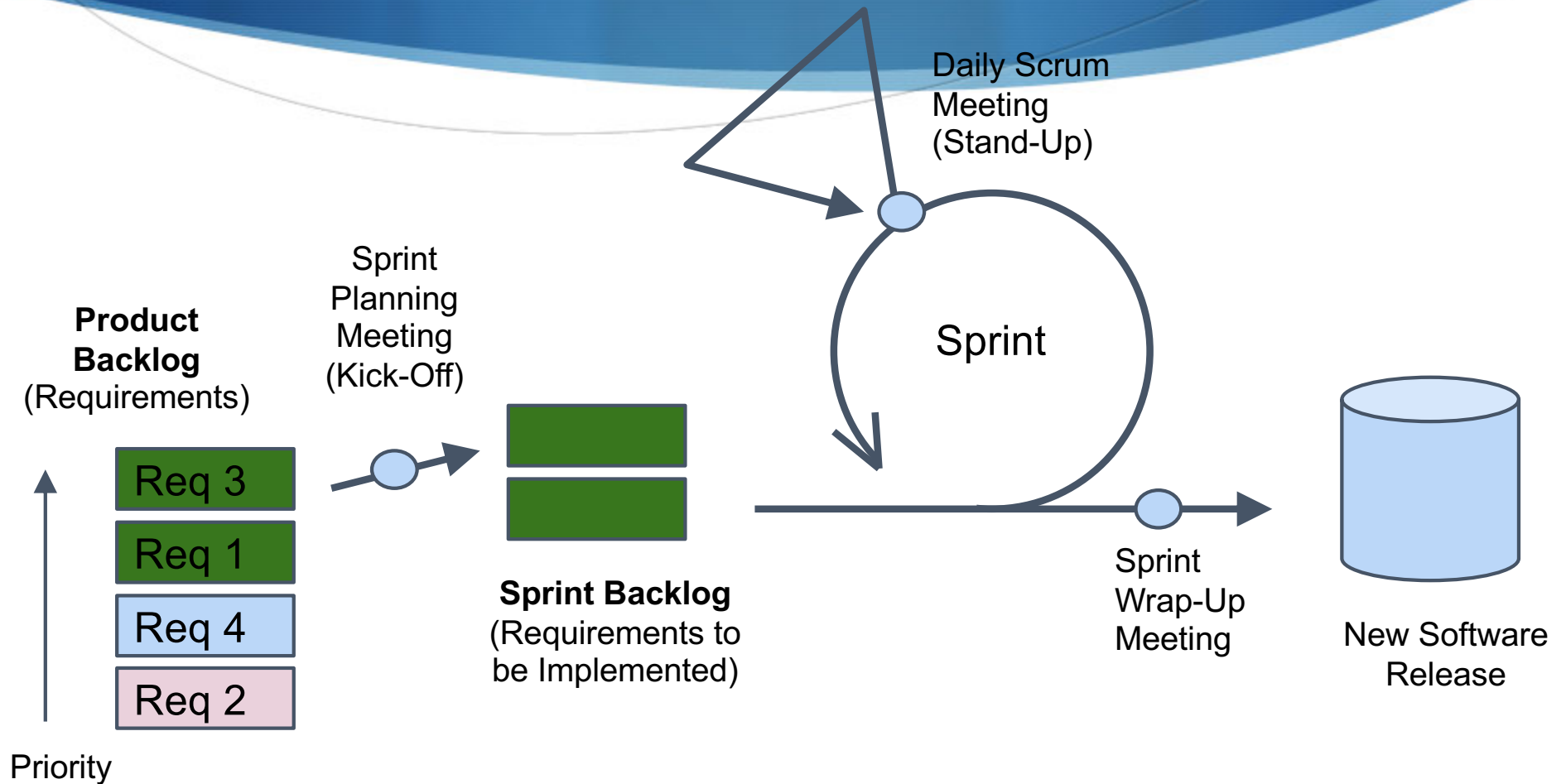
Agile is not ad-hoc. An iteration should have some kind of structure.



Scrum—An Agile Development Process



The Scrum Model



Scrum Roles

- ◆ **The Development Team**

- ◆ Usually small, 3-9 people.

- ◆ **The Product Owner**

- ◆ The “voice” of the customer.
- ◆ Communicates with customers and provides feedback to the team.
- ◆ Maintains and prioritizes the requirement backlog

- ◆ **The Scrum Master**

- ◆ The team “coach”.
- ◆ Removing impediments to success.
- ◆ Facilitating communication and meetings.

Daily Scrum Meetings

- ◆ A daily 15-minute meeting in which all participants are standing.
- ◆ Each person answers three questions:
 - ◆ What did you complete since the last scrum?
 - ◆ What will you complete before the next scrum?
 - ◆ What, if any, blocking issues (impediments to progress) do you need to resolve?

The Scrum Model: Advantages

- ◆ Very responsive to requirements change.
- ◆ Team members can choose how they approach their development responsibilities, as long as they still meet team goals.

The Scrum Model: Disadvantages

- ◆ Often results in a lack of documentation.
- ◆ Relies on a good scrum master to keep meetings productive, and relies on the communication skills of the team members.
- ◆ Contracts with customers may be challenging.
- ◆ Prioritizing changes can be difficult where there are multiple stakeholders.

Project Context Analysis

	1 (Low)	2	3	4	5 (High)
Potential loss due to defects/bugs					
Developers' experience/skills					
Rate of requirements change					
Team size (5, 10, 25, 50, 100+)					
Organization culture (adaptive to change)					
Pressure to develop early releases					
Business staff's commitment to work extensively with development team					
Developer's experience with similar systems					
Availability of reusable components					

References

- ◆ A number of slides are taken/adapted from <https://greg4cr.github.io/courses/spring19csce247/index.html>.
 - ◆ with permission from Gregory Gay.
- ◆ Software Engineering (9th edition) by Ian Sommerville.
- ◆ Software Engineering A Practitioner's Approach (8th edition) by Roger S. Pressman and Bruce R. Maxim.
- ◆ Fundamentals of Software Engineering (2nd edition) by Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli.
- ◆ Software Engineering Theory and Practice (4th Edition) by Shari Lawrence Pfleeger and Joanne M. Atlee.