

An Introduction to the Rational Unified Process

68% of all Software Projects fail - ZDNET



- McKinsey – 17% of large IT Projects fail miserably
- Geneca - Large IT Projects run 45% over budget, 7% over time, delivering 56% less value
- 75% Project participants lack confidence in their project

Software Projects fail from around 5 to 47 factors – Alexandria University

- Organizational Structure
- Badly Defined Requirements
- Unrealistic or Unarticulated goals
- Inability to handle project complexity
- Sloppy development practices
- Inaccurate estimates



Project failures can be controlled



- Delivery dates impact project delivery
- Projects estimations can be made as close as possible
- Risks can be re-assessed, controlled and managed
- Staff can be awarded for long work hours

What is RUP ?

A software engineering process based on best practices in modern software development

- A disciplined approach to assigning and managing tasks and responsibilities in a developing organization
- Focus on high quality software that meets the needs of its end users within a predictable schedule and budget

A process framework that can be tailored to specific organization or project needs

RUP is a methodology for delivering projects in a maximum performance manner

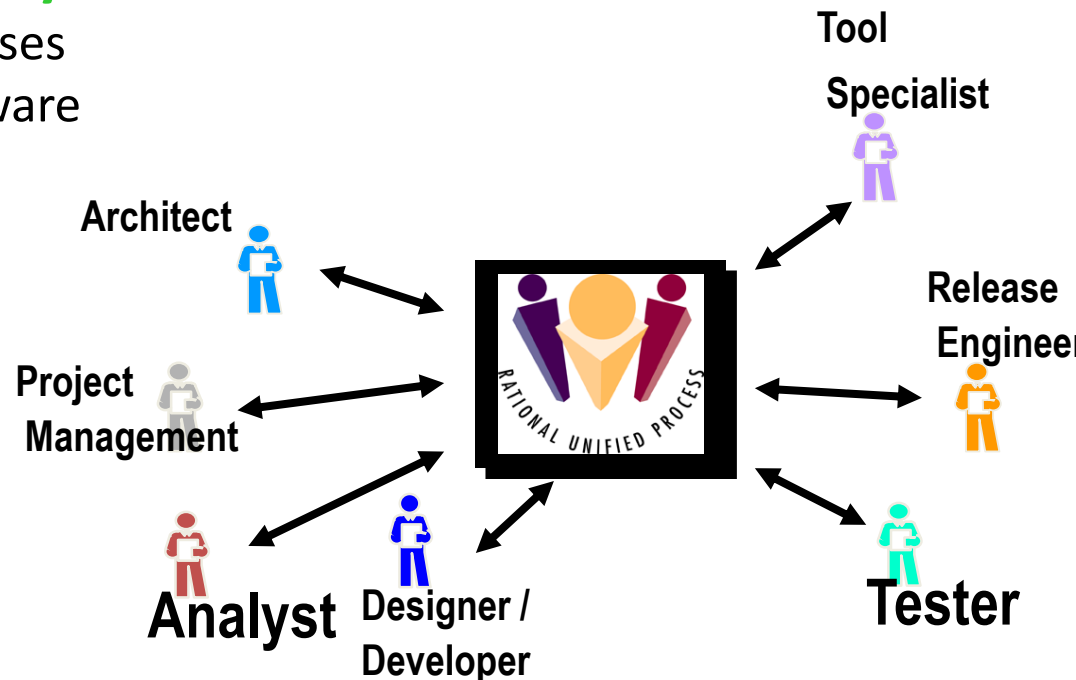
RUP uses an integration of approaches & initiatives

► Team-Unifying Approach

The RUP unifies a software team by providing a common view of the development process and a shared vision of a common goal

► Increased Team Productivity

- knowledge base of all processes
- view of how to develop software
- modeling language
- Rational provides many tools



Key Aspects of RUP

Risk-driven process

- Risk management integrated into the development process
- Iterations are planned based on high priority risks

Use-Case driven development

- Use cases express requirements on the system's functionality and model the business as context for the system
- Use cases are defined for the intended system and are used as the basis of the entire development process

Architecture-centric design

- Architecture is the primary artefact to conceptualize, construct, manage, and evolve the system
- Consists of multiple, coordinated views (or models) of the architecture

