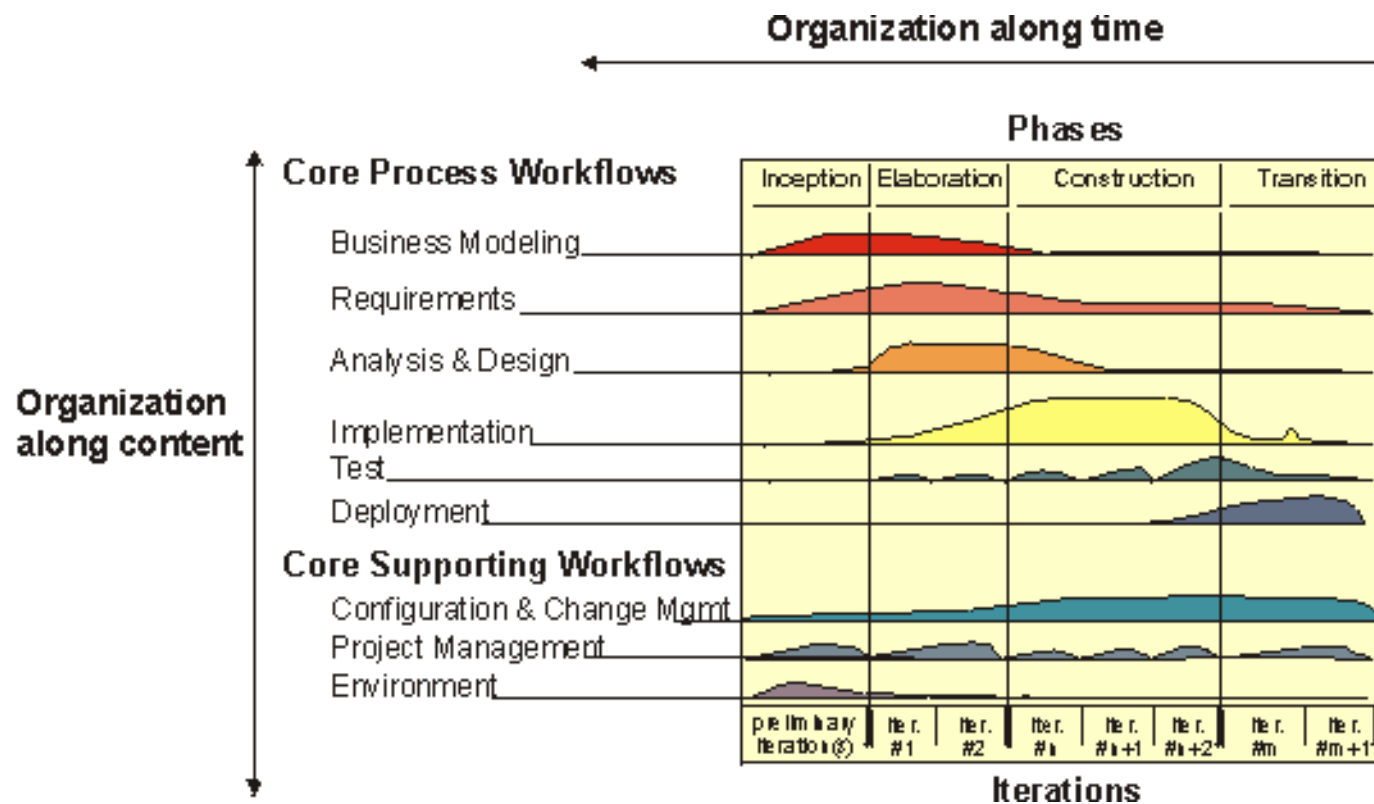


RUP *Rational Unified Process*

- **RUP is an iterative development process**
- **RUP is built on the "Six Best Practices"**
- **RUP has four phases:**
 - *Inception*
 - *Elaboration*
 - *Construction*
 - *Transition*
- **In each phase nine different workflows are active**

RUP structure

The picture shows the workload of each workflow during the project's phases



Why RUP

- **Risks**

- **Early and continuous documentation of the most urgent and the most probable risks.**
- **Planning**
- **Follow up**

- **UML**

- **(If used) A common modelling language**

- **Use cases**

- **May be used as test cases, end-user documentation and design description**

- **Glossary**

- **A clearly defined terminology makes everybody in a project aware of the exact meaning of a term. At every time.**

