



## **Project Lifecycle**

**Version 1.0**



## Table of Contents

1.1. Purpose .....	3
1.2. Definitions, Acronyms, and Abbreviations.....	3
1.3. Overview.....	4
2.1. Strategy .....	5
2.2. Project Peculiarities Reflected in the Process of Development.....	6
2.3. Release.....	6
2.4. Project Team .....	9
2.5. Project Repository .....	11
2.5.1. Rational Project .....	11
2.5.2. Change Version Control .....	13
2.5.3. Documentation .....	13
2.5.4. Project Site .....	14
3.1. Technical Issues .....	15
3.2. Organizational Issues .....	15

## 1. Introduction

### 1.1. Purpose

The purpose of this document is to describe the process of software development fulfilled by Inreco LAN. The document is meant for internal use of InrecoLAN staff and for the Customer to get acquainted with the way the work is organized and performed. The technological process of development is based on the Rational Unified Process.

### 1.2. Definitions, Acronyms, and Abbreviations

- **Actor** - Someone or something, outside the system that interacts with the system.
- **Artifact** - a piece of information that is produced, modified, or used by a process
- **Defect** - An anomaly, or flaw, in a delivered work product. Examples include such things as omissions and imperfections found during early lifecycle phases and symptoms of faults contained in software sufficiently mature for test or operation. A defect can be any kind of issue you want tracked and resolved.
- **Enhancement Request** - A type of stakeholder request that specifies a new feature or functionality of the system
- **Iteration** - A distinct sequence of activities with a base-lined plan and valuation criteria resulting in a release.
- **Phase** - The time between two major project milestones, during which a well-defined set of objectives is met, artifacts are completed, and decisions are made to move or not move into the next phase.
- **Project** - Projects are performed by people, constrained by limited resources, and planned, executed, and controlled. A project is a temporary endeavor undertaken to create a unique product or service. Temporary means that every project has a definite beginning and a definite ending. Unique means that the product or service is different in some distinguishing way from all similar products and services. Projects are often critical components of the performing organizations' business strategy.
- **Role** - a definition of the behavior and responsibilities of an individual, or a set of individuals working together as a team, within the context of a software engineering organization
- **Repository** - A storage place for object models, interfaces, and implementations.
- **Requirement** - A requirement describes a condition or capability to which a system must conform; either derived directly from user needs, or stated in a contract, standard, specification, or other formally imposed document.
- **RUP** – Rational Unified Process

- **Use-case** - A description of system behavior, in terms of sequences of actions. A use case should yield an observable result of value to an actor. A use case contains all alternate flows of events related to producing the "observable result of value".
- **Vision** - The user's or customer's view of the product to be developed, specified at the level of key stakeholder needs and features of the system.
- **Workflow** - the sequence of activities performed in a business that produces a result of observable value to an individual actor of the business.
- **Production Site** – A release version of the system. It is meant to be used by the clients.
- **Mirror** – version of the system identical to the Production Site. It is meant for development and testing.

All terms in this document are used according to the Rational Unified Process Glossary.

### 1.3. Overview

The document contains general information on how the work on a project is organized.

Rational Rose diagrams used in the document as illustrations do not necessarily correspond to the actual state of the system and cannot be used outside the document

All artifacts produced within the project are considered property of the Customer of the specific project.

Rational Unified Process, Rational RequisitePro, RationalClearQuest, Rational Test Manager, Rational Rose, Rational ClearCase, Rational XDE, Rational SoDA, Rational Suite are registered trademarks of Rational Software Corporation.  
Microsoft Project, Microsoft Project Central, Microsoft Word, Microsoft .NET, Microsoft Visual Source Safe, Microsoft Windows NT, Microsoft Visual Studio are registered trademarks of Microsoft Corporation  
Visual Café is a registered trademark of WebGain

## 2. Project

### 2.1. Strategy

Nowadays the process of software development should be directed towards:

- creating high-quality software within the specified schedule;
- keeping within the project budget;
- quality control at each stage;
- meeting the customer's requirements for the software product.

During the process of high-quality software development a great number of miscellaneous activities are performed. Many important issues should be taken into consideration. In order to organize and manage all the necessary activities, one should use formal rules and organizational techniques. They change the entirely creative (and thus unpredictable) process of software development into a high-tech process of production. Using such techniques and methods enables the developers to successfully achieve the goals identified above. The complete product is deployed at the scheduled time, within the planned budget; it meets all the Customer's requirements and has a minimum of undetected errors. This means that the risks are reduced both for the Customer and for the Developer.

