

SE 329 – Software Project Management

Software Development Lifecycle

Lotfi ben Othmane

Goal

- What is a development lifecycle?
- What are the common software development approaches?
- Why do we need many approaches?
- Why applied development processes are different?

Main Software Development Activities

1. Requirements elicitation and analysis
 2. Solution design
 3. Coding
 4. Testing
 5. Releasing
- Operation is part of the life-cycle

Sequencing Software Development Activities

Exercise - What is the sequence of the project development activities of your selected project?

1. Requirement elicitation and analysis
 2. Solution design
 3. Coding
 4. Testing
 5. Release
- Operation is part of the life-cycle

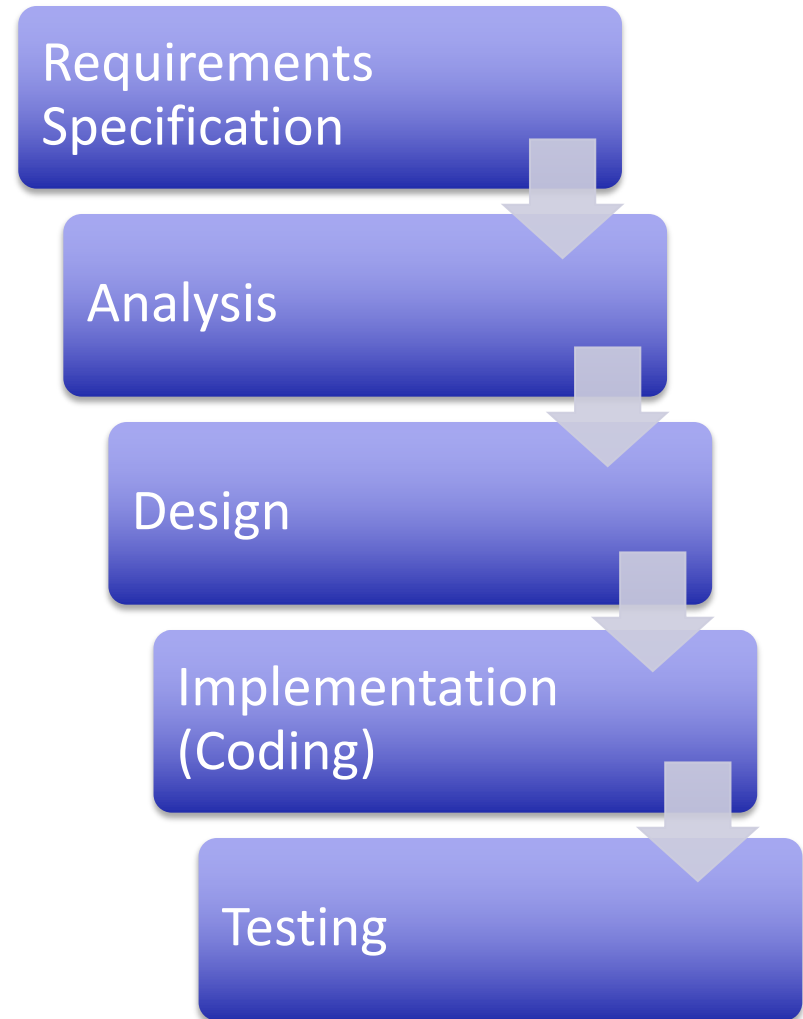
Sequencing Software Development Activities