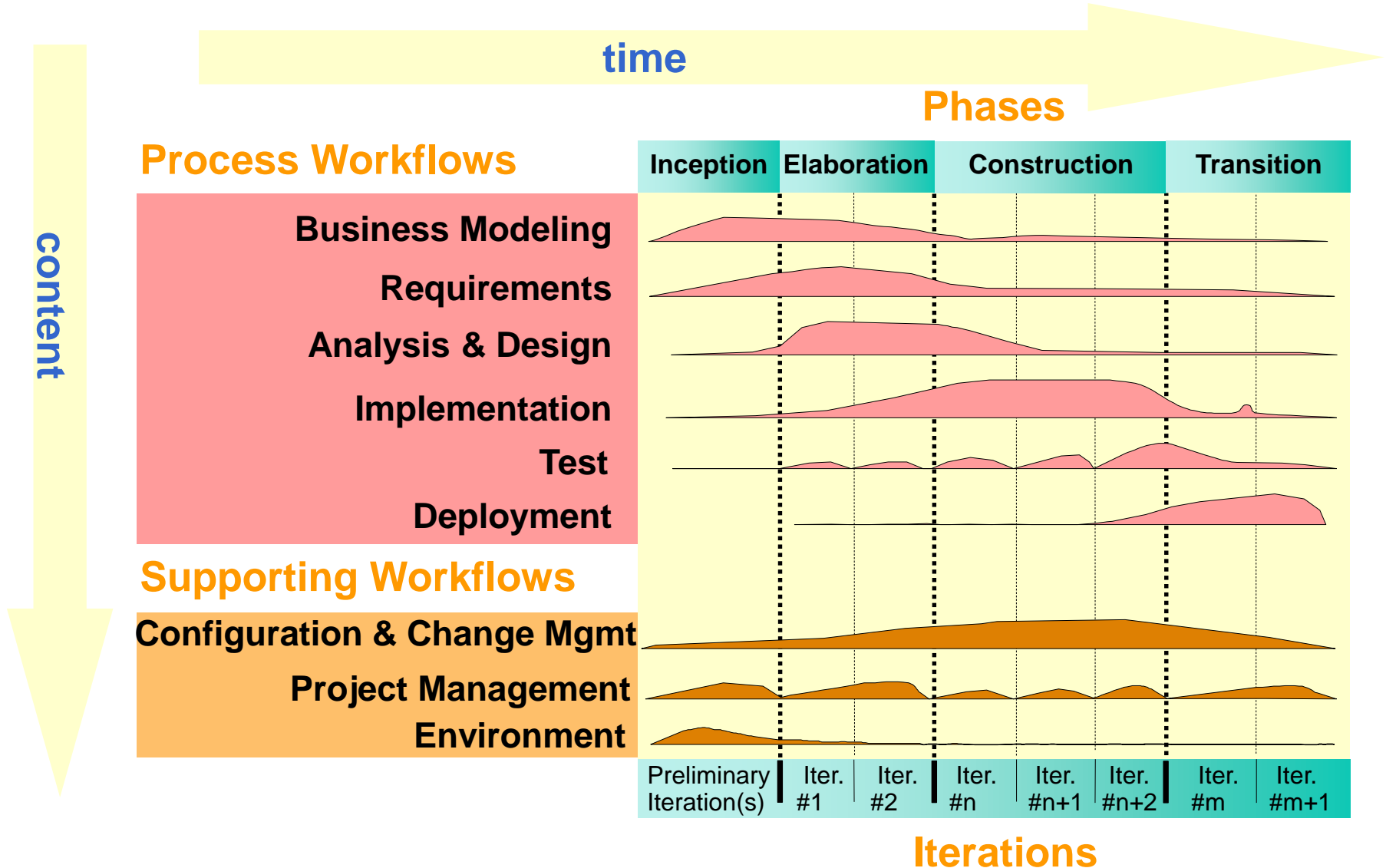
The Rational Unified Process

- RUP is a method of managing OO Software Development
- It can be viewed as a Software Development Framework which is extensible and features:
 - Iterative Development
 - Requirements Management
 - Component-Based Architectural Vision
 - Visual Modeling of Systems
 - Quality Management

The Development Phases

- Inception Phase
- Elaboration Phase
- Construction Phase
- Transition Phase

Rational Unified Process (RUP)



Inception Phase

The idea for the project is stated. The development team determines if the project is worth pursuing and what resources will be needed.

