

**IVOLUNTEER SOCIAL NETWORK**

**Configuration Management Plan**

**Project Code : IVSN**

**Ha Noi, 16/05/2016**

SIGNATURE PAGE

AUTHOR: Nguyễn Thạc Sơn 16/05/2016

Team member

REVIEWERS: Nguyễn Minh Đức

Project Manager

APPROVAL: Bùi Đình Chiến

Supervisor

Record of change :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Effective Date | Changed Item | Add, Modify, Delete | Change Description | Reason for Change | Revision Number |
| 16/05/2016 | Add new | Add |  |  | 0.1 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 1. INTRODUCTION

The purpose of this document is to identify and describe Configuration management (CM) process implementing in the project.

## 1.1. Role & Responsibility

|  |  |  |
| --- | --- | --- |
| **Role** | **Responsibility** | **Full name** |
| Project Manager | Have overall responsibility of the project:  - Project planning and scheduling.  - Task assignment and tracking processing.  - Review documents.  - Reporting to supervisor. | DucNM |
| Technical Leader | - Technical Leader is responsible for technologies, requirements, architecture design, techniques used in project and source code version.  - Coding functions and modules of system. | DucNM |
| Programmer #1 | - Study technologies, software degisn documents  - Coding functions and modules of system. | SonNT |
| Test Leader | - Create test plan, test case, test report, quality report.  - Execute test. | TamNV |
| Tester #1 | - Support creating test plan, test case, test report, quality report.  - Execute test. | TrongPV |
| Design  Leader | - Design Leader is responsible for user interface design.  - Review design of others member. | PhucVT |
| Designer #1 | - Support creating screen design. |  |

## 1.2. Definitions and Acronyms

| Acronym | Definition | Note |
| --- | --- | --- |
| IVSN | iVolunteer Social Network |  |
| ADD | Architecture Design Document |  |
| CI | Configuration Item |  |
| CM | Configuration Management |  |
| PP | Project Plan |  |
| CSCI | Computer Software Configuration Items |  |
| DDD | Detail Design Document |  |
| PM | Project Manager |  |
| QA | Quality Assurance Officer |  |
| SRS | Software Requirement Specification |  |
| Source | Source Code |  |
| URD | User Requirement Document |  |
| TP | Test Plan |  |
| PIC | Person in Charge |  |
| VSS | Microsoft Visual Source Safe |  |

# 2. CONFIGURATION MANAGEMENT PROCESS

## 2.1. CI Identification and Naming convention

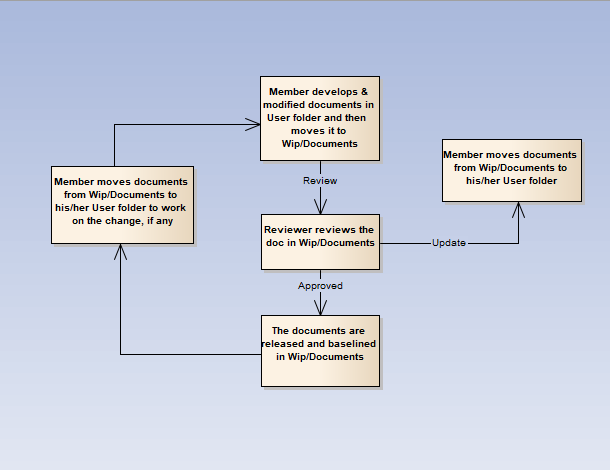
| CSCI | Configuration Items | Naming Conventions |
| --- | --- | --- |
| Project Management | PP | IVSN\_ProjectPlan\_v<x.x> \_language |
| CM Plan | IVSN\_Configuration Management Plan\_v<x.x>\_language |
| Introduction | IVSN\_Introduction\_EN |
| Risk Management | IVSN\_Risk Management\_language |
| Requirement & Design | Software Requirement Specification | IVSN\_Software Requirement Specification\_v<x.x> \_language |
| Screen Design | IVSN\_Screen design\_v<x.x>\_language |
| Architecture Design | IVSN\_Architecture Design\_v<x.x>\_language |
| Class Design | IVSN\_Class Design\_v<x.x>\_language |
| Data Design | IVSN\_Data Design\_v<x.x>\_language |
| Prototype | Prototype\_v<x.x> |
| Build | Source code | IVSN\_Source Code\_v<x.x> \_Tested/Untested |
| Test Plan | IVSN\_Test Plan\_v<x.x>\_language |
| System Test Case | IVSN\_System Test Case\_v<x.x>\_language |
| Integration Test Case | IVSN\_Integration Test\_v<x.x>\_language |
| Unit Test Case | IVSN\_Unit Test Case\_v<x.x> \_language |
| Test report | IVSN\_Test Report\_v<x.x>\_language |
| Process | Guideline\_Name of guideline | IVSN\_Guideline\_Title of guideline |
| Checklist\_Name of checklist | Checklist\_Title of checklist\_v<x.x> |
| Document type | MS Word | \*\*\*.docx |
| MS Project | \*\*\*.mpp |
| MS Excel | \*\*\*.xlsx |

