Lecture 1.2
Web Development Process Model

Course Instructor
Engr. Madeha Mushtaq

Outline

- Development Process model
  - software development process activities
- Requirement for a web development process model
- Rational unified process model (RUP)
  - suitability for web application development

SDLC Vs. Software Development Process Model

- SDLC: Overall process that software development teams use to plan, design, build, test, and deploy software.
- The SDLC typically includes the following phases:
- Requirements gathering
- Design
- Implementation
- Testing
- Deployment
- Maintenance

SDLC Vs. Software Development Process Model

- Software development process model: is a specific framework or methodology that software development teams use to guide their work through the software development life cycle.
- Some examples of software development process models include:
- Waterfall model:
- Agile model:
- DevOps model:

SDLC Vs. Software Development Process Model

- Waterfall model: A linear approach to software development that emphasizes thorough planning and documentation.
- Agile model: An iterative approach to software development that emphasizes collaboration, flexibility, and rapid feedback loops.
- DevOps model: An approach that emphasizes continuous integration and continuous delivery to streamline the software development process and speed up the release cycle.

1. Process model

- A set of related activities that leads to the production of a software product
  - development of software from scratch
  - extending and modifying existing systems
- Common activities
  - Software specification
  - Designing and implementation
  - System validation
  - System evolution

1. Process model

- The well-known software development processes can be grouped into two categories:
- Lightweight processes – better known as agile processes and
- Heavyweight processes.
- “Light” or “heavy” refers to the degree of process formalization, i.e., how many documents and models are created.

1.1 Process activities

- **Software specification:**
- The functionality of the software and constraints on its operation must be defined
  - critical stage (can lead to problems in design and implementation)
- Activities:
  - Feasibility study
  - Requirement elicitation and analysis
  - Requirement specification
  - Requirement validation

1.1 Process activities…

- **Software design and implementation:**
- Design is the description of
  - System structure
  - Data models
  - Interface between components
- Implementation: Converting a system specification into an executable system

1.1 Process activities…

- **System validation:**
- Intended to show that the system
  - confirms its specification
  - meets customer's expectations
- Development testing
  - tested by the people developed the components
- System testing
  - finding component integration errors
- Acceptance testing
  - System is tested by the customer's provided data

1.1 Process activities…

- **Software evolution:**
- Software is flexible as compared to hardware
  - Changes can be made to the system during development or after the development.

2. Requirements for a web application development process

- Evolving from informational medium to application medium
- Existing approaches are over-pragmatic
  - lead to short development time
- Web engineering does not have its own mature development process model.
- SE development process models are adopted.

2. Requirements for a web application development process…

- Handling Short development cycles
  - Development time is short
    - Normally does not exceed six month
  - Immediate delivery mechanism
    - Capture share in the market
  - Leaves less freedom for systematic development process.

2. Requirements for a web application development process…

- Handling changing requirement
  - Requirements often emerge during development
    - as developer understand the unknown business
  - Integrate changes rapidly to remain in competition
  - User involvement is more critical
    - due to emerging and unstable requirements

2. Requirements for a web application development process…

- Reuse and integration
  - to meet time constraints developer try to reuse components
    - Leads to integration issues
  - Development can not be isolated from the development of other applications within the organization.

2. Requirements for a web application development process…

- Adapting to web application's complexity level
  - process depends upon the level of complexity
  - process is adapted dynamically
    - for low complexity, it should be like lightweight process
    - for high complexity, it should be like heavyweight process

