

Event zone

PROJECT PLAN

**Project code: EZ**

**Document code: EZ\_ProjectPlan\_v1.1**

**Hoa Lac, September 30th 2015**

Signature page

Author： Phan Thanh Vu 19/9/2015

Project Manager

Reviewer： Nguyen Van Cuong 20/9/2015

Technical Leader

Authorizer： Tran Binh Duong 20/9/2015

Supervisor

**Change History**

**\*A - Added M - Modified D – Deleted**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Effective Date | Changed Item | A\* M, D | Change Description | Reason of Change | Version |
| 19/9/2015 | **Create New** | **A** | **Create Document** |  |  |
| 30/9/2015 | **M,D** | **M,D** | **Modified Man-hour calculating**  **Modified Risk**  **Delete Risks aren’t related** |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Contents**

[1. Project Introduction 8](#_Toc431392363)

[1.1. Project Description 8](#_Toc431392364)

[1.2. Scope and Purpose 8](#_Toc431392365)

[1.3. Assumptions and Constraints 10](#_Toc431392366)

[1.4. Project Objectives 11](#_Toc431392367)

[1.4.1. Standard Objectives 11](#_Toc431392368)

[1.4.2. Specific Objectives 12](#_Toc431392369)

[1.5. Critical Dependencies 12](#_Toc431392370)

[1.6. Project Risk 13](#_Toc431392371)

[2. Project Development Approach 16](#_Toc431392372)

[2.1. Project Process 16](#_Toc431392373)

[2.2. Request Change Management 18](#_Toc431392374)

[2.3. Quality Management 18](#_Toc431392375)

[2.3.1. Defect Prevention Strategy 18](#_Toc431392376)

[2.3.2. Review Strategy 19](#_Toc431392377)

[2.3.3. Unit Test Strategy 21](#_Toc431392378)

[2.3.4. Integration Test Strategy 22](#_Toc431392379)

[2.3.5. System Test Strategy 23](#_Toc431392380)

[2.3.6. Defect Estimation Strategy 23](#_Toc431392381)

[2.3.7. Measurement Strategy 24](#_Toc431392382)

[3. Estimation 25](#_Toc431392383)

[3.1. Size 25](#_Toc431392384)

[3.2. Man-hour 25](#_Toc431392385)

[3.3. Schedule 26](#_Toc431392386)

[3.3.1. Project Milestones and Deliverables 26](#_Toc431392387)

[3.3.2. Project Schedule 28](#_Toc431392388)

[3.4. Infrastructure 28](#_Toc431392389)

[4. Project Team 30](#_Toc431392390)

[4.1. Structure 30](#_Toc431392391)

[4.2. Project Team 30](#_Toc431392392)

[4.3. External Interface 32](#_Toc431392393)

[4.3.1. University Interface 32](#_Toc431392394)

[5. Communication and Reporting 33](#_Toc431392395)

[6. Configuration Management 36](#_Toc431392396)

[6.1. Introduction 36](#_Toc431392397)

[6.2. Configuration Management Process 36](#_Toc431392398)

[6.2.1. CI Identification and Naming Convention 36](#_Toc431392399)

[6.2.2. Version Numbering Rule 40](#_Toc431392400)

[6.2.3. Directory structure 41](#_Toc431392401)

[6.2.4. Other CM Rules 42](#_Toc431392402)

[Figure 1 Iterative and Incremental Software Process Model 15](#_Toc431392257)

[Table 1: Project Description 8](#_Toc431392271)

[Table 2 Assumptions and Constraints 10](#_Toc431392272)

[Table 3 Standard Objectives 11](#_Toc431392273)

[Table 4 Standard Goal Objectives 11](#_Toc431392274)

[Table 5 Critical Dependencies 11](#_Toc431392275)

[Table 6 Project Risk 12](#_Toc431392276)

[Table 7 Request Change Management 17](#_Toc431392277)

[Table 8 Defect Prevention Strategy 18](#_Toc431392278)

[Table 9 Review Strategy 20](#_Toc431392279)

[Table 10 Unit Test Strategy 21](#_Toc431392280)

[Table 11 Integration Test Strategy 21](#_Toc431392281)

[Table 12 System Test Strategy 22](#_Toc431392282)

[Table 13 Defect Estimation Strategy 23](#_Toc431392283)