There exist a large number of techniques of software project management. They differ in the degree of formalization of processes, and each of them is aimed at specific groups of projects. For the management of medium-size and large projects we use the strategy of the Rational Software company – the Rational Unified Process (RUP). This strategy allows formalizing all the processes and their interrelations, and at the same time it allows the developers to be rather flexible and to treat each project individually depending on its peculiarities.

The RUP is based on the spiral model of development. It means that every now and then in the process of development the quality is assessed and the task is defined more precisely. This approach allows to quickly find and solve the problems that might appear and to involve the specialists of the Customer into the discussion of intermediate results and clarification of requirements, which in the long run ensures that the system as a whole completely meets Customer's needs.

One of the main advantages of the RUP is that this strategy is backed by a set of software tools called Rational Suite. It includes such tools as RegquisitePro for requirements management, ClearQuest for change management, Test Manager for test design and testing, Rose for building a model of the system, and many others. Rational Suite is one of the few packages that cover the whole lifecycle of a software project.

The RUP has a large library of artifacts and roles, a variety of templates for different documents. For each type of project a set of the artifacts to be developed, the degree of formalization of different processes should be defined and the roles of project participants specified. Though the RUP is a set of formal procedures, it is the developer who decides how and to what extent these procedures should be used for managing the specific project. Essentially, ways of using the RUP in different projects are a know-how of the development company. The following sections of the document will describe the interpretation and use of the RUP in Inreco LAN's projects.

## **2.2. Project Peculiarities Reflected in the Process of Development**

Frequently we deal with non-typical projects. Standard approaches described in textbooks cannot be applied to them. Below are the main peculiarities of projects taken into consideration:

- The task often consists of two parts: to modify the existing system and to create new sub-systems within the existing one;
- When the project is launched, there is practically no project documentation for the system. That is why the process of reengineering is of great importance;
- The active participation of the specialists of the Customer in the work is needed;
- Limited budget and strict deadlines. As a result, there is a need to use the available resources effectively and to plan the work very carefully.

When working on the project, the group of developers makes up one team with the specialists of the Customer, and this approach enhances team productivity.

## **2.3. Release**

Within large tasks, a "project" is a release, i.e. a version of the system that contains a set of new and modified features. All the workflows, phases, tasks mentioned below are performed within the framework of a single release. The average development time for a Release is about three months. The peculiarity of the project shows itself in the parallel development of several releases. As a result new versions of the system appear every month or a month and a half. The frequent updating of the system makes the development dynamic, allows to get efficient feedback on the changes made and to correct the coming releases depending on the results of the ones already issued. Such intensive development rhythm is possible due to the optimal organization of work.

Having analyzed the Rational strategy, we interpreted and adjusted it for the needs of our projects. The main principle is to single out a set of activities which are necessary

and at the same time sufficient for the development of a high quality product at the specified time.

The RUP defines the following workflows within a project:

- Business Modeling
- Requirements
- Analysis and Design
- Implementation
- Test
- Deployment
- Configuration and Change Management
- Project Management
- Environment

All the workflows are connected with one another. They are not performed successively but are carried out parallel to each other throughout the entire lifecycle of the project.

The main workflows used in our projects have a number of peculiarities:

- The Business Modeling workflow is performed entirely on the Customer's side. Based on the users' suggestions and on the analysis of the previous experience of working with the system, changes in the business model of the system are made.
- The Analysis & Design and Implementation workflows are closely interwoven as the same team members are involved in them. This is brought about by the strict limitations to the size of the project team and is meant to provide highly efficient use of the existing resources.
- The Deployment workflow is performed by the specialists of the Customer and developers together.

In the spiral model of development the project is conventionally broken into several phases. These phases are essentially the loops of the spiral. The RUP has the following set of phases :

- Inception
- Elaboration
- Construction
- Transition

One or more iterations may be performed in each of the phases.

As the lifecycle of a Release is rather short and all the team members are in the same room together and are always in touch with each other, the division into iterations is not

as accurate as prescribed by the RUP. Thus, we do not have Iteration Plans, they are largely covered by the Project Plan.

