Software Development Methodologies

Lecturer: **Raman Ramsin**

Lecture 15

Agile Methodologies: AUP

Agile Unified Process (AUP)

- Proposed by Ambler as a simplified version of the Rational Unified Process (RUP).
- Introduced in 2005; revised in 2006.
- Describes a simple, easy to understand approach to developing business systems using agile techniques and concepts while remaining true to the RUP.

Agile Unified Process (AUP)

- UML is the main modeling language, but AUP is not restricted to UML.
- Use case driven
- Iterative-incremental
- Architecture-centric
- Covering the full generic lifecycle

AUP: Process – Phases

- Like RUP, the overall development *cycle* consists of four *phases:*
  - ***Inception:*** The goal is to identify the initial scope of the project, a potential architecture for the system, and to obtain initial project funding and stakeholder acceptance.
  - ***Elaboration:*** The goal is to prove the system architecture.
  - ***Construction:*** The goal is to build working software on a regular, incremental basis which meets the highest-priority needs of project stakeholders.
  - ***Transition:*** The goal is to validate and deploy the system into the production environment.

AUP: Process – Iterations and Disciplines

- Each phase can be further broken down into *iterations.*
- An iteration is a complete development loop resulting in a release of an executable increment to the system.
- Each iteration consists of seven work areas (*disciplines*) performed during the iteration.
- For each discipline, AUP defines sets of:
  - *artefacts* (work products);
  - *activities* (units of work on the artefacts);
  - *roles* (responsibilities taken on by development team members).

AUP: Process – Disciplines

1. **Model**. The goal is to understand the business of the organization, the problem domain being addressed by the project, and to identify a viable solution to address the problem domain.

2. **Implementation**. The goal is to transform the model(s) into executable code and to perform a basic level of testing, in particular unit testing.

3. **Test**. The goal is to perform an objective evaluation to ensure quality. This includes finding defects, validating that the system works as designed, and verifying that the requirements are met.

4. **Deployment**. The goal is to plan for the delivery of the system and to execute the plan to make the system available to end users.

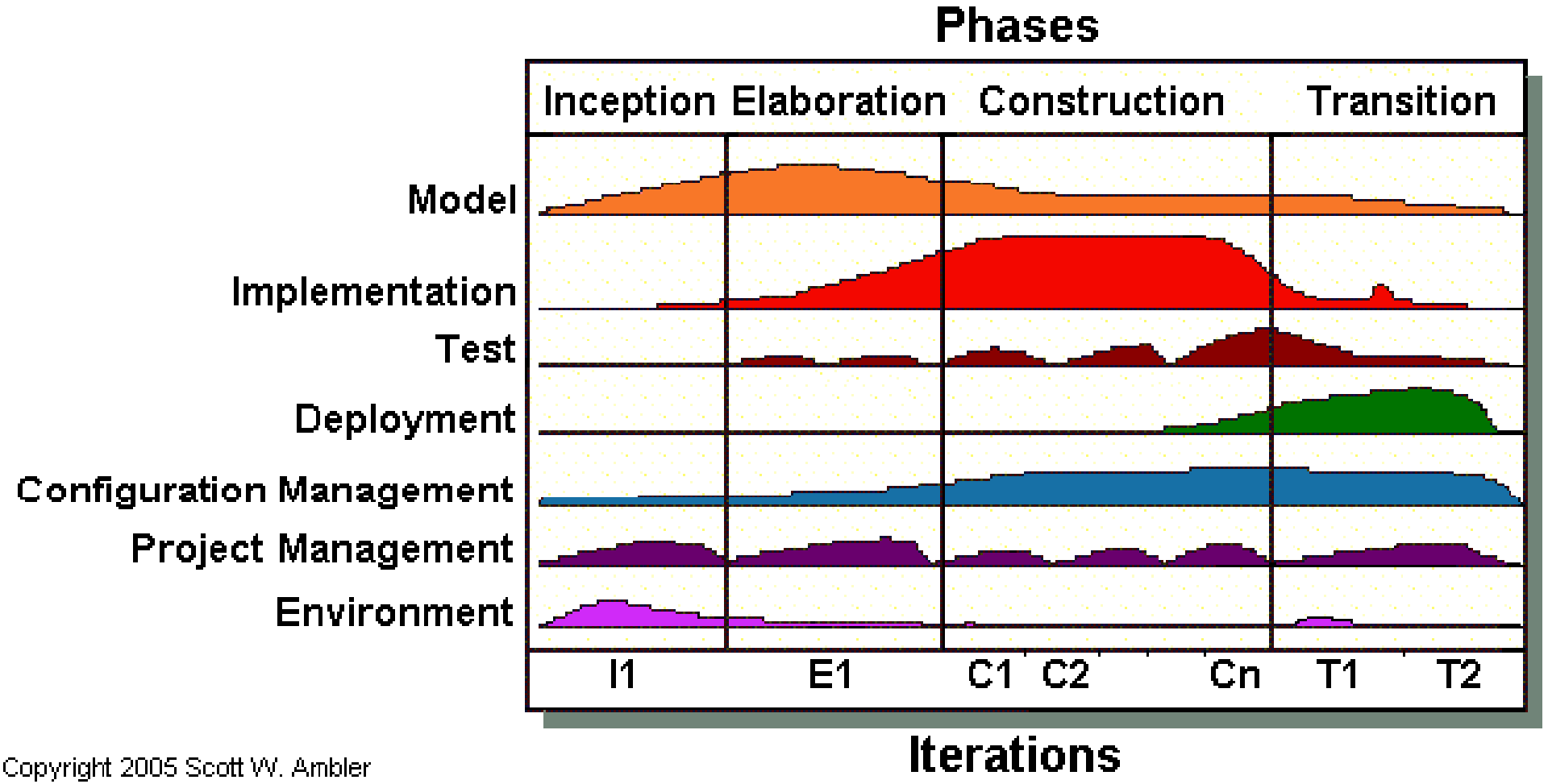
AUP: Process – Disciplines (Contd.)