[Table 14 Measurement Strategy 24](#_Toc431392284)

[Table 15 Man-hour Estimation 25](#_Toc431392285)

[Table 16 Milestones and Deliverables 27](#_Toc431392286)

[Table 17 Infrastructure 29](#_Toc431392287)

[Table 18 Project Team 30](#_Toc431392288)

[Table 19 Human Resource Allocation 31](#_Toc431392289)

[Table 20 University Interface 31](#_Toc431392290)

[Table 21 Communication and Reporting 35](#_Toc431392291)

[Table 22 CI Identification and Naming Convention 38](#_Toc431392292)

[Table 23 Directory Structure 41](#_Toc431392293)

# Project Introduction

## Project Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Code** | Event Zone | Contract Type |  |
| **Customer** | FPT University | 2nd Customer |  |
| **Project Level** | Group | Project Rank |  |
| **Group** | JS0801 | Department |  |
| **Project Format** | Internal | Project Manager | Phan Thanh Vu |
| **Project Category** | Development | Business Area |  |
| **Application Form** | Website Application |  |  |

Table 1: Project Description

## Scope and Purpose

The project is registered and implemented as capstone project for the team member. Our purpose is meet the requirements of FPT University studying program. The core idea of the system is making connection between event managers and event viewers to easier than ever. The information is completely made by user and process by system. Users who want to promote their event to more and more people can register their event here and make a space to for everyone to come and build up information. They can also live streaming their event for people who can’t come to watch online. Users who want to view event can search on their location to find the best fit event for them or search by name. They can also interact with others viewers by like, comment or share their watching events.

Function and feature to be implemented:

*For User:*

* Search Event by Text or by Location
* Manage Event
* Streaming Event Online
* Interact in Event: Like, Share, Comment
* Follow Category
* View Statistic
* Report Violated Events
* Appeal Violation Reports

*Out-of-scope:*

* Making List Events
* Streaming Server
* View Schedule
* Follow Other Users

## Assumptions and Constraints

|  |  |  |
| --- | --- | --- |
| No | Description | Remarks |
| Assumptions | | |
| 1 | Project use C# .NET as code language | Scope |
| 2 | Project use Microsoft SQL for database | Database |
| 3 | Project have been registered to Department of Information and Communications and Department of Culture, Sports and Tourism | Others |
| 4 | Team member should have work at least 5 hours a day | Schedule |
| 5 | Budget: 500USD | Budget |
| Constraints | | |
| 1 | The Project should be delivered before 06/12/2015 | Schedule |
| 2 | While doing project, PM should submit all 6 reports before deadline | Schedule |
| 3 | Team budget doesn’t exceed 500USD | Budget |

Table 2 Assumptions and Constraints

## Project Objectives

### Standard Objectives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metrics | Unit | Submitted | Resubmitted | Remark |
| Start Date |  | 1/9/2015 |  |  |
| End Date |  | 21/12/2015 |  |  |
| Period | Day | 86 |  |  |
| Maximum Team Size | Person | 6 |  |  |
| Billable Effort | Man-hour | 4368 |  |  |
| Calendar Effort | Man-hour | 4368 |  |  |
| Used Effort | Man-hour | 3042 |  |  |

Table 3 Standard Objectives

|  |  |  |  |
| --- | --- | --- | --- |
| Metrics | Unit | Goals | References |
|
| Leakage | Wdef/mm | 5 |  |
| Effort Efficiency | % | 95 |  |
| Timeliness | % | 100 |  |

Table 4 Standard Goal Objectives

### Specific Objectives

We don’t have specific objectives

## Critical Dependencies

|  |  |  |  |
| --- | --- | --- | --- |
| No | Dependency | Delivery Date | Remarks |
| 1 | FPT University requires: Progress report, Meeting Minutes must be in Japanese |  | Before delivered date |
| 2 | All reports must be delivered to FPT University and Supervisor |  | Before delivered date |
| 3 | Project must be completed and delivered to FPT University | 06/12/2015 |  |
| 4 | Presentation must be in both English and Japanese | 21/12/2015 |  |