- **Interactive development**

- **By doing a piece of work at the time, review it and test its functionality. Actually saves time because early mistakes can be revealed**

- **Test**

- **Verify the requirements**

Best Practices

- **Develop Iteratively**
- **Manage Requirements**
- **Use Component Architecture**
- **Model Visually**
- **Verify Quality**
- **Control Changes**

Causes of failure

- **Ad hoc Requirement management**
- **Inprecise and non comprehensive communication**
- **Instable architecture**
- **Too complex**
- **Inconsistensis in requirement, design and implementation**
- **Insufficient tests**
- **Subjective status judgement**
- **Inablility to manage risks**
- **Uncontrolled changes**
- **Insufficient development automation**

Develop Interactively

- Circle picture goes here

Phases and iterations

The software lifecycle is broken into *cycles*, each cycle working on a new generation of the product. RUP divides one development cycle in four consecutive *phases*:

- **Inception phase**
- **Elaboration phase**
- **Construction phase**
- **Transition phase**

Each phase is concluded with a well-defined *milestone* - a point in time at which a certain critical decisions must be made, and therefore key goals must have been achieved.

Iteration

Each phase in RUP can be further broken down into iterations. An *iteration* is a complete development loop resulting in release (internal or external) of an executable product, a subset of the final product under development, which grows incrementally from iteration to become the final system.

Manage requirements

The secret of requirement management is to accept that the requirements change.

It is a continuous process to identify requirements together with the end-user

Requirement management is to:

- **Elicit, organize, and document required functionality and constraints,**
- **Evaluate impact of changes ,**
- **Track and document tradeoffs and decisions.**

Requirement management makes it possible to prioritise and track requirement.

Use Component Architecture

- **The process focuses on early development and baselining of a robust executable architecture, prior to committing resources for full-scale development.**
- **RUP supports *component-based software development***

Stable, independent modules with clear interfaces isolates software dependensis.

Model visually

Visual abstractions help you to

- **Communicate different aspects of your software;**
 - **See how the elements of the system fit together**
 - **Make sure the building blocks are consistent with your code**
 - **Maintain consistency between a design and its implementation**
 - **Promote unambiguous communication.**
-
- **Use cases makes minimal risk for**
 - **Hiding of details**
 - **No spaghetti structures**
 - **Good design gives quality**

Verify Quality

- **Early tests pay off thousandfold**
- **The software from each release (every iteration) is tested and verified**
- **Test cases are created from each use case and each use scenario.**

Decisions are made on real test results

High risk areas are tested more thoroughly

Control changes

All change control should go through CM, who should be able to foresee effects of changes and could be able to plan for them, and manage all *artifacts*

CM is convener for the CCB (Change Control Board).

CCB is the formal instance for decisions in change requests.

Members of the CCB can be representatives from the different workflows for example:

- architect ,
- test designer,
- project manager,
- system analyst,
- stakeholders
- user representant.

Formal change requests makes communication easier in the project.

Everybody must have access to all artifacts

Use Cases

- **Use cases are a kind of contract between the stakeholder and the developer**
- **Its important to use notions consistent, create a glossary for the whole project**
- **Use cases are important (if well written) in the test process.**

Development process

- **Control the order of the activities**
- **Define what artifacts that shall be creates**
- **Specify tasks at individual as well as group level**
- **Establish criterium to monitor and measure progress of the project**

Phases

- **A phase is concluded with a well defined milestone**
- **The Inception and the Elaboration phases are the two most creative parts. Function and design established all requirements are elicited.**
- **The Construction and the Transition phases are the building parts. Most of the programming and testing is done. The deployment and delivery.**

Inception phase

Inception means start.

The project are proposed.

- **The business case is established**
- **The scope of the project is delimited.**

To accomplish this you must identify all external entities with which the system will interact (*actors*) and define the nature of the interaction on a high-level. This involves identifying all use cases and describing a few significant ones. The business case includes success criteria, risk assessments, and estimate of the resources needed, and a phase plan showing dates of major milestones.

Inception outcome

- **Vision document**
 - **Initial use-case study (10%-20% complete)**
 - **Initial project glossary**
 - **Initial business case**
 - **Initial risk assessment**
 - **Project plan**
 - **Business model (if necessary)**
 - **One or several prototypes.**
-
- Stakeholders decide if or if not commence a full scale project.