3. Rational unified process

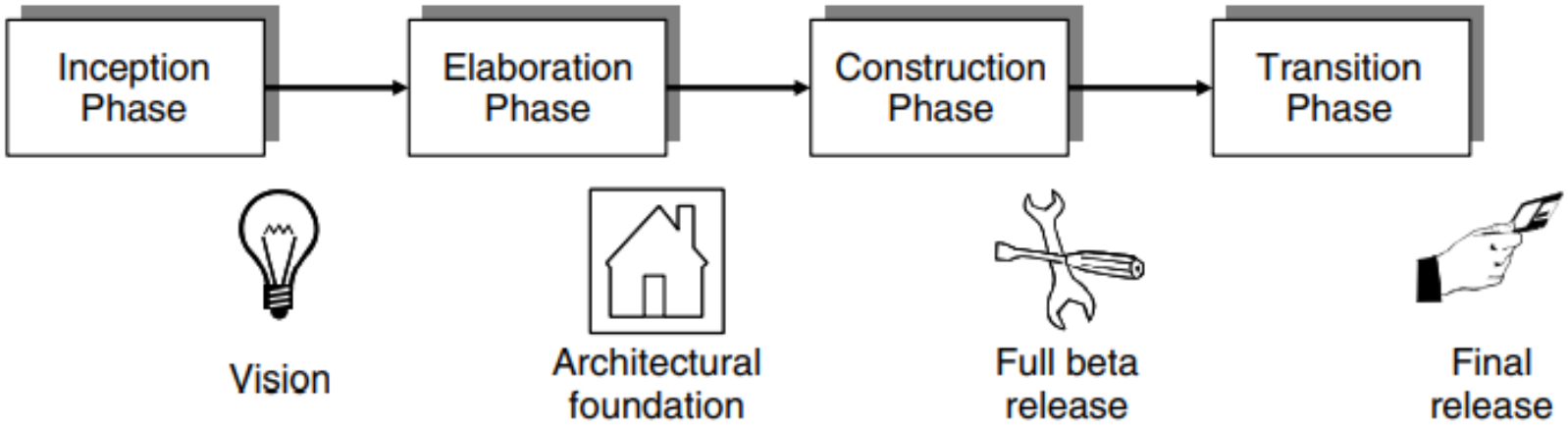
- RUP is a heavyweight, phase oriented, incremental and iterative process.
- Described in three perspectives
  - Dynamic perspective: phases
  - Static perspective: activities in phases
  - Practice perspective: good engineering practices

3. Rational unified process

- **RUP phases:**
  - Inception
  - Elaboration
  - Construction
  - Transition

3. Rational unified process

- **RUP phases:**



3. Rational unified process

- **RUP phases:**
- **Inception:** Define the business case for the project
- Goals:
  - Business case
    - Identify and interact with external entities
    - Asses the business contribution
- Artifacts:
  - business case

3. Rational unified process

- **Elaboration:** establish understanding with the problem
- Goals:
  - Establish software scope
  - Discriminating critical use-cases
  - Estimating cost, schedules and risks
- Artifacts:
  - development plan, use-case model, architectural description

3. Rational unified process

- **Construction:** involves system design, programming and testing
- Goals:
  - Develop the design
  - Implement the design
  - Validate the system
- Artifacts:
  - System, training material

3. Rational unified process

- **Transition:** Installing the system in real environment
- Goals:
  - Testing in real environment
  - training
  - Bug fixing, performance enhancements
- Artifacts:
  - A documented system working correctly