Table 5 Critical Dependencies

## Project Risk

Table 6 Project Risk

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Risk Description | Root Cause | Avoidance  Plan | Contingency  Plan | Probability | Impact | Status |
| 1 | Overestimate or under estimate time of project | Wrong estimate about time of each phases. | - Create detail plan for each project’s phase.  - Breakdown project to some mile-stone releases. Main function development is high priority  - Get the team more involved in planning and estimating. Get early feed-back and address slips directly with stakeholders.  - Consult supervisor’s advice | - Implement task in parallel or overlapping them  - Overtime to push progress faster | High | High |  |
| 2 | Run over budget | -Spend too much | - Saving  - Reasonable spending | - Make a contributions | Low | Medium |  |
| 3 | Member do not have enough knowledge to follow project | -Member does not have skill about technologies | - Training  - Encourage member learn from others | - Working in group | Medium | Medium |  |
| 4 | Have some problem with team member such as: absent, overtime, conflict, healthy… | -Personal  -Over whelming or lazy  -Weather | - Closely monitor  - Allocation of reasonable working time, concerned about member | - Overtime  - Increase work performance  - Mobilize others members | High | High |  |
| 5 | Missing Features |  | - Brainstorming between members  -Team meeting to define and discuss about all features in project | - Team meeting with supervisor to determine whether features are implemented or not  - If features is must executed, Project manager create implement plan for this feature and push progress quickly | Medium | High |  |
| 6 | Productivity isn’t met plan | - Low team member’s motivation  - Time lost | - Concentrate on creating good plan and deadline for each feature, estimate reasonable time for each project phase  - Organize team building so that team member can get well to each other | - Short iterations, right people on team, coaching and team development.  - Conduct a meeting so that the team members can express their opinions and find out solutions | Medium | High |  |

# Project Development Approach

## Project Process

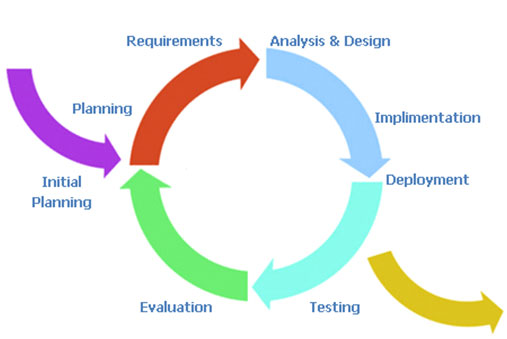
[[1]](#footnote-1)

Figure 1 Iterative and Incremental Software Process Model

Iterative development prescribes the construction of initially small but ever-larger portions of a software project to help all team member to uncover important issues early before problems or faulty assumptions can lead to disaster. Iterative development prescribes the construction of initially small but ever-larger portions of a software project to help all team member to uncover important issues early before problems or faulty assumptions can lead to disaster.

Project Life Cycle:

* Initiating: Register capstone project and supervisor, create project introduction
* Planning: Make Project plan to control process and schedule of project
* Executing:
  + Inception Phase: Deal with the scope of the project, requirements at higher level
  + Basic Elaboration and Construction Phase: Design and Deliver project with basic functions and User Interface
  + Detailed Elaboration and Construction Phase: Fills in architecture components incrementally with production-ready code, which is produced through the analysis, implementation, design and testing of functional requirements
  + Final Construction and Transition Phase: Delivers the system to the production operating environment
* Monitoring and Controlling: Guarantee that all purposes are met, qualities are assured
* Closing: Create final report and prepare presentation.

## Request Change Management

|  |  |
| --- | --- |
| Recorded Change Request File | EZ\_RequestChange.xls |
| Change Request Recorder | HaNS |
| Change Request Reviewer | People whose work will be affected by change of requests.  PM |
| Change Request Approval | Supervisor and PM |

Table 7 Request Change Management

## Quality Management

### Defect Prevention Strategy

|  |  |  |
| --- | --- | --- |
| Item | Strategy | Expected Result |
| Requirement missing | List up all of requirement into SRS document | 10–20% reduction in defect injection rate and about 2% improvement in productivity |
| Careless mistake in Design Document Format/Template wrong | After designing, all team will review Document Format and content based on last meeting reports | Improvement in quality as overall defect removal efficiency will improve; some benefits in productivity as defects will be detected early |
| Data is not reliable | Search data from reliable sources -> Filter information | Avoid confusing data. |
| Duty neglecting | Check task list per 2 day | Focus on what I am doing |
| Inconsistency | Define naming convention for document, coding convention for coding | Easy to manage and combine. |