## 2.2. Project Infrastructure

| Tool | Version | Purpose | Note |
| --- | --- | --- | --- |
| Git | 2.7 | For source version control | https://github.com |
| GitHub Desktop | Free | For using git with GUI | https://desktop.github.com/ |
| Astah | 7.0 | For architecture design | http://astah.net/ |
| MS Office | 2013 | For Documentation |  |
| Lightshot | Free | For screen capture Document | https://app.prntscr.com/en/index.html |
| Sublime Text | 2 (Trial) | Text editor | <http://www.sublimetext.com/3.html> |
| Visual Studio 2015 | Enterprise | Develop |  |
| Trello | Free | Tasking and task management | https://trello.com/ |
| MongoDB 3.2.6 | Free | Develop | https://www.mongodb.com/ |
| Draw.io | Free | Draw diagram | https://www.draw.io/ |

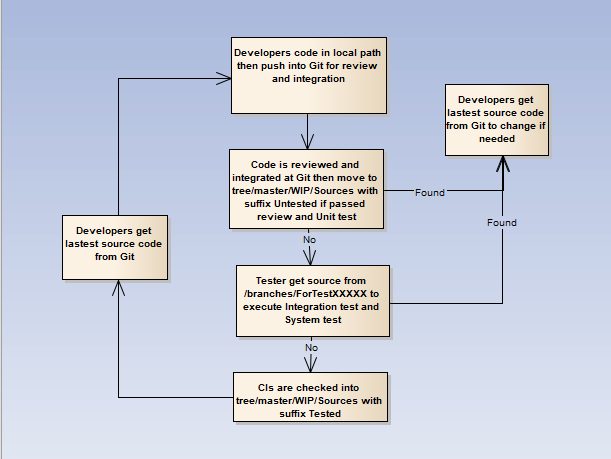
## 2.3. CI Baseline Procedure

### 2.3.1. For Document



Hình 1.

### 2.3.2. For Source Code



Hình 2.

## 2.4. Project Baseline Procdure

| No. | Baseline Name | When Baseline | PIC |
| --- | --- | --- | --- |
| 1 | Startup | Within 7 days from approval. It is mandatory requirement that version of all CI at Startup baseline to be archived in separate folders in Archive area | Project team |
| 2 | Solution | When Architectural design v1.0 is released and baseline | Project team |
| 3 | Construction | Right the end of development phase | Project team |
| 4 | Wrap-up | After the final release. It is mandatory requirement that version of all CI at Wrap-up baseline to be archived in separate folders in Archive area | Project team |

## 2.5. Directory structure & Access right

### 2.5.1. Promotion Area

| Area | Purpose |
| --- | --- |
|
| Develop Area | Area for different users to store his/her owned items |
| Review Area | To store items that is ready for review.  Reviewer get to be-reviewed items from this area |
| Test Area | Just applicable for Source items.  To store items passed Unit Test and Code Review |
| Release Area | To store the items ready for release and all released versions of items  Users get the most recent items for their usage from this area |
| Archive Area | To archive all released versions of each CI  Archive area is a protected area for project baselines where all the CIs cannot be changed by any member |

