

PROJECT PLAN

Cluster C

Table of Contents

- [1. Introduction](#)
 - [1.1. Purpose of This Document](#)
 - [1.2. Scope of This Document](#)
- [2. Project overall:](#)
- [3. Deliverables:](#)
 - [3.1 Feature deliverables:](#)
 - [3.2 Customer deliverables](#)
- [4. Cluster development background](#)
 - [4.1 Integration system](#)
 - [4.2 Key Contacts and Stakeholders:](#)
- [5. Project process planning:](#)
 - [5.1. Inception](#)
 - [5.2. Construction iterations](#)
 - [5.3. Deployment:](#)
- [6. Project resource requirement](#)
 - [6.1 Staffing/skills training:](#)
 - [6.2. Tools and techniques:](#)
- [7. Project Schedule](#)
- [8. Project Configuration and Data Management](#)
- [9. Online communication and cluster documentation:](#)
- [10. References:](#)

1. Introduction

1.1. Purpose of This Document

This document outlines cluster project execution and cluster management.

1.2. Scope of This Document

In this document, we will give an overall view on what teams need to be aware of in regard to cluster, how teams in the cluster manage to cooperate throughout the project. Specifically this includes the following aspects:

- Final deliverables by both team and cluster
- Supposed environment cluster will be working on
- Contact details of stakeholders
- Cluster project process
- Resource requirement.
- Configuration and data management

2. Project overall:

This is described in cluster requirement and more details in team projects' plans.

3. Deliverables:

3.1 Feature deliverables:

See cluster requirement document for details.

3.2 Customer deliverables

Team:

- Project Code Base
- Database Schema
- How-To Documentation + Built In
- Configuration Management Guide
- Installation Guide
- All project documentation for marketing purposes

Cluster:

- API for the cluster project system with a comprehensive documentation on how to use and installation/configuration guide

4. Cluster development background

4.1 Integration system

- MySQL - SQL database system (Persistence tier) for storing project data
- Ruby - Server side application programming language
- Rails - framework for quicker implementation

4.2 Key Contacts and Stakeholders:

Name	Role	Contact
Kevin Maciunas	Course Coordinator/Client	kevin.maciunas@adelaide.edu.au
Sam Reid	Group 7 - Rotating Role	samuel.reid@student.adelaide.edu.au
Lingsheng Wu	Group 7 - Rotating Role	linsheng.wu@student.adelaide.edu.au
Khanh Hoang	Group 7 - Rotating Role	buikhanh.hoang@student.adelaide.edu.au
Andrew Nguyen	Group 8 - Rotating Role	a1627698@student.adelaide.edu.au
Qun Xu	Group 8 - Rotating Role	xuqun123@126.com
Kamal Arieff	Group 8 - Rotating Role	kamal.arieff@gmail.com
Louis Griffith	Group 8 - Rotating Role	biding.biting@hotmail.com

All role of cluster members in a specific sprint are made available to everyone in cluster to look up, so that one knows who to contact with.

5. Project process planning:

5.1. Inception

- The whole cluster tries to gather all possible front requirements and at the end of this phase divides the requirements into two separate portions for each team in the cluster to take care of.
- Cluster works together to build high-level plans and requirements for both two teams to follow.
- Each teams has responsibility for creating plans for their own teams in more details based on requirements they are deal with.

5.2. Construction iterations

- **Cluster meeting session allocation:**

Both Product Owners from two teams will be working together to figure out the appropriate meeting session time for the cluster apart from the one allocated by SEGP course.

Scrum masters from both teams will be responsible for preparing working space for this session.

- **Cluster sessions' activities:**

Demonstration from each team:

- ☐ What has done in the last iteration.
- ☐ What will be next to deal with in the current time and
- ☐ What is possibly needed for the following iterations.

This demonstration is aimed to increase the collaboration of two teams with a clear understanding on what the other team is building.

