

# Software Engineering

Prepared by: Neha Tripathi

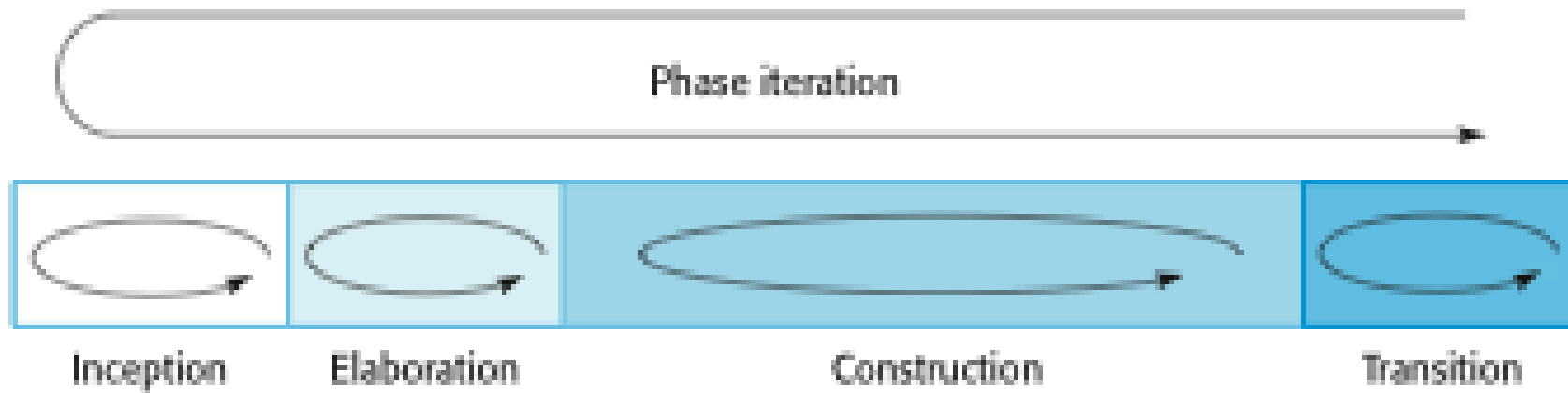
# **The Rational Unified Process (RUP) Model**

# RUP Model

- The **Rational Unified Process (RUP)** is an **iterative** software development framework.
- It was created by the Rational Software Corporation, a division of **IBM** since 2003.
- The Rational Unified Process is a modern generic process model that is organized into phases(inception, elaboration, construction, and transition) but separates activities (requirements, analysis, and design, etc.) from these phases.
- RUP incrementally grows an effective solution over multiple iterations
- A modern generic process derived from the work on the **UML** and associated process.
- **3 perspectives** of RUP Model
  - A **dynamic** perspective that shows phases over time;
  - A **static** perspective that shows process activities;
  - A **practice** perspective that suggests good practice.

- **Unified modeling language** :The RUP is a guide for how to effectively use the Unified Modeling Language(UML). The UML is a industry-standard language that allows us to clearly communicate requirements, architectures and designs.
- **Tools**: They are used to create and maintain visual modeling, programming, testing.
- **Configurable process**: The Unified Process fits small development teams as well as large development organizations.

# RUP Model representation



# Phases in the Rational Unified Process

- **Inception**
  - Establish the business case for the system.
- **Elaboration**
  - Develop an understanding of the problem domain and the system architecture.
- **Construction**
  - System design, programming and testing.
- **Transition**
  - Deploy the system in its operating environment.

# RUP Iterations

- **In-phase iteration**
  - Each phase is iterative with results developed incrementally.
- **Cross-phase iteration**
  - As shown by the loop in the RUP model, the whole set of phases may be enacted incrementally.