Common approaches

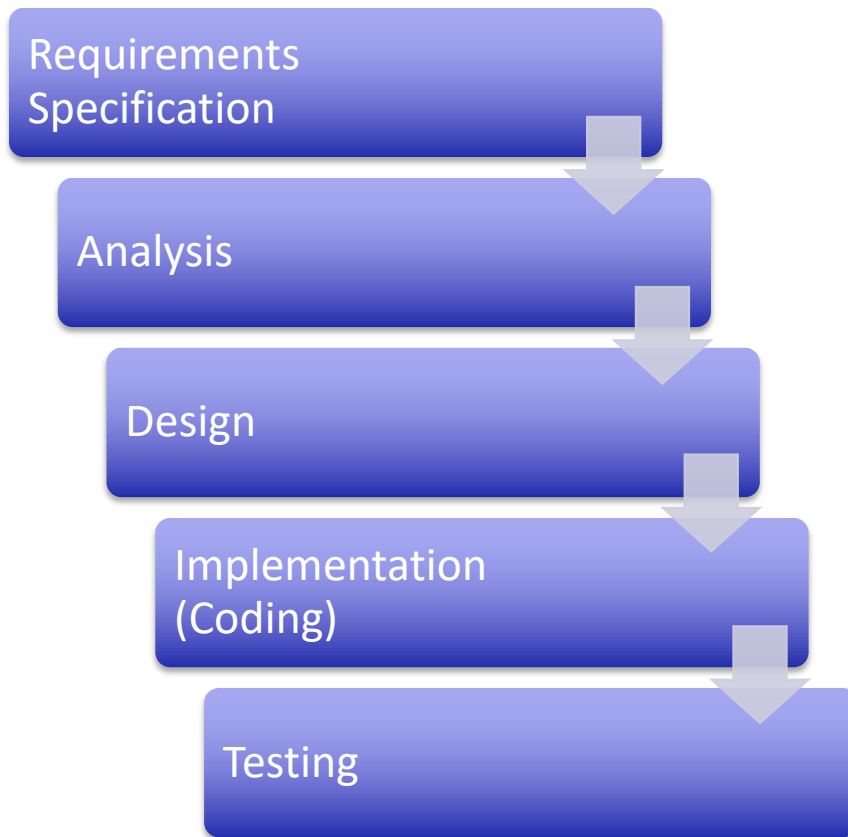
1. Waterfall model
2. Iterative model
3. Agile approach

Waterfall Model

- Process viewed as a sequential set of activities
 - Elicit requirements, analyze and design, code, test, release
 - Prototyping could be part of requirements determination
- Finish one stage before moving to the next
 - Backtrack if necessary



Assumptions for the Waterfall Model



1. Requirements are understood and specified before code is designed
2. Requirements analyst produces a real written specification
 - Significant effort to develop useful specification
 - Evaluate for completeness, consistency, etc.
3. Software are built in accordance with written requirements
 - Like a checklist

Iterative Model

- Process viewed as a sequence of iterations, each building on the last
- Build minimal useful subset, test, release, build next version by extension. Early iterations may be prototypes

What is an Iteration?



An iteration is a distinct sequence of activities based on an established plan and evaluation criteria, resulting in an executable release (internal or external)

IBM

Assumptions of the Iterative Model

1. Requirements can be understood well enough to build a minimal useful subset
2. Early iterations allow for extension of subsets
 - Clearly identified model structure

What is an Iteration?



An iteration is a distinct sequence of activities based on an established plan and evaluation criteria, resulting in an executable release (internal or external)

IBM

The Agile Approach

<https://agilemanifesto.org/>



Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Success of a given project is measured by the satisfaction of the customer

