Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 30/10/2018 | 1.0 | Requirement Management Plan | Quản Trọng Tú |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 Overview 4

2. Requirements Management 4

2.1 Organization, Responsibilities, and Interfaces 4

2.2 Tools, Environment, and Infrastructure 4

3. The Requirements Management Program 5

3.1 Requirements Identification 5

3.2 Traceability 5

3.3 Attributes 5

3.3.1 Status 5

3.3.2 Benefit 6

3.3.3 Effort 6

3.3.4 Risk 6

3.3.5 Stability 6

3.3.6 Target Release 6

3.3.7 Assigned to 7

3.3.8 Reason 7

3.4 Reports and Measures 7

3.5 Requirements Change Management 7

3.5.1 Change Request Processing and Approval 7

3.5.2 Project Baselines 7

3.6 Workflows and Activities 7

4. Milestones 8

5. Training and Resources 8

# Introduction

This Requirements Management Plan provides an overview of how the requirements of our application payroll management system are managed. It includes the purpose, scope, definitions.

## Purpose

This Requirements Management Plan is to describe and illustrate how the application payroll management system project will set up requirements documents and requirement types, and their respective requirement attributes and traceability

## Scope

This Requirements Management Plan aims to people who is planning to build up an application payroll management system.

## Definitions, Acronyms, and Abbreviations

* UML: Unified Modeling language (UML) is a standardized modeling language enabling developers to specify, visualize, construct and document artifacts of a software system. Thus, UML makes these artifacts scalable, secure and robust in execution. UML is an important aspect involved in object-oriented software development.
* **Baseline:** A baseline is a 'snapshot' in time of one version of each artifact. It provides an official standard on which subsequent work is to be based, and to which only authorized changes can be made.
* **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 formal agreement

## Overview

In this section, the Requirement Management Plan provide the information and induction to establish this application. It enables the reader to move on the section that is most essential to them.

**Requirements Management**: describes the requirement types, requirement attributes and the document type the project will maintain.

# Requirements Management

## Organization, Responsibilities, and Interfaces

In my group:

- Tu is responsible for Requirement Definition.

- Hiep, Toan are responsible for Prototyping.

- Hiep, Phuc is responsible for Business Modeling.

- Tuan Anh, Tu are responsible for Analysis & Design.

## Tools, Environment, and Infrastructure

This Project includes UML diagrams to provide descriptions of our application’s activities. The interface of application payroll management is design with some programming languages such as Java (Java Swing). More over, this project connect to mySQL for managing payroll easily.

The following tools will be used in the Requirements discipline:

| **Tool** | **Purpose** |
| --- | --- |
| Rational Software Modeler (RSM) | * Creates visual models of the requirements (actor and use case model diagrams) |

# The Requirements Management Program

This section describes the various requirement types. It describes the traceability between these requirement types, the attributes of the requirement.

## Requirements Identification

|  |  |  |
| --- | --- | --- |
| **Artifact**  **(Document Type)** | **Traceability Item** | **Description** |
| Stakeholder Requests (STR) | Stakeholder Request (STRQ) | Key requests, including Change Requests, from stakeholders |
| Vision (VIS) | Stakeholder Need (NEED) | Key stakeholder or user need |
| Vision (VIS) | Feature (FEAT) | Conditions or capabilities of this release of the system |
| Use-Case Model | Use Case (UC) | Use cases for this release, documented in Rational Rose |
| Supplementary Specification (SS) | Supplementary Requirement (SUPP) | Non-functional requirements that are not captured in the use-case model |

## Traceability

Traceability matrix and traceability graph can be made to enable users to find the origin of each requirement and track every change that was made to the requirement.

* Requirement Traceability Matrix (RTM): verify that all stated and derived requirements are associated with corresponding design elements, system components, modules and other project deliverables. The requirements traceability matrix is a tool to ensure that deliverables meet the requirements of the project.

