Computer Shop System

Requirements Management Plan

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 31/10/2019 | 1.0 | None | Hoang Trung Hieu |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 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 Attributes for <traceability item> 5

3.4 Reports and Measures 7

3.5 Requirements Change Management 7

3.5.1 Change Request Processing and Approval 7

3.5.2 Change Control Board (CCB) 8

3.5.3 Project Baselines 8

3.6 Workflows and Activities 9

4. Milestones 9

5. Training and Resources 9

Requirements Management Plan

# Introduction

This Requirements Management Plan provides an overview of how the requirements of our online computer shop are managed. Through this document, it is simple to understand how computer shop system works. Moreover, the details of plan will be showed. It includes the purpose, scope, definitions.

## Purpose

This requirements management plan shows the tool and process that are used in the computer shop system. The requirements management plan is to describe how the online computer shop 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 establish an online computer shop.

## Overview

The rest of this document contains Requirements Management, The Requirements Management Program, Milestones and Training and Resources. Requirement management is a brief description of who is going to be responsible for performing the various activities described in the requirements workflows as well as tools, environment and infrastructure are used in fulfilling the requirements management functions throughout the project. The requirement management program is also inspected and evaluated. Milestones are section which identifies the internal and customer milestones related to the requirements management effort. Finally, training and resources describe the software tools, personnel, and training required to implement the specified requirements management activities.

# Requirements Management

## Organization, Responsibilities, and Interfaces

In our group, Nguyet is reponsible for Business Modeling, Hieu and Quang is responsible for Requirements Definition, Dat and Trang is responsible for Analysis & Design and Prototyping.

## Tools, Environment, and Infrastructure

|  |  |
| --- | --- |
| **Tool** | **Purpose** |
| StarUML | Draw usecase diagram, class diagram and sequence deiagram. |
| Word | Write Project |
| Visual Studio Code | To write and edit code |
| Notepad++ | To write and edit code |
| Excel | To write report and list |

The interface of the Computer shop website is designed with some programing languages such as HTML, CSS, JavaScript. And the back-end use Mysql and PHP.

# The Requirements Management Program

## 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 requirements.

## Attributes

### Attributes for <traceability item>

Attributes assigned to each requirement will be used to manage the software development process and to prioritize the features for each release.Status

Tracks progress during definition of the project baseline and subsequent development.

|  |  |
| --- | --- |
| 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.  Only PM can change the state from Proposed to Approved |
| Rejected | Capabilities that are not deemed to be useful and or feasible and have been rejected by the official channel or the requirement is not a valid stakeholder request  Only PM can change the state to Rejected |
| Incorporated | Features incorporated into the product baseline at a specific point in time. |

****Benefit****

Describes the relative importance of STRQ or FEAT in the Target Release

|  |  |
| --- | --- |
| Critical | Indicates that the requirement is of the highest importance. |
| Important | Indicates that the requirement is critical to the long-term success of the product. However, meeting this requirement can be planned for in a later release if cost and schedule considerations indicate this to be the most prudent course of action. The architecture will enable this as a firm future requirement and cannot prohibit its eventual implementation. |
| Useful | Indicates that this requirement will add significant utility to the product offering. However, if meeting the requirements adds significant cost or duration to the program, it can be disregarded. |

****Effort****

Set by the development team. Because some supplementary requirements 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 effort 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 development team based on the probability that this requirement will cause undesirable events to the project, such as cost overruns, schedule delays or even cancellation. Remember that in the RUP we want to target the high risk requirements in the first iterations to enable our clients to walk away if things aren’t going to work.**

**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 and/or the probability of the risk occurring is less.**

**Low: The impact of the risk is minimal and the probability of the risk occurring is low. <25%**

****Stability****

Set by the development team. Relative ranking on how difficult the feature or supplementary requirement will be to implement. Used in managing scope and determining development priority

****Target Release****

This helps tester determine what was implemented. PM can use it to determine how many requirements were not implemented.

****Assigned to****

Functional team or teams this requirement is assigned to names of the team.

****Reason****

Any assumptions that stakeholder made while creating the stakeholder request or that made while creating features.

## Reports and Measures

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

The Rational Unified Process (RUP) is an iterative software development process framework created by the Rational Software Corporation, a division of IBM since 2003. RUP is not a single concrete prescriptive process, but rather an adaptable process framework, intended to be tailored by the development organizations and software project teams that will select the elements of the process that are appropriate for their needs. RUP is a specific implementation of the Unified Process.

## Requirements Change Management

### Change Request Processing and Approval

When working on project, expect change. While change can have a significant impact on a project, it’s change requests that aren’t appropriately approved, incorporated, and communicated that cause significant issues and have negative impacts that often spiral out across the organization. In this section, we look at how to manage change requests so that an informed decision can be made about whether or not to approve them, and how change can be incorporated into a project with as little disruption as possible.

The first question to consider is what exactly the scope of the change request is. A change request could be related to the business, stakeholder, or functional requirements. Along with identifying what the change is, you’ll want to identify the benefit of making change or the business need driving the change as well. This will help your change approval team determine whether or not the proposed change is worth the effort.

Do the changes affect other modules and arise conflicts? If the changes have this negative effect, rejected them.

If the changes bring many values, they are higher prioritized.

If the changes affect even change structure, they are implemented early.

How much does the change extend the baseline? If the change drags the project baseline, deprioritized them.

Based on the change assessment, the changes which are of highest priorities will be approved by the CCB. Lower-priority changes will have status of pending, and lowest-priority changes will be rejected.

### Change Control Board (CCB)

The change control board includes team leader. Our team consists of five members and Hoang Trung Hieu is team leader. Opinions of every member are proposed and discussed to change requests appropriately, finally, approvals are followed by team leader.

### Project Baselines

The project baseline was approved by the team leader.

• 15-10-2019: Project kick start.

• 16-10-2019 – 23/10/2019: Determine project scope, a vision document, a use-case model survey, an initial project glossary, initial risk assessment.

• 24-10-2019 – 2/11/2019: Identify all use-cases and actors, supplementary requirements (non-functional or not associated with a use case), software architecture description and execute architectural prototype.

• 3-11-2019 – 2-12-2019: Provide a product ready to put into the hands of end users, the software product integrated on the adequate platforms and a description of the current release.

• 3-12-2019 – 12-12-2019: Deployment-specific engineering: cutover, commercial packaging and production, sales roll-out, field personnel training.

Tuning activities: bug fixing, enhancement for performance and usability.

Assessment of the deployment baselines against the complete vision and the acceptance criteria for the product.

• 14-12-2019: Project finishes.

## Workflows and Activities

Firstly, business modeling is required. Then, all the members discuss and make the decision on what our group needs to establish an online computer shop. Finally, Hieu and Quang is assigned to be in charge of the Requirements Definition part and set out Requirements Management Plan.

# Milestones

- 14-10-2019: Ideas on requirements were proposed.

- 17-10-2019: Ideas were unified to make the official and ultimate requirements of this project.

- 23-10-2019: A Requirements Management Plan was set out.

- 23-10-2019 – 7-11-2019: The Requirements Management Plan was modified and completed.

# Training and Resources

StarUML or Vision or similar software is required to draw UML diagrams included in our project report. Visual Studio Code or programming software’s is required to build our online computer shop system. Since our website‘s backend is included with Mysql and PHP.