Object-Oriented Design

Lecturer: Raman Ramsin

Lecture 2: USDP Overview

Software Development Methodology (SDM)

- A framework for applying software engineering practices with the specific aim of providing the necessary means for developing software-intensive systems

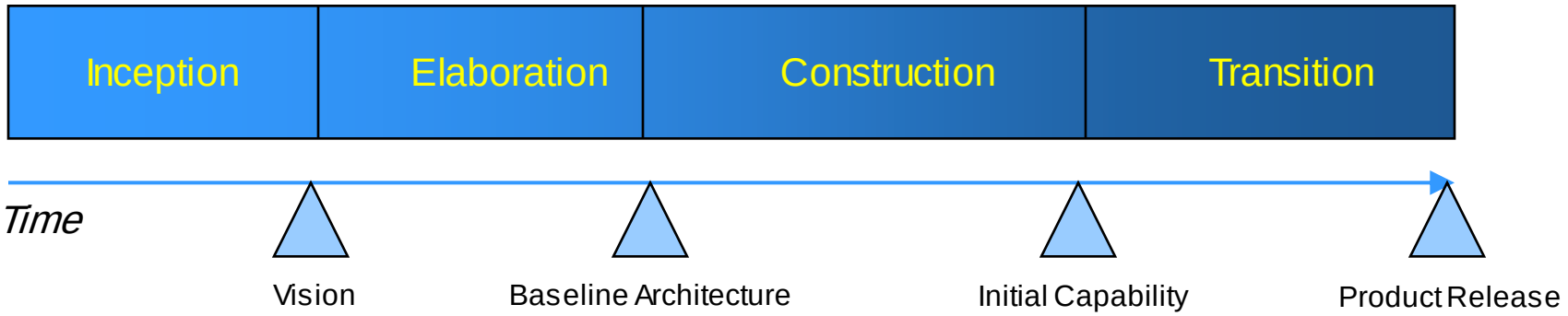
- Consisting of two main parts:
  - A set of modeling conventions comprising a *Modeling Language* (syntax and semantics)
  - A *Process*, which
    - provides guidance as to the order of the activities,
    - specifies what artifacts should be developed using the *Modeling Language*,
    - directs the tasks of individual developers and the team as a whole, and
    - offers criteria for monitoring and measuring a project's products and activities

Unified Software Development Process (USDP)

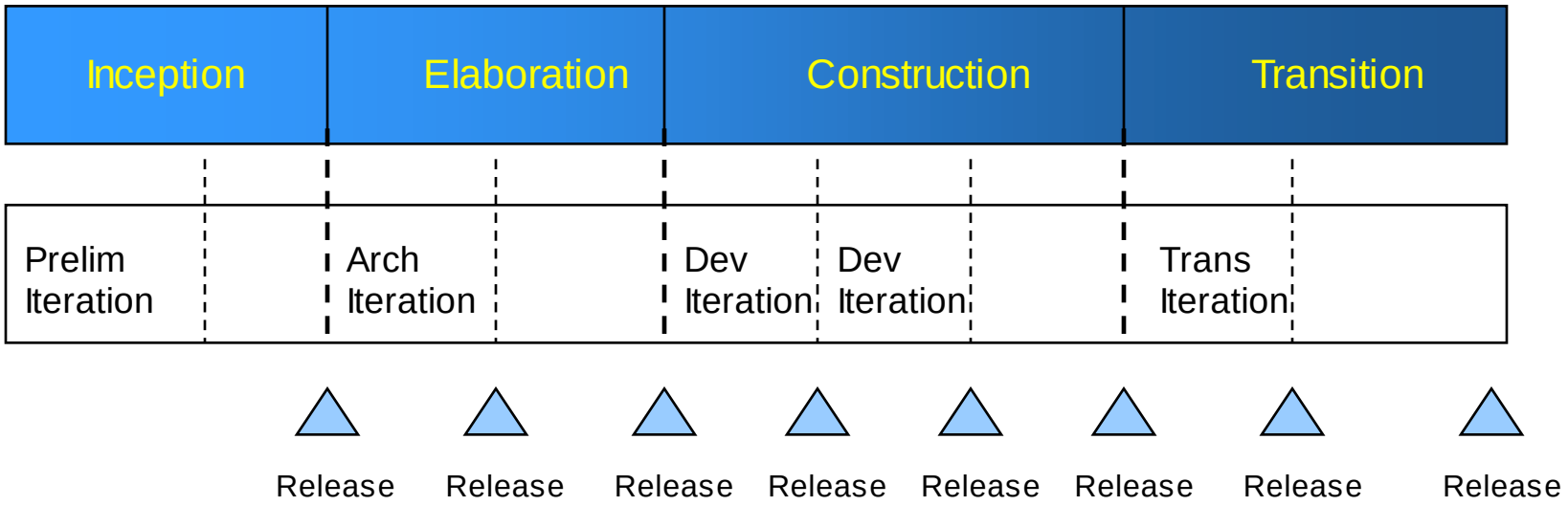
- Also known as Unified Process (UP)
- First introduced in 1999
- A refined, simplified, and non-proprietary version of the Rational Unified Process (RUP)
- UML-Based
- Use-Case-Driven
- Architecture-centric
- Iterative and Incremental