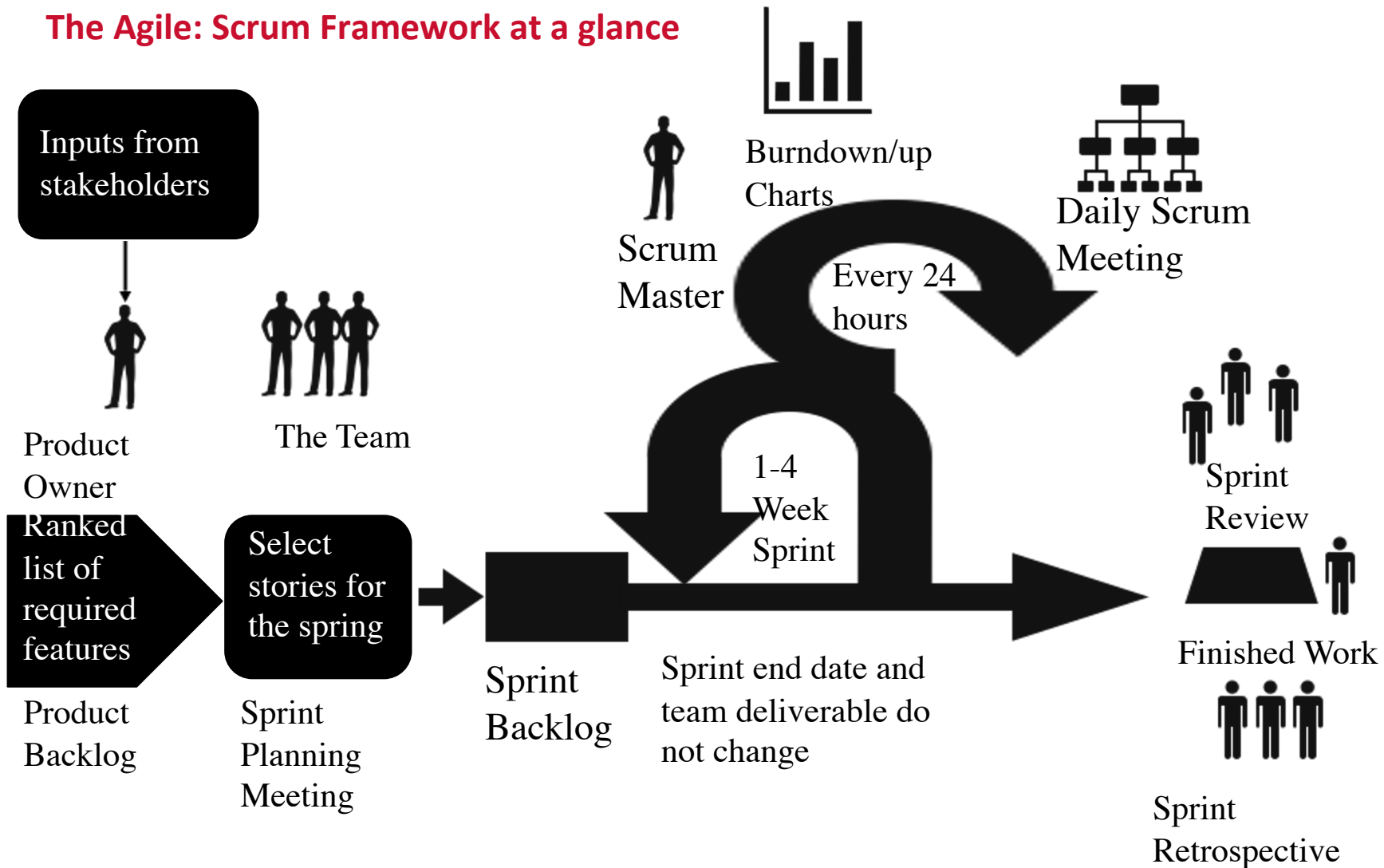
The Agile Approach

- Many small, quick iterations, known in scrum as sprints
- Each iteration implements a user story
- Client validates increments

There are several agile methods such as Scrum

The Agile Approach

The Agile: Scrum Framework at a glance



Assumptions of the Agile Approach

- Requirements gathered informally from customers are binding
 - Code is the only record
- Requirements cannot be understood before code is developed
- Requirements could be implemented in small increments

Discussion – Development Processes

If you would have to work on a project like the one you have selected in the previous discussion, which one of the following approaches would you use?

1. Waterfall
 2. Iterative
 3. Agile
- Why?

Variability of Development Approaches

- There are different approaches and different assumptions
- Software engineering provides a range of processes and methods to address these difficulties, for example:
 - Waterfall: conformance and regulation
 - Iterative: risk mitigation
 - Agile: rapid customer feedback