### 2.5.2. Directory Structure

| Main Folder | Sub Folder | Purpose | Map to Area | Access right |
| --- | --- | --- | --- | --- |
| Project Directory: | | | | |
| WIP | Documents | Documents of Requirements, Design, Test, … | Release + Review | Full: PM, CC  Modify: PIC  Read: All |
| Deliverables | Document of Reports (1->6) to deliver | Release + Review | Full: PM, CC  Modify: PIC  Read: All |
| Meeting minutes | Store project meeting minutes, including meeting minutes with customer | Release + Review | Full: PM, CC  Modify: All |
| Plan | Estimation, Project Plans, Project schedule, Task list | Review + Release | Full: PM, CC  Modify: PTL  Read: All |
| Record | Store project records, divided into  Review: include Review, Test and Inspection records  Change request  Acceptance | NA | Full: PM, CC  Modify: All |
| Source | Store VSS file of Source code | Archive | Refer to VSS directory |
| Users | Workplace for each members | Review + Delevop | Full: PM, CC  Modify: PIC  Read: All |
| Reference | Process | Store Documents and Other materials/data supplied by customer or those support software development and production operation in the project… | Release | Full: PM, CC  Modify: PIC  Read: All |
| Template | Store Guidelines/Standards/Forms/Templates/Checklist specified for the project usage | Release | Full: PM, CC  Modify: PIC  Read: All |
| Final |  | Final document | Release | Full right: Project QAs  Read right: All |

### 2.5.3. Access Control

Access right of non-project team members (ex: auditor, external reviewer, etc…) must be get permission of PM and granted in the pre-defined duration, then revoked at expiry date by CC. As soon as a member is out of the project, his or her access right is revoked also.

The access right is reviewed frequently and updated by CC at <baseline point and project closure time>.

After project asset is approved by QA at project closure time, PM informs to IT Department to revoke the access right of all project team members. If someone has a request for data reference, audit, etc…, he or she must get the approval of authorized person, normally Group Leader or Division Leader, and then send the request to IT Department. IT Department is responsible for implementing such kind of requests.

## 2.6. Version numbering rule

### 2.6.1. For Document

The version level is maintained as numbered identifier with two components

Version

Revision

1.0

The original version will be numbered 0.1. Subsequent revisions will be numbered 0.2, 0.3. The release version will be 1.0.

**Version number**, which appears to the left of the decimal. It changes only when the core architecture of the item changes. For example: when an item is completely overhauled, with substantial internal changes, the version 1.0 would become version 2.0

**Revision number**, which appears to the right of the decimal. It changes when existing content is changed, but the overall structure and flow of the item remains the same. The normal sequence of revision is 1.1, 1.2 and so on.

### 2.6.2. For Source Code

Software executable and support files are generally identified by name and version number, such as “Main DB v1.1.a”. The scratch edition will be 1.0. The version numbering scheme consists of three components:

Revision

1.1a

Version

Update

**Version number**, which appears to the left of the decimal. It changes only when the core architecture of the software item changes, as when moving from one area of the development tool to another, when an application is completely overhauled, or the user interface changes fundamentally. In this case, version 1.1a would become version 2.0.

**Revision number**, which appears to the right of the decimal. It changes when new features, functionality or other content are added or significantly changed. In normal case, the core architecture or user interface have been extended or limited in some manner. The most common reason for changing the revision number is when adding a new module or other functionality to the software. The normal sequence of revision is 1.0, 1.1, and 1.2 and so on.

**Update level** is appended or incremented when the only change to the software item is to correct one or more defects, without the addition of any new functionality. Version 1.1 would become v1.1a, v1.1b and so on. This updating is over ridden when a combination revision, involving bug fixes and new feature additions, is performed. In such a case, the software revision number is incremented and any update indicator is dropped, as in v1.1b to v1.2

## 2.7. Other CM Rule

**Rule for deliverables releases:** When a member releases any WP, he/ she must self-review carefully before have other members review and have clearly comments.

**Rule for changing document:** When a member wants to modify documents, he/she must copy the documents from Documents folder to User folder to work on the change (if any), then merge it to Document folder with another version and has clearly comment. Members are absolutely not allowed to edit directly on the Project folder.

**Rule for development:** Refer Coding guideline.