Table 8 Defect Prevention Strategy

### Review Strategy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review Item | Reviewer | Review Format | Review Method | Completion Criteria |
| Project plan | All team member, Supervisor | Group review | Self-review |  |
| Project schedule | All team member, Supervisor | Group review | Self-review |  |
| CM Plan | All team member, Supervisor | One-person review | Self-review |  |
| Requirements specification document | All team members, Supervisor | Group review | Self-review |  |
| Use Case catalog | All team members, Supervisor | Group review | Self-review |  |
| Design document | All team members, Supervisor | Group review | Self-review |  |
| Stage plans | PM, Supervisor | One-person review | Self-review |  |
| Complex/first time generated program specs incl. test cases, interactive diagrams | All team members, Supervisor | Group review | Self-review |  |

Table 9 Review Strategy

### Unit Test Strategy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item to Be Unit Test | Type of Unit Test | Unit Test Technology | Tool | Completion Criteria |
| Code | White-box | Unit test case  Test script | None | - Number of UTC/KLOC: 40 UTC/KLOC  -Number defects/KLOC: 3-4 defects/KLOC  - Statement coverage: 97%  - Branch coverage: 100%  - Path coverage: 100% |

Table 10 Unit Test Strategy

### Integration Test Strategy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item to Be Integration Test | Type of Integration Test | Integration Test Technology | Tool | Completion Criteria |
| Do test by flow of functions and items which have concern each other | Black-box |  | Checklist, Boundary | - Number of UTC/KLOC: 30  - Number of defects/KLOC: 2-3 |

Table 11 Integration Test Strategy

### System Test Strategy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item to Be System Test | Type of System Test | System Test Technology | Tool | Completion Criteria |
| Test whole system | Black-box |  | None | - Number of UTC/KLOC: 30  - Number of defects/KLOC: 2-3 |

Table 12 System Test Strategy

### Defect Estimation Strategy

|  |  |  |  |
| --- | --- | --- | --- |
| Review/ Test Stage | Target Value | Defects Rate | Basis of estimation |
| Requirements review | 15 | 11% | Referenced similar project estimations |
| Design review | 14 | 9% | Referenced similar project estimations |
| Code review | 29 | 20% | Referenced similar project estimations |
| Unit Test | 57 | 40% | Referenced similar project estimations |
| Integration Test | 15 | 10.2% | Referenced similar project estimations |
| System Test | 10 | 6.8% | Referenced similar project estimations |
| Total | 138 | 100% |  |

Table 13 Defect Estimation Strategy

### Measurement Strategy

|  |  |  |  |
| --- | --- | --- | --- |
| Aggregation Target of Data | Target | Person in Charge | Time |
| Size: No. of KLOC | Achieve target | PM | At the end of stages |
| Effort: No. person-day | Match with calendar effort | Team members | Daily |
| Quality: No. defects detected | Match with target quality | Reviewer, Tester | Right after the review/test |
| Schedule | On-time | PM | Weekly and at the end of stages |

Table 14 Measurement Strategy

# Estimation

## Size

The size of the project is limit by Capstone Project Requirement.

## Man-hour

|  |  |  |
| --- | --- | --- |
| Activities / Process | Planned Man-hour | Estimated Time Usage of Total (%) |
|
| Initiating | 283 | 9.3 |
| Planning | 51.52 | 1.7 |
| Inception | 460 | 15.1 |
| Basic Elaboration and Construction | 850.43 | 28 |
| Detailed Elaboration and Construction | 442.83 | 14.6 |
| Final Construction and Transition | 291.5 | 9.6 |
| Monitoring and Controlling | 159 | 5.2 |
| Closing | 503.5 | 16.6 |
| Total | 3042 |  |