5. **Configuration Management**. The goal is to manage access to the project artifacts. This includes not only tracking artifact versions over time but also controlling and managing changes to them.

6. **Project Management**. The goal is to direct the activities that take place on the project. This includes managing risks, directing people, and coordinating with people and systems outside the scope of the project to be sure that it is delivered on time and within budget.

7. **Environment**. The goal is to support the rest of the effort by ensuring that the proper process, guidance (standards and guidelines), and tools (hardware, software, etc.) are available for the team as needed.

AUP: Process – Disciplines in Iterations and Phases



[Ambler 2006]

AUP: Process – *Inception* Phase

- Tasks include:
  1. Define project scope.
  2. Estimate cost and schedule.
  3. Define risks.
  4. Determine project feasibility.
  5. Prepare the project environment: reserving workspace for the team, requesting the people needed, obtaining the necessary hardware/software, and tailoring the AUP.

AUP: Process – *Elaboration* Phase

- Tasks include:
  1. Produce an architectural prototype for the system.
  2. Evolve the requirements model.
  3. Draft a coarse-grained project plan for the construction phase.
  4. Ensure that critical tools, processes, standards, and guidelines are put in place for the construction phase.
  5. Understand and eliminate high-priority risks of the project.

AUP: Process – *Construction* Phase

- Tasks include:
  1. Prioritize and understand the requirements.
  2. Model storm a solution.
  3. Coding and testing the software.
  4. Deploying early releases of the system to obtain user feedback.

AUP: Process – *Transition* Phase

- Tasks include:
  1. Test and validate the complete system.
  2. Integrate the system with existing systems.
  3. Convert legacy databases and systems to support the new release.
  4. Train the users of the new system.
  5. Deploy the new system into production.

AUP: Process – *Model* Discipline

- Encompasses RUP's Business Modeling, Requirements, and Analysis & Design disciplines.
- Agility is observed by creating models and documents which are just barely good enough, limiting them to an absolute minimum.
- Performed through “model storming sessions” which:
  - Involve a few people, usually just two or three, who discuss an issue while sketching on paper or a whiteboard.
  - Typically last for five to ten minutes (it’s rare to model storm for more than thirty minutes).
  - The people get together, gather around a shared modeling tool (e.g. the whiteboard), explore the issue until they're satisfied that they understand it, then they continue on (often coding).

AUP: Strengths and Weaknesses

- ***Strengths***
  - Iterative-incremental process
  - Based on system architecture
  - Based on structural, functional and behavioral modeling (logical and physical) of the problem domain and the system
  - Based on system functionality captured in use cases
  - Traceability to requirements through the use of use cases throughout the process as the basis for tasks and tests

AUP: Strengths and Weaknesses

- ***Strengths*** **(Contd.)**
  - Design-based development
  - Iterative development engine governed by planning/reviewing
  - Seamlessness observed (though limited) due to use-case based activities and design-based development
  - Risk-based process
  - Formal features can be added via UML/OCL
  - Configurability and flexibility addressed

AUP: Strengths and Weaknesses

- ***Weaknesses***
  - Modeling may jeopardize agility if limits are not strictly observed; the list of AUP's models that are produced as the absolute minimum is actually quite long.
  - Tackling model inconsistency is not explicitly addressed.

Reference

- Ambler, S. W., “The Agile Unified Process (AUP)”, Ambysoft Corporation, 2006, Available on the Web at: http://www.ambysoft.com/unifiedprocess/agileUP.html.