Unified Software Development Process

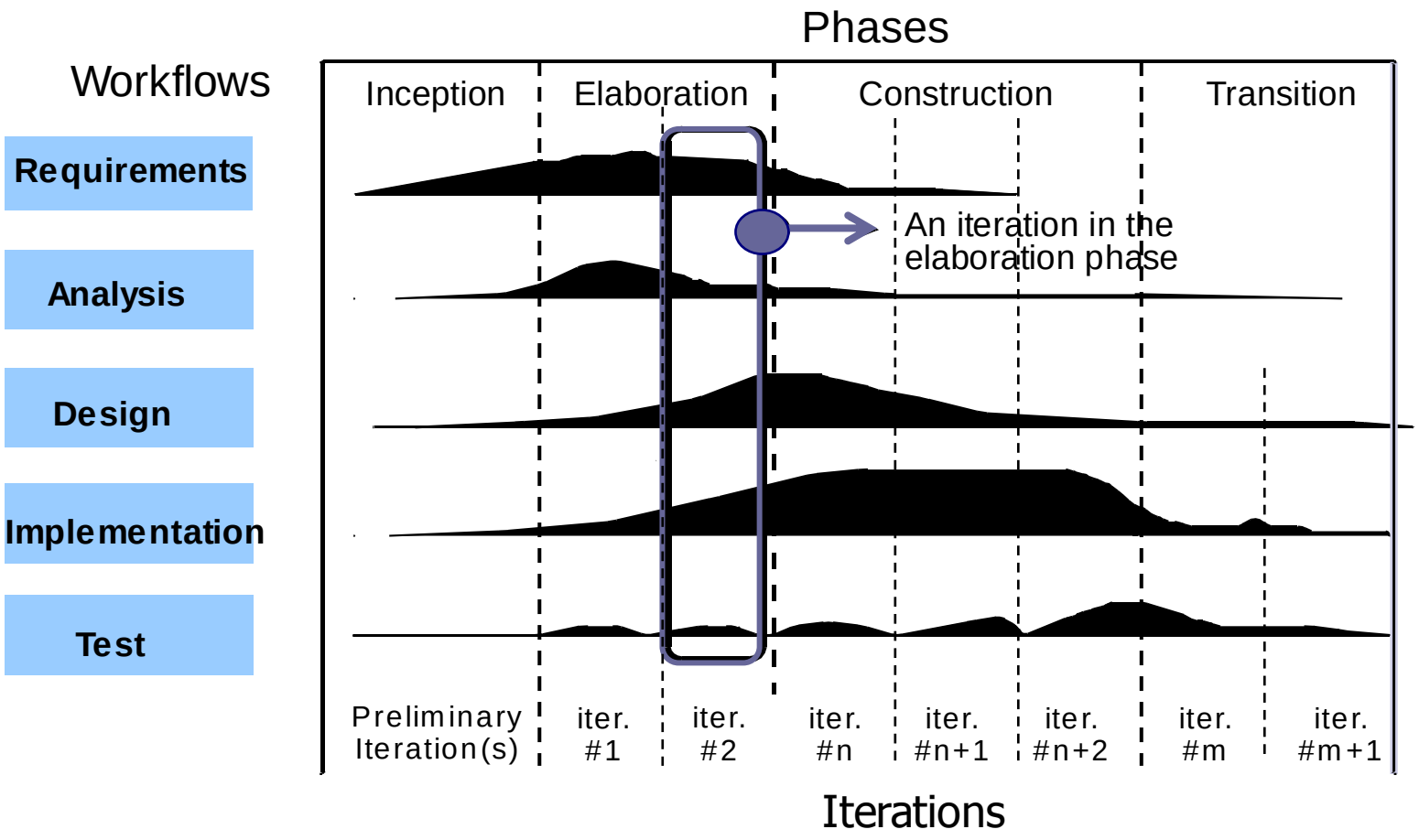
- Software lifecycle is decomposed over time in four sequential phases
  - **Inception (Vision Milestone)**
    - Define the vision of the product, scope of the project and the business case
  - **Elaboration (Architecture Milestone)**
    - Refine the definition of the product
    - Define and baseline an architecture
    - Develop a more precise plan for its development and deployment
  - **Construction (Initial Operational Capability Milestone)**
    - Build the product to the point where it can be delivered to its end-users for the first time
  - **Transition (Product Release Milestone)**
    - Transition the product to the user community; this includes manufacturing, delivering, training, supporting and maintaining the product