Table 15 Man-hour Estimation

## Schedule

### Project Milestones and Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Stage | Delivery Date | Descriptions | Verification |
| 1 | Create Project Introduction | 09/09/2015 | Analyze Background.  Compare between existing system.  Propose system.  Document are reviewed. |  |
| 2 | Submit Report 1 | 13/09/2015 | Document are reviewed.  Completed. |  |
| 3 | Create Project Plan | 17/09/2015 | Document are reviewed |  |
| 4 | Submit Report 2 | 20/09/2015 | Document are reviewed.  Completed. |  |
| 5 | Create Report 3 Draft | 27/09/2015 | Document are reviewed.  Completed. |  |
| 6 | Submit Report 3 Full | 04/10/2015 | Document are reviewed.  Completed. |  |
| 7 | Architecture Design | 07/10/2015 | Document are reviewed. |  |
| 8 | Screen Prototype | 07/10/2015 | Document are reviewed. |  |
| 9 | Screen Design | 07/10/2015 | Document are reviewed. |  |
| 11 | Submit Report 4 Draft | 01/10/2015 | Document are reviewed.  Completed. |  |
| 12 | Create Test Plan | 13/10/2015 |  |  |
| 13 | Source Code draft | 18/10/2015 | Basic functions are done and tested |  |
| 14 | Source Code Alpha | 01/11/2015 | User Management functions are done and tested |  |
| 15 | Submit Report 4 Full | 11/11/2015 |  |  |
| 16 | Source Code Beta | 15/11/2015 | API functions are done and tested |  |
| 17 | Submit Report 5 | 25/11/2015 |  |  |
| 18 | Submit Report 6 | 29/11/2015 |  |  |
| 19 | Lesson Learned | 30/11/2015 |  |  |
| 20 | All Report | 06/12/15 |  |  |
| 21 | Present Capstone Project | 19/12/2015 |  |  |
| 22 | Project Complete | 20/12/2015 |  |  |

Table 16 Milestones and Deliverables

### Project Schedule

Schedule is described in EZ\_ProjectWBS.mpp

## Infrastructure

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Description | Deadline | Remarks |
| Development Environment | | | |
| Operating System | Windows 8.1/10 |  |  |
| Browser | Chrome (Latest version) |  |  |
| Technoogy | | | |
| DBMS | SQL Server 2010 |  |  |
| Development language | .NET C# MVC Model |  | For back-end |
| Development language | HTML, CSS, JavaScript |  | For front-end |
| Hardware Requirements | | | |
| Hardware Configuration | 4GB RAM |  |  |
| Equipment and Tools | | | |
| Design | Astah Professional |  |  |
| Source Version Control | Git |  |  |
| Code Review | None |  |  |
| Project Management Tool | Microsoft Project 2013 |  |  |
| Task Tracking | Microsoft Excel 2013 |  |  |
| Documentation | Microsoft Word 2013 |  |  |

Table 17 Infrastructure

# Project Team

## Structure

## Project Team

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Role | Responsibility | Name | Format | Percent of Used Man-hour of Estimated Man-hour |
| PM | Have overall responsibility of the project:  - Project planning and scheduling  - Task assignment and tracking processing  - Review documents  - Reporting to supervisor | Phan Thanh Vu |  | 76.7 |
| PTL | PTL is responsible for the technical project execution | Nguyen Van Cuong |  | 68.4 |
| Dev #1 | Database Management, C# and DB Integration | Nguyen Thi Chuong |  | 67.4 |
| Dev #2 | User Interfaces Coder | Nguyen Ngoc Duong |  | 71.8 |
| Dev #3 | User Interfaces Coder | Duong Thi Lan Anh |  | 66.7 |
| Test Leader | - Create test plan, test case, test report, quality report  - Execute test. | Nguyen Sen Ha |  | 71.8 |

Table 18 Project Team

The table below as details of the input human resources allocation of the entire project.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Role | Name | W1-Sep | W2-Sep | W3-Sep | W4-Sep | W1-Oct | W2-Oct | W3-Oct | W4-Oct | W1-Nov | W2-Nov | W3-Nov | W4-Nov | W1-Dec | W2-Dec | W3- Dec | W4 -Dec | Total (day) |
| PM | Phan Thanh Vu | 65% | 52% | 73% | 83% | 83% | 83% | 81% | 95% | 81% | 80% | 80% | 83% | 47% | 68% | 39% | 54% | 91 |
| PTL | Nguyen Van Cuong | 49% | 54% | 16% | 83% | 83% | 83% | 68% | 83% | 83% | 77% | 79% | 83% | 16% | 68% | 40% | 54% | 91 |
| Dev #1 | Nguyen Thi Chuong | 50% | 50% | 8% | 83% | 83% | 58% | 83% | 83% | 83% | 77% | 80% | 81% | 33% | 67% | 33% | 50% | 91 |
| Dev #2 | Nguyen Ngoc Duong | 50% | 50% | 8% | 83% | 83% | 83% | 67% | 83% | 83% | 77% | 80% | 81% | 8% | 67% | 100% | 67% | 91 |
| Dev #3 | Duong Thi Lan Anh | 51% | 52% | 14% | 83% | 83% | 60% | 68% | 83% | 83% | 77% | 67% | 79% | 36% | 68% | 38% | 52% | 91 |
| Test Leader | Nguyen Sen Ha | 49% | 51% | 9% | 83% | 83% | 9% | 77% | 100% | 83% | 83% | 83% | 89% | 43% | 67% | 34% | 51% | 91 |