Discussion on:

- ☐ New or changing requirements from customers: Which team will be responsible for which part of the new or changing requirements, what potential problems to the cluster are raised from these new or changing requirements.
- ☐ Based on team's demonstration, cluster will have a discussion on feature prioritization: Suggestions in this regard from one team are given to the other for a consideration. These suggestions are based on both current situations of the suggester and on the sake of the other team. However feature prioritization are up to each team to decide. But final decision on that should be made visible to other team to view so as to aware of any potential risk that may affect them.
Though the general sub-system developed by each team can be quite independent, this kind of demonstration is critical to deal with any unforeseen potential problems that may suddenly pops up, causing serious issues later when each team are working together to build the whole cluster system in integration phase.
- ☐ Any specific constraints/dependencies and risk/assumption stemming from the work of one team that can influence the way the the other to develop their own system and any possible solution for them.
- ☐ Cluster project integration requirements for specific iterations:
 - Integration environment: Tools, time, data, configuration.
 - System tests for cluster system, quality/user acceptance criteria checklist.
 - A set of functionalities that the system will have after the integration
 - Time, location for both team to carry out a cluster code integration execution.
- **Integration:**
 - Running and testing the system.
 - Resolving any defect or conflict components detected during the running of the integrated system.
 - Generating documentation for any integration configuration, data usage, and other important notes.

5.3. Deployment:

- Deploy the whole system to the experimental production server provided.
- A full installation and configuration for the server should have been done at this point in time.

6. Project resource requirement

6.1 Staffing/skills training:

This is done in individual team but outsourcing assistance in this respect is made available across the cluster and class as well.

6.2. Tools and techniques:

In order to ensure the integration of both teams' works run consistently and minimize possibility of any configuration requirement, we come up with a set of tools, techniques and programming languages used for both teams: Tools and techniques used to develop and deploy the project and facilities for documentation generation and management. This table is likely to change during the time being, since there is not much experience on those facilities, but it will get solid quickly once people get used to good choices.

Framework	Ruby on Rails
Testing/QA techniques	TDD and BDD
Code editor/ IDE	Free choices: Vim, Sublime Text 3 (with ruby plugins), TextMate, Gedit, Notepad++ / Komod, Aptana RadRails, Netbeans, RubyMine, phpMyAdmin - for database admin and query debugger
Testing tools/Testing Scripting	Predefined rails testing framework, Rspec, Cucumber, Capybara (likely to change along the development process)
Cluster documentation storage (Configuration/data management/database schema or system design)	Google docs folder or Git - cluster sharing.
Code versioning	GitHub versioning system
Deployment	Capistrano

7. Project schedule:

Details are shown on giant charts provided by each team. Based on this, our cluster will allocate times to work together for a pre-integration of the codes.

8. Project Configuration and Data Management

Our cluster is provided with a Linux server by the University of Adelaide's IT Services.

No server environment installation or configuration for the server are initially available. This will be done in the configuration management stage (see Gant Chart of group 7 for more details on this).

Any configuration and data management will be documented in a separate Configuration Management document - convenient for all in the cluster to look up. The basic scheme for configuration for is as follows:

1. Install + Configure Ruby on Rails
2. Install + Configure MySQL Database
3. Install PHP if needed for #4
4. Install + Configure phpMyAdmin

9. Online communication and cluster documentation:

- Here is a initial list of online files for cluster communication and documentation, which is supposed to be stayed in long time (using google.doc and gitHub for storage):
 - Cluster discussion.
 - Cluster requirement.
 - Configuration and data usage.
 - Cluster coding standards
 - Security recommendations
 - Cluster sprint session folder which contains files detailing cluster session:
For example: File “Cluster session N-th” contains following information:
 - Meeting time and location.
 - Contact details with roles in current iteration
 - Detailing results of the session.
 - Other information
- For more iterative ways and specifics in the time beings, we use emails and other kind of communication.

10. References:

- Cluster requirement
- Group 7's project plan
- Group 8's project plan