Elaboration phase

Elaborate means refinement (careful development)

The purpose of the elaboration phase is to

- Analyze the problem domain,**
- Establish a sound architectural foundation**
- Develop the project plan**
- Eliminate the highest risk elements of the project.**

You should have a "mile wide and inch deep" view of the system.

Elaboration outcome

- **Use-case model 80% complete**
- **Supplementary requirements capturing the non functional requirements and requirements that are not associated with a specific use-case**
- **A Software Architecture Description**
- **An executable architectural prototype.**
- **A revised risk list and revised business case.**
- **A development plan for the whole 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.**
- **Preliminary user manual (optional).**

Construction phase

Construction means to build

During the construction phase all remaining components and application features are developed and integrated into the product, and all features are thoroughly tested.

Construction outcome

The outcome of the construction phase is a product ready to put in the hands of the end users.

At minimum, it consists of:

- **The software product integrated on the adequate platform**
- **The user manual**
- **A description of the current release**

This is considered as a "beta"-release

Transition phase

Transition means delivery

The purpose of the transition phase is to transition the software product to the user community.

Issues that usually arise:

- **New releases**
- **Correct some problems**
- **Finish the features that were postpone**

- **"beta-testing" to validate new system against user expectations**
- **The system might run in parallel with a legacy system that it is replacing**
- **conversion of operational databases**
- **training of user maintainers**
- **roll-out the product to marketing, distribution, and sales team**