# Static workflows in the Rational Unified Process

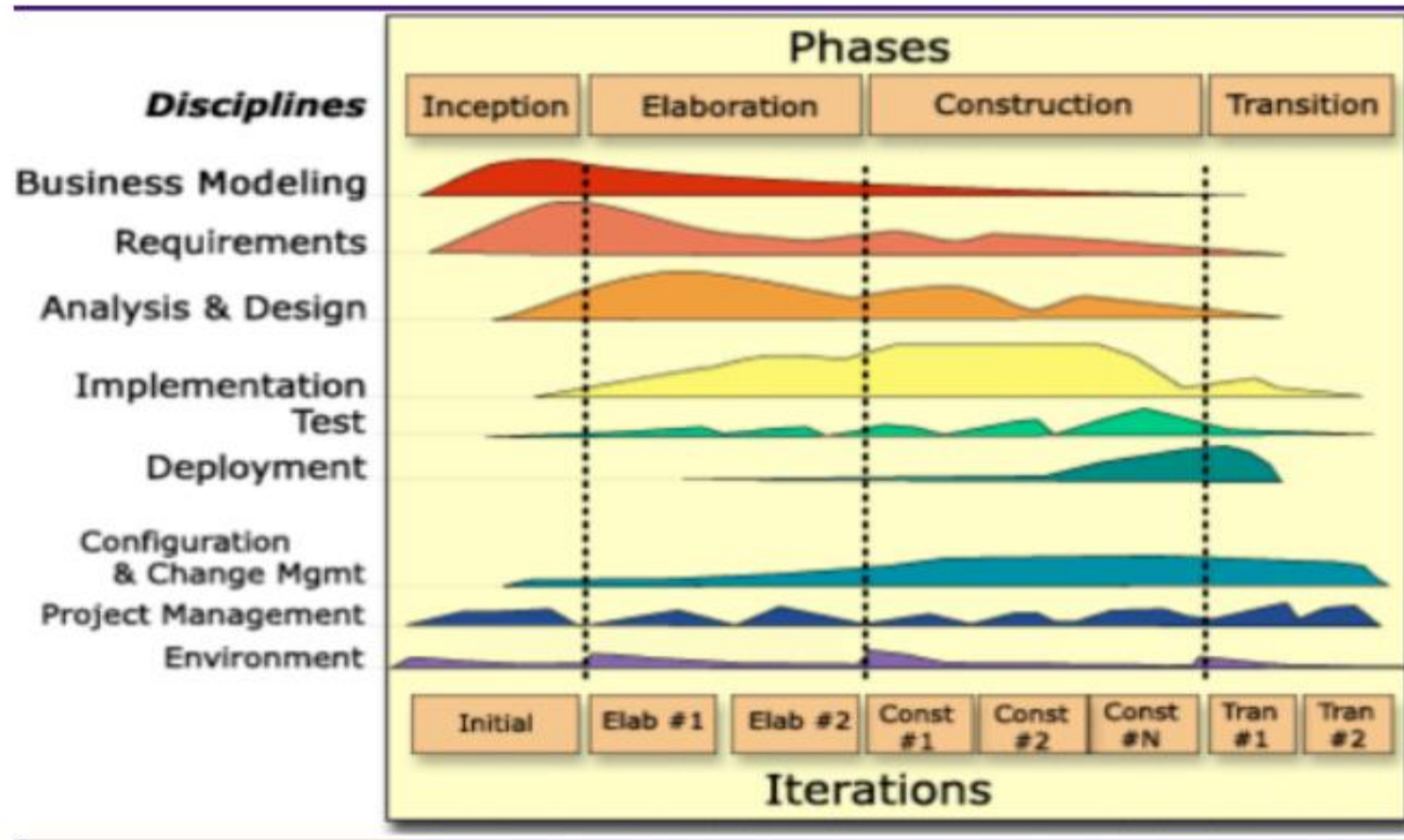
( 9 workflows= 6 core + 3 supporting)

Workflow	Description
Business modelling	The business processes are modelled using business use cases.
Requirements	Actors who interact with the system are identified and use cases are developed to model the system requirements.
Analysis and design	A design model is created and documented using architectural models, component models, object models and sequence models.
Implementation	The components in the system are implemented and structured into implementation sub-systems. Automatic code generation from design models helps accelerate this process.



Workflow	Description
Testing	Testing is an iterative process that is carried out in conjunction with implementation. System testing follows the completion of the implementation.
Deployment	A product release is created, distributed to users and installed in their workplace.
Configuration and change management	This supporting workflow managed changes to the system (see Chapter 25).
Project management	This supporting workflow manages the system development (see Chapters 22 and 23).
Environment	This workflow is concerned with making appropriate software tools available to the software development team.

# Effort Distribution of Activities



# RUP good practice

- Develop software iteratively
  - Plan increments based on customer priorities and deliver highest priority increments first.
- Manage requirements
  - Explicitly document customer requirements and keep track of changes to these requirements.
- Use component-based architectures
  - Organize the system architecture as a set of reusable components.
- Visually model software
  - Use graphical UML models to present static and dynamic views of the software.
- Verify software quality
  - Ensure that the software meet's organizational quality standards.
- Control changes to software
  - Manage software changes using a change management system and configuration management tools.

**Thank You!**