Table 19 Human Resource Allocation

## External Interface

### University Interface

|  |  |  |  |
| --- | --- | --- | --- |
| Department | Name | Contact Information | Responsibility |
| Management and Training Department | Ngo Thi Vinh Ha | hantv@fpt.edu.vn | Receive capstone project register |

Table 20 University Interface

# Communication and Reporting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Communication Format | Method/Tool | Time | Information | Participants/ Responsibilities |
| Tracking of Project Tasks | | | | |
| Task scheduling | MS Project | At the beginning of every stage, and weekly  Refinement and rescheduling as necessary |  | PM |
| Task assignment | In Excel file and via project weekly meeting | Weekly |  | PTL |
| Task status reporting | In Excel file and via project weekly meeting | Weekly |  | Project Team Members |
| Meeting of Project | | | | |
| Kick-off Meeting | Face to face | Initiation stage | Project introduction; Project plan review; Risk identification; Obtainment of commitment of relevant stakeholders | PM, Supervisor, Project Team Members |
| Project Progress Review Meetings | Face to face | Weekly and on event | Communicate project status  Communicate and resolve any open issue, risks, and changes  Discuss any suggested improvement | PM, Supervisor, Project Team Members |
| Weekly team meeting | Face to face | At the morning of Tuesday and Thursday weekly Can be canceled if necessary | Check working status of each member  Discussion | PM, Project Team Members, |
| Other | | | | |
| Information, resource sharing | Face to face Email  Git | When available | Information, documentation and resources | PM, Project Team Members |
| Raise issue Request support | Email Phone Skype Face to face | Upon request |  | Project Team Members |
| Team work | Face to face | Monday to Friday (9:00 – 17:30) |  | PM, Project Team Members |
| Technical training | Face to face | When available | Technical or knowledge training | PM, PTL, Project Team Members, External expert |
| Weekly report to the instructor | Email | Sunday | Report what was done in the week and what to do in the next week Request face to face meeting Raise issues if any | PM, Supervisor |

Table 21 Communication and Reporting

# Configuration Management

## Introduction

The purpose of this Configuration Management (CM) is to describe CM process implementing in the Event Zone project.