Phases and Iterations



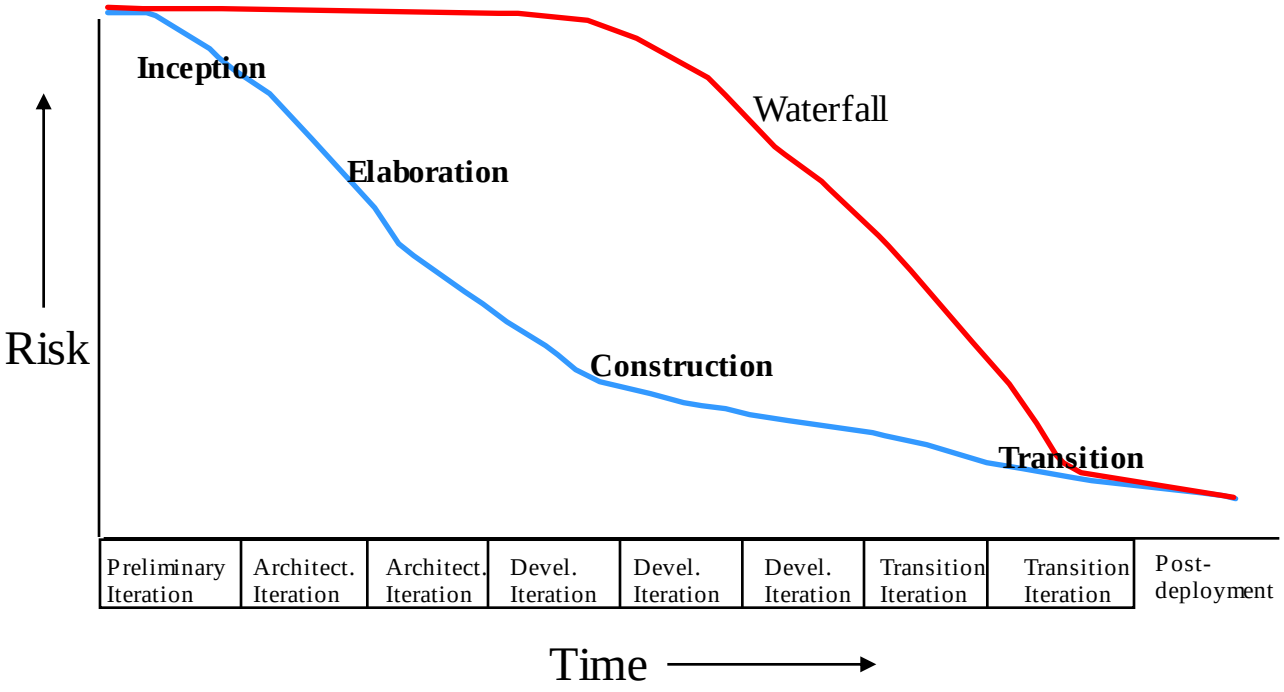
Iterations and Workflows



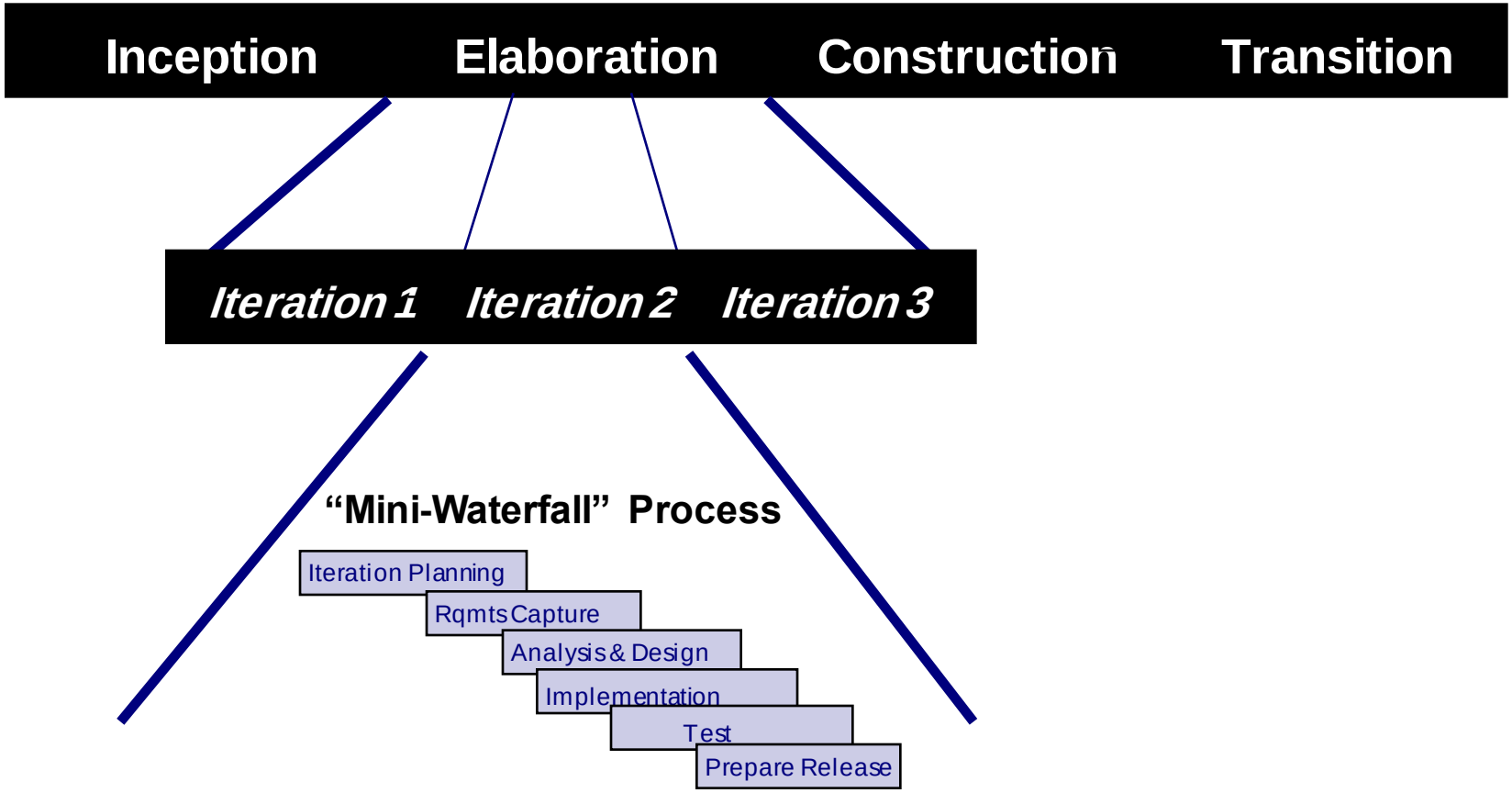
Features of the iterative approach

- Continuous integration
  - Not done in one lump near the delivery date
- Frequent, executable releases
  - Some internal; some delivered
- Attack risks through demonstrable progress
  - Progress measured in products, not documentation or engineering estimates

Risk Profile of an Iterative Development Process



Use Cases Drive the Iteration Process



Inception Phase

- The purpose of Inception is to "get the project off the ground":
  - establishing feasibility - this may involve some technical prototyping to validate technology decisions or proof of concept prototyping to validate business requirements;
  - creating a business case to demonstrate that the project will deliver quantifiable business benefit;
  - capturing essential requirements to help scope the system;
  - identifying critical risks.

Inception – Concerns

- The inception phase is a preparatory stage that attempts to answer the following questions:
  - What is the purpose and objectives of the project? Is it worth the effort?
  - Is the project feasible (e.g. technologically, financially, with current personnel)?
  - Should we buy the system, or build it?
  - Will it be developed now, or built from an existing system?
  - What are the estimated costs and risks?
  - Should we proceed with the project?
- This phase also deals with project planning and project management
  - This includes Gantt charts and plans, budgets, etc.