- The Inception phase has one iteration. At this stage together with the specialists of the Customer a set of Enhancement Requests is identified and clarified. Requirements begin to be defined. Interaction with the specialists of the Customer is very important for the adjustment of Requirements.
- During the Elaboration phase the system is designed. This phase has two iterations. In the first iteration the main workflow is Requirements. In this iteration a set of requirements to the system is formed and the functioning of the new and modified features is described. In the second iteration, the main workflow is Analysis and Design in which the mechanisms for the implementation of the specified Requirements are designed.
- The Construction phase includes implementation of changes and design of test procedures. The result is the “beta” Release. Depending on the complexity of the task this phase may have several iterations.
- The Transition phase includes testing, defects tracking and control and also activities of the Deployment workflow. An important milestone here is the issuing of the release, but the work keeps going till the successful operational usage of new version of the system begins. We consider it to be our main goal to develop high quality software that would ensure successful commercial usage. The phase has two iterations. The main workflow of the first iteration is Test and Defects tracking. The second iteration, short in time, deals with Deployment.

It should be pointed out that the workflows are not limited to the phases and iterations where they are considered the main. All the workflows influence one another. Thus, for example, the Requirements workflow may be performed in the Construction and even in the Transition phase if the need for clarification or modification of the Requirements appears in the course of Analysis or Implementation.

The Gantt chart in Fig.1 shows the groups of activities to be carried out for the development of a release.

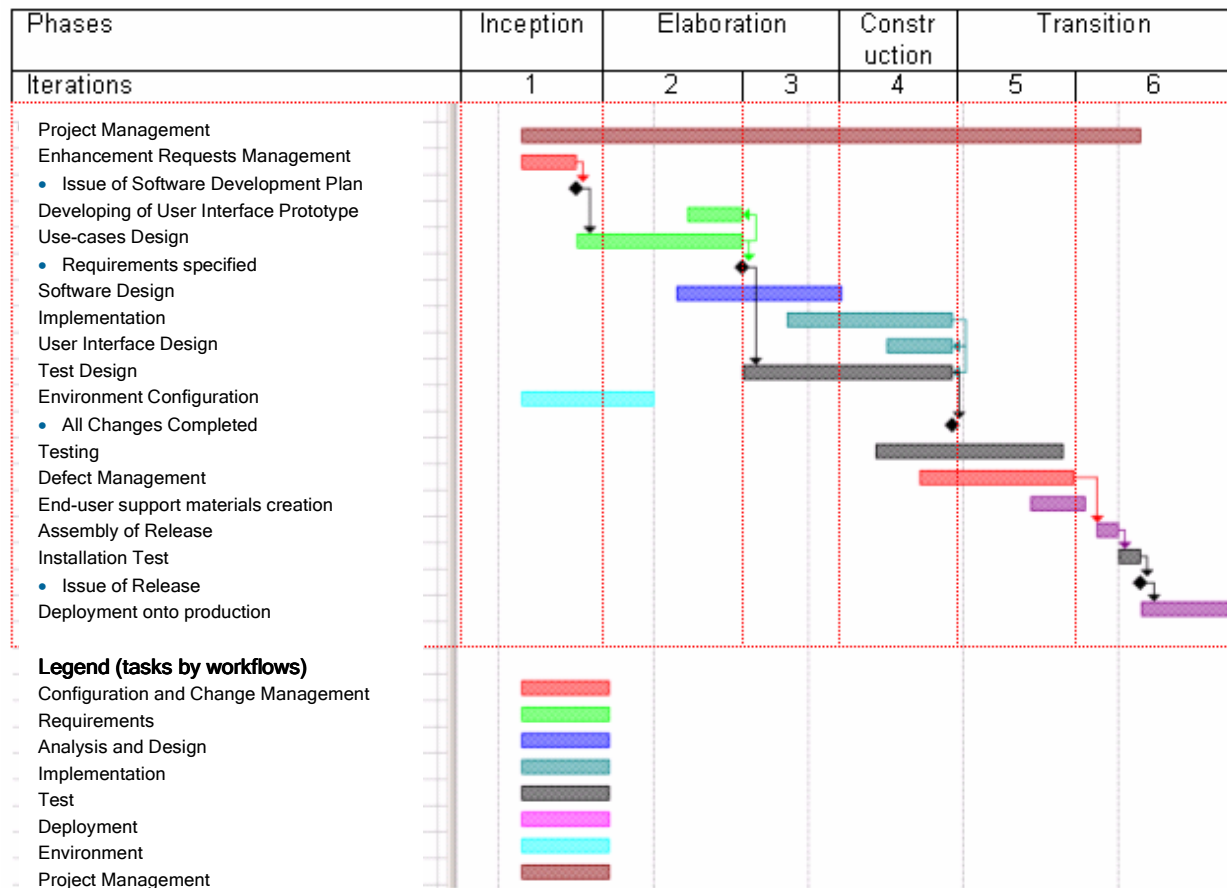


Fig.1 Main groups of activities.