## Attributes

### Status

|  |  |
| --- | --- |
| Proposed | Used to describe features that are under discussion but have not yet been reviewed and accepted by the "official channel", such as a working group consisting of representatives from the project team, product management, and user or customer community |
| Approved | Capabilities that are deemed useful and feasible, and have been approved for implementation by the official channel |
| Rejected | Rejected by the official channel |
| Incorporated | Features incorporated into the product baseline at a specific point in time. |

### **Benefit**

Because all requirements are not created equal, requirements management plan helps the entire project group to mange and rank what is the most requirement

|  |  |
| --- | --- |
| Critical | Essential features. Failure to implement means the system will not meet customer needs. All critical features must be implemented in the release or the schedule will slip. |
| Important | Features important to the effectiveness and efficiency of the system for most applications. The functionality cannot be easily provided in some other way. Lack of inclusion of an important feature may affect customer or user satisfaction, or even revenue, but release will not be delayed due to lack of any important feature. |
| Useful | Features that are useful in less typical applications or for which reasonably efficient workarounds can be achieved will be used less frequently. No significant revenue or customer satisfaction impact can be expected if such an item is not included in a release. |

### **Effort**

Set by the development team. Because some features require more time and resources than others, estimating the number of team or person-weeks, lines of code required or function points, for example, is the best way to gauge complexity and set expectations of what can and cannot be accomplished in a given time frame. Used in managing scope and determining development priority.

* High: Above average level of effort to complete.
* Average: Average level of effort to complete.
* Low: Below average level of effort to complete

### **Risk**

Set by the development team and based on the probability the project will experience undesirable events, such as cost overruns, schedule delays, or even cancellations. Most project managers find categorizing risks as high, medium, and low to be sufficient, although finer gradations are possible. Risk can often be assessed indirectly by measuring the uncertainty (range) of the projects’ team’s estimated schedule.

* High: The impact of the risk combined with the probability of the risk occurring is high. >75%
* Medium: The impact of the risk is less severe
* Low: The impact of the risk is minimal <25%.

### **Stability**

Set by the analyst and development team, this is based on the probability that the feature will change or the team’s understanding of the feature will change. Used to help establish development priorities and determine those items for which additional elicitation is the appropriate next action.

### **Target Release**

Records the intended product version in which the feature will first appear. This field can be used to allocate features from a **Vision** document into a particular baseline release. When combined with the status field, your team can propose, record, and discuss various features of the release without committing them to development. Only features whose Status is set to Incorporated and whose Target Release is defined will be implemented. When scope management occurs, the Target Release Version Number can be increased so the item will remain in the **Vision** document, but will be scheduled for a later release.

### **Assigned to**

In many projects, features will be assigned to "feature teams" responsible for further elicitation, writing the software requirements and implementation. This simple pull-down list will help everyone on the project team to better understand responsibilities.

### **Reason**

This text field is used to track the source of the requested feature. Requirements exist for specific reasons. This field records an explanation or a reference to an explanation. For example, the reference might be to a page and line number of a product requirement specification or to a minute marker on a video of an important customer interview

## Reports and Measures

All reports in this Requirement Definition are formatted as Rational Unified Process (RUP) specific documents.

**Size Metrics*:*** The following items will be analyzed and compared per iteration, per project and per month:

UCs, FEATs (that are functional requirement, non-functional quality requirements URPS, Interface requirements and DCs), STRQs, MECHs, Actors.

They will also be plotted versus total time to complete the iteration/project to look for any correlations. Separate Elaboration and Construction metrics will also be compared.

## Requirements Change Management

### Change Request Processing and Approval

The change request and approval will follow the standard change control procedures. The standard change control procedure in ECI case would be usage of project change control process. In this database, any ECI team member shall be able to submit change requests against ECI project requirement, defect against ECI project or any process improvement against RUP pilot or RUP adoption project.

### Project Baselines

* Project Baselines definition: the expected values or conditions against which all performances are compared
* Baselines provide an official standard on which subsequent work is based and to which only authorized changes are made.
* The baselining of requirements documents should be established at the end of Inception, Elaboration, Construction and Transition

## Workflows and Activities

The step follow ascending numbers

1. All members inside group discussed about total plan to build up payroll management application.
2. We divided and assigned tasks for every members.
3. Phuc and Hiep are assigned to be in charge of Requirements Definition part and set out Requirement Management Plan.

# Milestones

In this section both internal and customer releases updating the requirements.

* 28-10-2019: Ideas on requirement were proposed
* 29-10-2019: Setting collect the requirement.
* 30-10-2019: The Requirements Management Plant was modified and completed.

# Training and Resources

* Resources: Visual Paradigm, online Design Drawing tool or some similar software is required to draw UML diagrams included in our report. NetBeans IDE and XAMPP are required to build our an application payroll management system.
  + Design frontend – Java Swing
  + Implement Backend – Java Language and mySQL .
* Training: All of people should understand requirements and have basic knowledge about Java language and Query database.