Inception – Postconditions and Deliverables

Conditions of satisfaction	Deliverable
The stakeholders have agreed on the project objectives	A vision document that states the project's main requirements, features, and constraints
System scope has been defined and agreed on with the stakeholders	An initial use case model (only about 10% to 20% complete)
Key requirements have been captured and agreed on with the stakeholders	A project glossary
Cost and schedule estimates have been agreed on with the stakeholders	An initial project plan
A business case has been raised by the project manager	A business case
The project manager has performed a risk assessment	A risk assessment document or database
Feasibility has been confirmed through technical studies and/or prototyping	One or more throwaway prototypes
An architecture has been outlined	An initial architecture document

Inception Timeline

- An important idea with Inception is that we do not yet know if a project will take place!
  - Often 1 or 2 iterations are required for Inception
- Therefore, since a project may be rejected, it makes sense that the Inception phase should be very short
  - Therefore, if the project gets scrapped, little time (and money) would have been wasted
  - It is not uncommon for Inception to last ***a few days*** to a few weeks, maximum

Elaboration Phase

- The purpose of Inception is to understand the problem, whereas Elaboration explores the solution:
  - create an executable architectural baseline;
  - refine the risk assessment;
  - define quality attributes (defect discovery rates, acceptable defect densities, and so on);
  - capture use cases to 80% of the functional requirements;
  - create a detailed plan for the construction phase;
  - formulate a bid that includes resources, time, equipment, staff, and cost.

Elaboration and the Workflows

- In the Elaboration phase, the focus in each of the core workflows is as follows:
  - requirements - refine system scope and requirements;
  - analysis - establish what to build;
  - design - create a stable architecture;
  - implementation - build the architectural baseline;
  - test - test the architectural baseline.

Elaboration - Concerns

- After Elaboration, project risks are essentially eliminated
  - The Architecture and UI have been approved by customers and managers
  - Technically difficult software components have been implemented, or proof-of-concept code has been created to prove it was possible
  - Cost estimates are finalized, so budgets can be approved
  - Preliminary user manuals have been created and analyzed
- Analysis, architecture and design well underway after Elaboration

Elaboration – Postconditions and Deliverables

Conditions of satisfaction	Deliverable
A resilient, robust executable architectural baseline has been created	The executable architectural baseline
The executable architectural baseline demonstrates that important risks have been identified and resolved	UML static model UML dynamic model UML use case model
The vision of the product has stabilized	Vision document
The risk assessment has been revised	Updated risk assessment
The business case has been revised and agreed with the stakeholders	Updated business case
A project plan has been created in sufficient detail to enable a realistic bid to be formulated for time, money, and resources in the next phases	Updated project plan
The stakeholders agree to the project plan	
The business case has been verified against the project plan	Business case
Agreement is reached with the stakeholders to continue the project	Sign-off document

Construction Phase

- The purpose of Construction is to iteratively enhance and evolve the previously created artefacts into the target system:
  - complete all requirements, analysis, and design
  - evolve the architectural baseline generated in Elaboration into the final system.

Construction and the Workflows

- We can summarize the kind of work undertaken in each workflow during Construction as follows:
  - requirements - uncover any requirements that had been missed;
  - analysis - finish the analysis model;
  - design - finish the design model;
  - implementation - build the Initial Operational Capability;
  - test - test the Initial Operational Capability.

Construction – Postconditions and Deliverables

Conditions of satisfaction	Deliverable
The software product is sufficiently stable and of sufficient quality to be deployed in the user community	The software product The UML model Test suite
The stakeholders have agreed and are ready for the transition of the software to their environment	User manuals Description of this release
The actual expenditures vs. the planned expenditures are acceptable	Project plan

Transition Phase

- The purpose of Transition is the ultimate deployment of the software produced at the end of Construction:
  - conduct beta test and acceptance test, and correct defects;
  - prepare the user sites for the new software;
  - tailor the software to operate at the user sites;
  - modify the software if unforeseen deployment problems arise;
  - create user manuals and other documentation;
  - provide user consultancy;
  - conduct a post-project review.

Transition and the Workflows

- We can summarize the kind of work undertaken in each workflow during Transition as follows:
  - Requirements - not applicable.
  - Analysis - not applicable.
  - Design - modify the design if problems emerge in testing.
  - Implementation - tailor the software for the user site and correct problems uncovered in testing.
  - Test - beta testing and acceptance testing at the user site.

Transition – Postconditions and Deliverables

Conditions of satisfaction	Deliverable
Beta testing is completed, necessary changes have been made, and the users agree that the system has been successfully deployed The user community is actively using the product	The software product
Product support strategies have been agreed on with the users and implemented	User support plan Updated user manuals

References

- Jacobson, I., Booch, G. and Rumbaugh, J. *The Unified Software Development Process*, Boston, MA: Addison-Wesley, 1999.

- Arlow, J., Neustadt, I., *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design*, 2nd Ed. Addison-Wesley, 2005.