## Configuration Management Process

### CI Identification and Naming Convention

|  |  |  |
| --- | --- | --- |
| No | Configuration Items | Naming convention |
| Project Management | | |
| 1 | Project Plan | EZ\_ProjectPlan\_v[version number]  For example: EZ\_ProjectPlan\_v1.1 |
| Requirement & Design | | |
| 2 | SRS | EZ\_SoftwareRequirementsSpecification\_v[version number]  For example: EZ\_SoftwareRequirementsSpecification\_v1.0 |
| 3 | Architectural Design | EZ\_ArchitectureDesign\_v[version number]  For example: EZ\_ArchitectureDesign\_v0.2 |
| 4 | Screen Design | EZ\_ScreenDesign\_ v[version number]  For example: EZ\_ScreenDesign\_v1.2 |
| 5 | Data Design | EZ\_DataDesign\_v[version number]  For example: EZ\_DataDesign\_v10.2 |
| Source Code | | |
| 6 | Source Code | EZ\_SourceCode\_ v[version number][Tested/Untested]  For example: EZ\_SourceCode\_v1.0Tested |
| Support Document | | |
| 7 | User Manual | EZ\_UserManual\_v[version number]  For example: EZ\_UserManual\_v[version number] |
| Test | | |
| 8 | Unit Test Plan | EZ\_UnitTestPlan\_v[version number]  For example: EZ \_UnitTestPlan\_v1.0 |
| 9 | Integration Test Plan | EZ \_IntegrationTestPlan\_v[version number]  For example: EZ \_IntegrationTestPlan\_v1.0 |
| 10 | System Test Plan | EZ \_SystemTestPlan\_v[version number]  For example: EZ \_SystemTestPlan\_v1.0 |
| 11 | Unit Test Case | EZ \_UnitTestCase-[Test Name]\_v[version number]  For example: EZ \_UnitTestCase-Search\_v1.0 |
| 12 | Integration Test Case | EZ \_IntegrationTestCase-[Test Name] v[version number]  For example: EZ \_IntegrationTestCase-Search\_v1 |
| 13 | System Test Case | EZ \_SystemTestCase-[Test Name]\_v[version number]  For example: EZ \_SystemTestCase-Search\_ v1.0 |
| 14 | Test data | EZ \_TestData\_v[version number]  For example: EZ \_TestData\_v1.0 |
| 15 | Test Result | EZ \_TestReport\_v[version number]  For example: EZ \_TestReport\_v1.0 |
| Process | | |
| 16 | Guideline | EZ \_ [Name Of Guideline]Guideline\_v[version number]  For example: EZ \_UnitTestGuideline\_v1.0 |
| 17 | Convention | EZ \_[Name Of Convention]Conventions\_v[version number]  For example: EZ \_CodingConventions\_v1.0 |
| 18 | Template | EZ \_Template-[Name Of Template]\_v[version number]  For example: EZ \_Template-ChangeRequestForm\_v1.0 |
| File Type | | |
| 20 | MS Word | \*.docx |
| 21 | MS Excel | \*.xls |
| 22 | MS PowerPoint | \*.pptx |
| 23 | MS Project Plan | \*.mpp |
| 24 | MS Visio | \*.vsd or \*.vsdx |
| 25 | Images | \*.png or \*.jpg or \*.jpeg or \*.bmp or \*.gif |

Table 22 CI Identification and Naming Convention

### Version Numbering Rule

* For documents:
  + Each file has a version number as part of its identity. This version number is physically represented as a 2-part string with the following format: <version>.<revision>
  + For example, version 1.0 indicates 1 as the version, and 0 as the revision number.
  + The original version will be numbered 0.1. Subsequent revisions will be numbered 0.2, 0.3 and so on. The approved version will be 1.0.
  + Version number: appears to the left of the decimal. It is changed only when the core content of the item is significance changed. For example: when an item is completely overhauled, with substantial internal changes, the version 1.0 would become version 2.0.
  + Revision number: appears to the right of the decimal. It is changed when the existing content is changed, but the main (or core) content is remained. The normal sequence of revision is 1.1, 1.2, and so on.
* For Software source files:
  + Software executables and support files are generally identified by name and version number. The version number is physically represented as a 3-part string with the following format: <version>.<revision><update>
  + For example, version 1.1a indicates 1 as the version, 0 as the revision number, and a as the update level.
  + Version number: appears to the left of the decimal. It is changed only when the core content of the item is significance changed, as when moving from one are 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: appears to the right of the decimal. It is changed 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 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 function. Version 1.1 would become v1.1a, 1.1b and so on. This updating is overridden 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 1.2.

### Directory structure

|  |  |  |
| --- | --- | --- |
| Main folder | Sub-folder | Purpose |
| Finals | Report 1 | Store final deliverables of phase 1 and related Meeting Minutes and Progress Reports |
| Report 2 | Store final deliverables of phase 2 and related Meeting Minutes and Progress Reports |
| Report 3 | Store final deliverables of phase 3 and related Meeting Minutes and Progress Reports |
| Report 4 | Store final deliverables of phase 4 and related Meeting Minutes and Progress Reports |
| Report 5 | Store final deliverables of phase 5 and related Meeting Minutes and Progress Reports |
| Report 6 | Store final deliverables of phase 6 and related Meeting Minutes and Progress Reports |
| Reference | Store Reference materials needed in project | |
| Work In Progress | Working Space | Each Team Members has a folder to store working task |
| Documents | Documents which aren’t released to final |
| Coding | Codes aren’t finished |

Table 23 Directory Structure

### Other CM Rules

* Email subject naming convention: All email related to the Event Zone project must have prefix [EZ]. For example: [EZ] Weekly Report 18/09/2015
* Document changing rule: When a member want to modify a document, he/she must update version of that document with appropriate description for the modification.
* Development rule: Refer Coding guideline

1. <http://www.selectbs.com/analysis-and-design/what-is-a-software-development-process> [↑](#footnote-ref-1)