Process

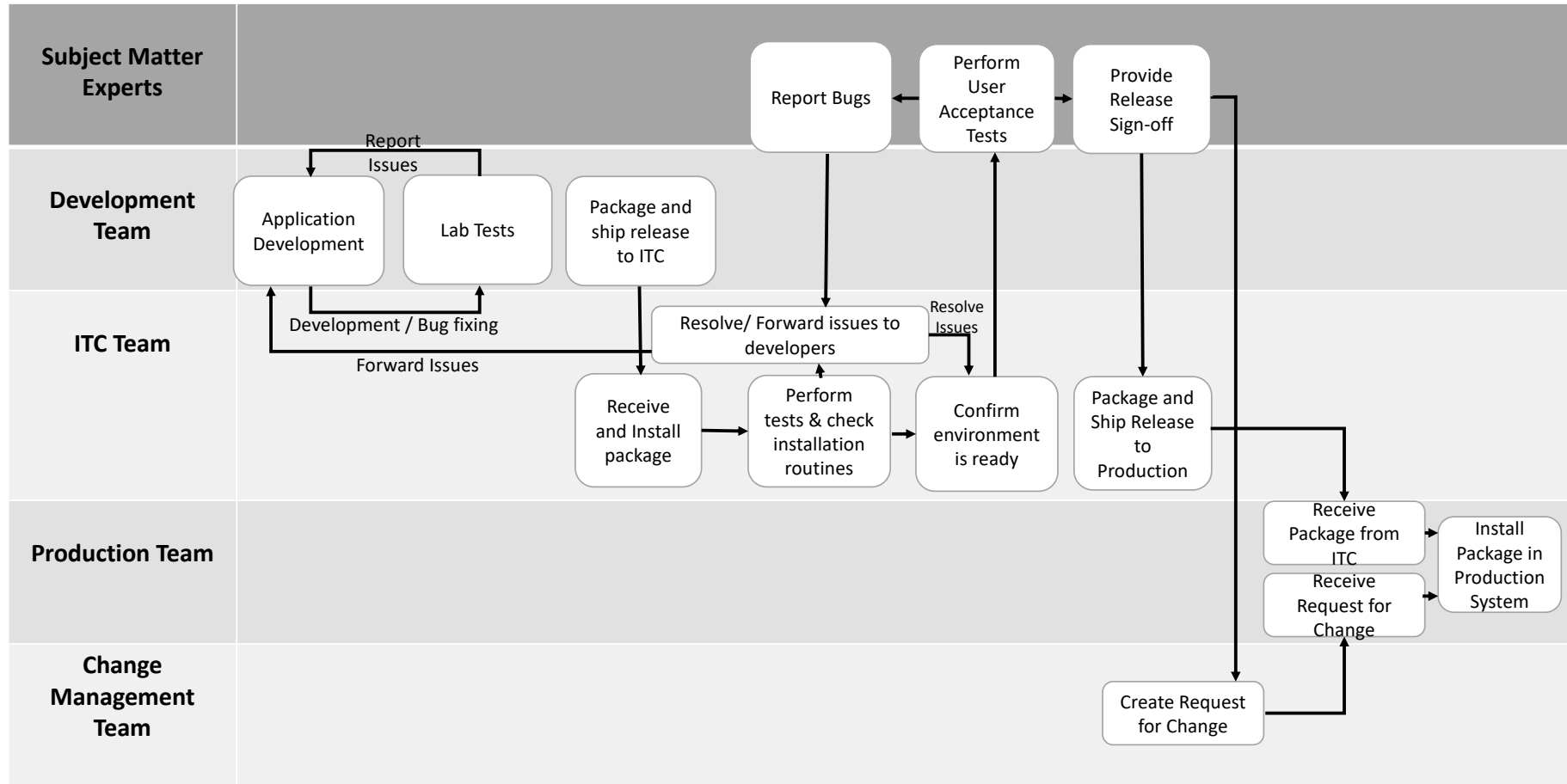
- A **process** is composed of a set of artifacts, activities, and roles and criteria to progress from an activity to another.
- **Activities** are performed by **roles** to produce **artifacts**
- Examples
 - Artifact: Requirements specification
 - Activity: Define requirements
 - Specify
 - Review
 - Criterion: Requirements must be reviewed before proceeding to design phase
 - Role(s): Systems Engineer, Architect, Customer

Software Development Processes

- Approaches are generic - like theories
- Processes help with:
 - **Work assignments**
 - are properly divided and assigned
 - result in code that work together
 - Modules work together to produce the desired result
 - **Collaboration**
 - The team members at a given site understand the team communication approach and the work of colleagues at other sites
 - Test each others code

Software Development Processes

– an Example



Constraint Examples for Development Process Engineering

1. Project duration and accountability
2. Form of requirements, design, test plan
 - Written document
 - Knowledge in the heads of the development team
3. Review procedures for documents and code
 - Formalized inspections with criteria for passing to next step
 - Informal peer review meetings
 - Office mate reviews
4. Release criteria

Constraint Examples for Development Process Engineering

5. Roles: Project manager, systems engineer, architect, developer, tester
 - Dedicated people? Shared roles?
6. Criticality: Critical projects lead to more formalized process
 - Avionics, medical software, defense

Tailoring the Process

- Process evolves with the project
- Consider what motivates people?
 - Having an impact?
 - Frequency of feedback?
- Pick appropriate roles, artifacts, activities, modes of communication
- Tailoring process to improve the efficiency of the team
 - Projects as opportunities for new skills, tools, resources

Participation– Development Process

Assume you are assigned to manage a software development project. Develop a simple development process for your team.

Software Development Activities

Practice 2 - Assume you are assigned to manage a software development project. Develop a simple development process for your team.

1. Identify the activities
2. Identify the roles
3. Assess the characteristics of the project: review, criticality, etc.?
4. Identify the appropriate sequence of activities
5. Other steps if needed

Self-Assessment

- What are the common software development approaches?
- Why do we need many?
- Why applied processes are different from the announced ones?

Thank you