Actually the chart shows only the main groups of activities. As stated above, any of the tasks performed may be continued in any phase. Defect management may serve as a good example here. Defects found during testing may refer to flaws in the test procedures themselves, to code errors, gaps in the specified requirements and even to inaccuracy of the task definition. Depending on the type of error changes will have to be made at the higher levels to fix it.

## 2.4. Project Team

The combination of tasks is the main criterion for forming the project team and assigning the functional roles to team members. The main goal is to form an optimum team for accomplishing the specified tasks.

Of all the many roles described in the RUP, we have selected a limited set of roles which embraces all the tasks set before us. Each of the roles is connected with several artifacts and thus the role owner is personally responsible for this or that part of the work. Table 1 contains a list of roles and artifacts used in the project.

Table 1. Roles and artifacts

Roles	Related artifacts
Project Manager	Software Development Plan Project Plan
Configuration Manager	Environment document set
Deployment Manager	Build Deployment Guide Migration Guide
Software Architect	Logical and Component Views of the Rose Model Use-case Realizations
Database Designer	Database Model
Implementer	Source Code
Business Designer	Glossary Use-case View of the Rose Model
Requirements Reviewer	Use-case Specifications
Test Analyst	Test Plan Test Results Report
Test Designer	Test Cases
Tester	Test Results
User Interface Designer	User interface prototype Software Design Specifications
Technical Writer	End-user support materials

The second principle for building the team is staying within the planned project budget. For this purpose each of the project participants performs several similar functional roles (for example, Test Designer and Tester). Besides, some of the roles that are not of great importance at the moment are distributed among the team members. For example, changes in the database are not significant right now and so in order not to involve another individual, the Database Designer role was assigned to the Project Manager who is qualified for that.

Besides, having analyzed the amount of work performed, we have divided the staff from the point of view of the degree of their employment. Most of the team members do not work full-time, some of them are not even on the staff and are involved only occasionally (User Interface Designer, Technical Writer, etc.)

The third principle of the organization of work is the hierarchical structure of the team. With the help of the distribution of functions and the subordination, all arising problems are solved quickly and effectively.

A project management team is assigned for each project. Together they solve the most difficult problems. Project management team consists of:

- The Project Manager who determines the main strategy and also deals with organizational issues;
- A project manager responsible for technical issues;
- An individual responsible for Analysis & Design and Implementation workflows;
- An individual responsible for the Requirements and Test workflows.

A manager of any level is responsible for all the work done in his area, each of the staff members regularly gives report to his manager of the work done. This helps to avoid contradictory decisions and confusion.

Another important principle of the organization of work is mutual control. The results of any work are used by some other team member at the next stage. For example, test procedures are designed by one tester, but they are performed by another. One team member builds the Release, somebody else performs its test installation. This approach does not let a team member get too deep into the task before him and allows to avoid such situations when something important seems obvious to the Developer and thus is left out of the documents and specifications.

The project team is dynamic. It is open to slight changes brought about by the peculiarities of the current tasks.

## **2.5. Project Repository**

The Repository centralizes and arranges the storage and usage of all the data on the project. The Repository allows project participants to access all the data they are interested in. Access to all the information is authorized. There is remote access via the Internet to most of the data, which is very important for interaction with the specialists of the Customer and for settling urgent matters when the Developer has to work late at night because of the time difference between the Customer and the Developer.

The repository has several parts which are described below.

### **2.5.1. Rational Project**

The main part of the Repository is Rational Project which integrates information from different Rational Suite tools.

Table 2. Rational Project contents

Software	Purpose	Contents
Rational RequisitePro	Requirements management	Glossary, Use-case specifications, Requirements
Rational ClearQuest	Change management	Enhancement requests, Founded defects
Rational Test Manager	Test design and testing	Test plans, Test cases, Test results
Rational Rose	System architecture design	Rose models

All data within Rational Project are interrelated. This makes it possible to follow the chain Enhancement request – Use-cases – Rose diagrams – Test – Defects, i.e. the full development cycle from the specification of the task to fixing the defects. This approach makes a complicated project manageable. On the other hand, it helps to organize “interfaces” between the team members involved in different tasks, to significantly reduce the amount of time needed for discussion and to encourage unambiguous communication

Rational Project stores history for each of the artifacts. This allows to reconstruct the work done on this or that task.