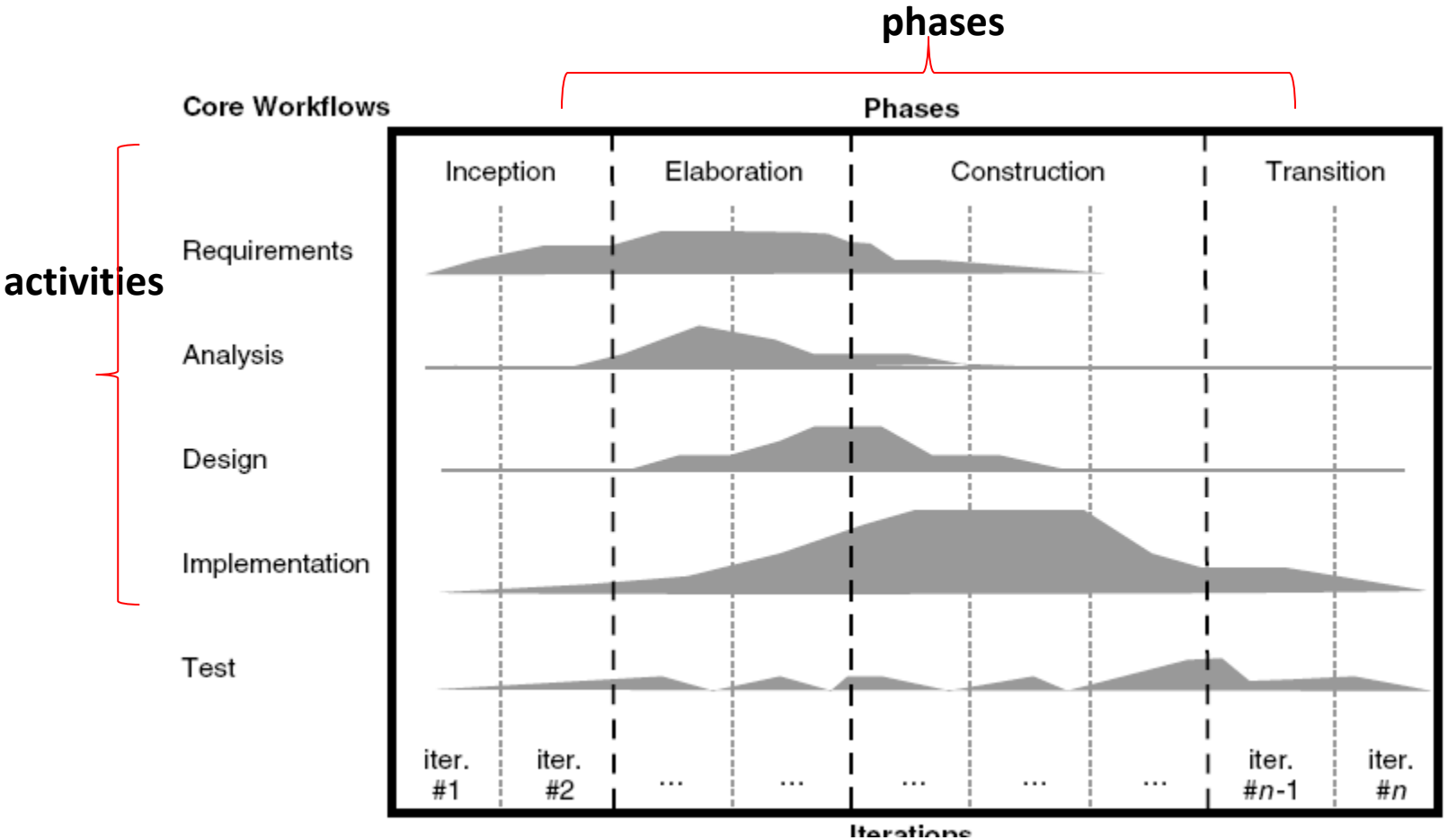
3. Rational unified process

- RUP activities (workflows):
  - Requirements
  - analysis
  - design
  - implementation
  - test

3. Rational unified process

- RUP good practices:
  - Develop software iteratively
  - Manage requirements
  - Use component-based architectures
  - Visually model software-using UML
  - Verify software quality
  - Control changes to software

3. Rational unified process…



3.1 RUP for web application

- Inception phase:
- Definition is problematic for web application
  - no concrete view of the system at beginning
  - has target group but needs are unknown
- Elaboration phase:
  - due to short development time, first version has priority over clearly defined end-product

3.1 RUP for web application…

- Construction phase:
  - exists in web development process
- Transition phase:
  - is meaningful for web application development

3.1 RUP for web application…

- Handling short development cycles:
  - Conflicting
    - short cycle means concession in modeling and documentation while RUP is heavyweight.
- Handling changing requirements:
  - Conflicting with time constraints
    - require concrete vision at the end of inception phase which require more time in web application due to evolving requirements

3.1 RUP for web application…

- Parallel development of different releases:
  - can be met with RUP
    - RUP only allow parallel development in construction phase
- Reuse and integration:
  - Conflicting
    - It requires coordination with development processes of other applications RUP does not describe this

3.1 RUP for web application…

- Adapting to a Web application's complexity level:
  - RUP can be adopted for later stages when complexity of web application is understood

Extreme Programming (XP)

- Extreme Programming (XP) Extreme Programming is an example of an agile process model.
- XP is a light weight process model.
- Does XP meet the Requirements of Web Application Development?
- Are there any other process models that are suitable for web application development?

References

- Web Engineering, The Discipline of Systematic Development of Web Applications, Chapter 10, G. Kappel, B. Proll, S. Reich. & W. Retschitzegger.