The outcome of the inception phase is:

- A vision document: a general vision of the core project's requirements, key features, and main constraints.
- A initial use-case model (10% -20%) complete).
- An initial project glossary (may optionally be partially expressed as a domain model).
- An initial business case, which includes business context, success criteria (revenue projection, market recognition, and so on), and financial forecast.
- An initial risk assessment.
- A project plan, showing phases and iterations.
- A business model, if necessary.
- One or several prototypes

- Understand what to build.
 - A vision document:
 - Optional business model
 - An initial project glossary
- Identify key system functionality.
 - A initial use-case model (10% -20%) complete.
- Determine at least one possible solution.
 - One or several prototypes.
- Understand the costs, schedule, and risks associated with the project.
 - An initial risk assessment.
 - Business case
- Decide what process to follow and what tools to use.
 - A project plan

Elaboration Phase

- Requirements Analysis and Capture(deeper)
 - Use Case Analysis
 - Use Case (80% written and reviewed by end of phase)
 - Use Case Model (80% done)
 - Scenarios
 - Sequence and Collaboration Diagrams
 - Class, Activity, Component, State Diagrams
 - Glossary (so users and developers can speak common vocabulary)
 - Domain Model
 - to understand the problem: the system's requirements as they exist within the context of the problem domain
 - Risk Assessment Plan revised
 - Architecture Document

The outcome of the elaboration phase is:

- A use-case model (at least 80% complete) — all use cases and actors have been identified, and most usecase descriptions have been developed.
- Supplementary requirements capturing the non functional requirements and any requirements that are not associated with a specific use case.
- A Software Architecture Description.
- An executable architectural prototype.
- A revised risk list and a revised business case.
- A development plan for the overall project, including the coarse-grained project plan, showing iterations” and evaluation criteria for each iteration.
- An updated development case specifying the process to be used.
- A preliminary user manual (optional

Construction Phase

- Focus is on implementation of the design:
 - cumulative increase in functionality
 - greater depth of implementation
 - greater stability begins to appear
 - implement all details, not only those of central architectural value
 - analysis continues, but design and coding predominate

The outcome of the construction phase is a product ready to put in hands of its end-users. At minimum, it consists of:

- The software product integrated on the adequate platforms.
- The user manuals.
- A description of the current release.

Transition Phase

- The transition phase consists of the transfer of the system to the user community
- It includes manufacturing, shipping, installation, training, technical support and maintenance
- Development team begins to shrink
- Control is moved to maintenance team
- Alpha, Beta, and final releases
- Software updates
- Integration with existing systems (legacy, existing versions, etc.)

Dynamic Elements Phases and Milestones

Major
Milestones



Lifecycle
Objectives

Lifecycle
Architecture

Initial
Operational
Capability

Product
Release
time

Inception

Elaboration

Construction

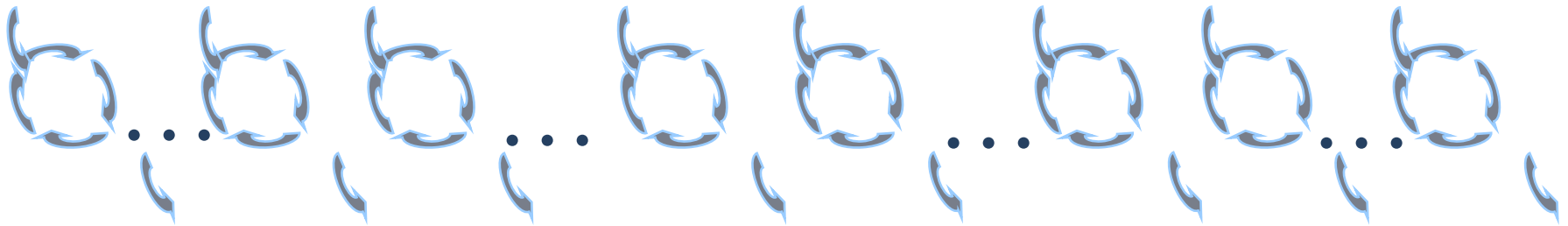
Transition

Define scope
of project

Plan project,
specify features,
baseline
architecture

Build product

Transition
product to
end user
community



Thank you