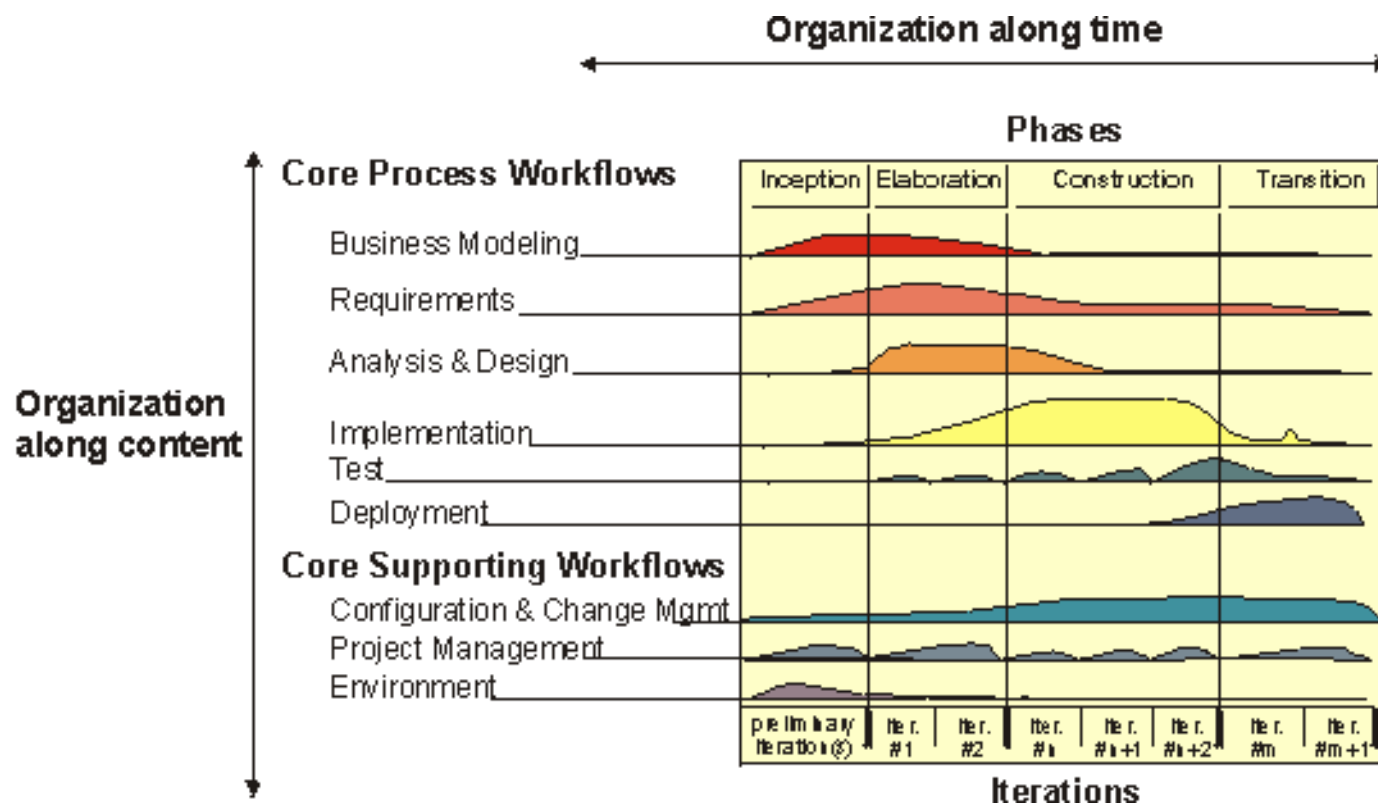
Transition achievements

The primary objectives of the transition phase include:

- **Achieving user self -supportability**
- **Achieving stakeholders concurrence that deployment baselines are complete and consistent with the evaluation criteria of the vision**
- **Achieving final produce baseline as rapidly and cost effective as practical**

Workflows

There are nine core process workflows in RUP, which represent a partitioning of all workers and activities into logical groups.



Business modelling

- **Business modelling are conducted during the inception and the elaboration phases.**

Active workers:

- **Business process analyst,**
- **Business Designer**
- **Business model reviewer**

In business modelling we document business processes using so called business use cases. This assures a common understanding among all stakeholders of what business process needs to be supported in the organization.

Requirements

- **The goal of the requirement workflow is to describe what the system should do and allows the developers and the customer to agree on that description. To achieve this we elicit organize and document required functionality and constraints; tracks and document tradeoffs and decisions.**

The system analyst:

- **Capture a common vocabulary**
- **Develop Requirements Management Plan**
- **Find Actors and Use-Cases**
- **Develop Vision**
- **Elicit Stakeholders Requests**

The software architect:

- **Prioritize Use-Cases**

The requirement reviewer:

- **Review Requirements**

Analysis and Design

The goal of the Analysis and Design workflow is to show *how* the system will be realized in the implementation phase.

Architect:

- **Architectural analysis**

Designer

- **Use-case analysis**
- **Use-case design**
- **Subsystem design**
- **Class design**

Implementation

- **The purposes of the implementation are to code and test the system**

Implementor

- **Implement a component**
- **Fix a defect**
- **perform unit test**

Integrator:

- **Plan system integration**
- **Plan subsystem integration**
- **Integrate subsystem**
- **Integrate system**

Code reviewer

- **Review code**

Test

In the test workflow test cases procedures and other verification methods are created and described.

Test designer:

- **Plan test**
- **Design test**
- **Implement test**
- -
- **Evaluate test**

Tester

- **Execute test**

Designer

- **Design test classes and packages**

Deployment

To deploy is to pack distribute and install the software at the end-user

Deployment manager:

- **Develop deployment plan**
- **Manage acceptance test**
- **Define bill of materials**
- **Write release notes**

Technical writer

- **Develop support materials**

CM

Configuration and Change Management (CM), monitors and administrates changes in the projects artifacts so that they are consistent with the requirement.

CM is responsible for release and version control and are convener for the CCB (Change Control Board)

Configuration Manager:

- **Set up CM environment**
- **Establish CM policies**
- **Write CM plan**

Change Manager:

- **Establish Change Control Process**
- **Review change request**

Project Management

Software project management is the art of balancing competing objectives, managing risk, and overcoming constraints to deliver, successfully, a product which meets the needs of both customers and users.

Project manager

- **Initiate project**
- **Develop iteration plan**
- **Develop quality assurance plan**
- **Develop product acceptance plan**
- **Monitor project status**
- **Schedule and assign work**
- **Report status**
- **Handle exceptions and problems**

Project Reviewer:

- **Project approval review**
- **Project planning review**
- **Iteration plan review**
- **Iteration Evaluation Criteria Review**

Environment

The purpose of the environment workflow is to provide the software development organization with the software development environment - both processes and tools - that are needed to support the development team.

Process Engineer

- **Development case**
- **Project specific templates**

Software architect:

- **Design guidelines**
- **Programming guidelines**

Tool Specialist:

- **Tool guidelines**
- **Tools**

... Lot of more guidelines