Access to all Rational tools is authorized and differentiated according to the functional roles of the staff members.

Most Rational tools have web-interface. This provides remote access to the necessary information. This possibility is invaluable in the interaction with the Customer.

Particular attention should be given to Rational Rose models. They constitute the core of the project, practically every team member works with them. The models contain the most complete information about the system in several views:

- Use-case view – level of design (Requirements);
- Logical View – level of development (Analysis and Design);
- Component View – level of implementation (Implementation);
- Deployment View – level of system deployment (Deployment View).

Rose Models are integrated with the main development tools – Visual Café and Visual Studio (via Rational XDE).

However, it should be pointed out that the use of Rational Project demands highly qualified staff and permanent maintenance. It may also lead to many technical difficulties. That is why it is not worthwhile using it full-scale in small projects.

### **2.5.2. *Change Version Control***

Version control system stores all the source codes and compiled classes, and also the main documents used in the project.

MS Visual Source Safe Net is used as the version control system in most of our projects. Its main advantages are the following. It is easy to use and may be integrated with the main development tools.

For complicated and long-term projects Rational ClearCase may be used. It is one of the powerful tools for change version control. Not only the source code is controlled but also all artifacts that make up Rational Project. Besides, several baselines may be defined, which allows parallel development of several versions of the system.

Access to the version control system is authorized and differentiated according to the functional roles of the staff members.

The Software Architect is responsible for using the version control system. The Deployment Manager is responsible for making changes in the Release and for placing them onto the mirror.

### **2.5.3. *Documentation***

In the process of development a large number of documents are created. Most of them are regularly modified. The project has a system of document circulation aimed at centralizing the process of creation, storage and distribution of the documents.

The documents are stored in the NT Sever file system and are arranged according to their type and version. Access to project documentation is authorized and differentiated according to the functional roles of the team members.

There is a unified template for all of the documents. This document may serve as an example.

The Project Manager responsible for technical issues is also responsible for the project documentation. He makes sure the documents appear at the specified time. When a new document is created or an existing one is modified, a version and code are assigned to it. The code contains abbreviations of the project and the name, version number and the code for the language of the document. Then the document is placed in the repository in two formats (doc and pdf). Also a hard copy of the document is made.

#### **2.5.4. Project Site**

The Project Site is the web-interface to the Repository. The site contains the latest versions of practically all documents on the project, links to web-interfaces of different Rational software tools and also to the Rose models. The main goal is to provide easy access to any of the project artifacts. The site is particularly important for remote work. Specialists of the Customer may access any document they are interested in at any time, they may analyze the current progress, make suggestions and adjustments. The Project Site and E-mail are the main ways of information exchange between the development team and the Customer. Unlike E-mail, the project site provides constant centralized access to information for all the users.

Access to the project site is authorized.

The Project Manager responsible for technical issues is also responsible for maintaining the project site. Project Site News is sent regularly via E-mail to all the team members and specialists of the Customer to notify them of the appearance of new documents and of changes in other artifacts.

## 3. Organizational and Technical Support of Development

### 3.1. Technical Issues

Technical issues are usually addressed at the initial stages of the project (the Inception phase). The infrastructure is created and expanded. Its configuration remains the same throughout the whole Release lifecycle.

Technical issues depend largely on the peculiarities of each individual project. There are however certain general requirements concerning the setup of the environment.

The standard task solved here is the setup of workplaces of analysts, developers and testers. The workplaces of the same functional area should be identical. The following software is installed:

- System software;
- The necessary utilities and development tools;
- Rational tools.

There are a number of instructions that regulate the configuration of the workplaces.

One more task is the maintenance of the project Repository (see section 2.5) that includes Rational Project, Control Version System, Documentation and Project Site. Technical aspects of managing the Repository are regulated by a number of internal company documents.

### 3.2. Organizational Issues

Organizational support includes maintenance of a set of rules that regulate the process of development in general. These tasks are closely connected with the Project Management workflow and are solved by the Project Managers.

The document Docflow Rules may serve as an example of an artifact produced in this workflow. It regulates document flow in the project. Some of its ideas are given in section 2.5.3.

The Master-Class of Software Technologies where the work on the projects is carried out is under the authority of both Inreco LAN and Vladimir State University. Thus, an important task of organizational support of development is our collaboration with Vladimir State University to provide adequate maintenance work, power supply issues, connections to ISPs, security services, etc.