Software Requirements Specification

for

Continues Integration Pipeline Implementation for Tech11Software

Prepared by

Aswin G SUGUNAN (29)

Jefin Jacob (3)

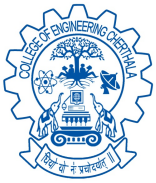
Nitin Suresh (5)

Vishnu Bose (39)

GUIDE :Mrs. Greeshma M G

Remarks of Guide :

Remarks of Project Coordinators:



October 2015

Department of Computer Science and Engineering
College of Engineering, Pallippuram P O, Cherthala, Alappuzha-688541,
Phone: 0478 2553416, Fax: 0478 2552714
http://www.cectl.ac.in

Contents

Chapter 1

INTRODUCTION

Software development, as we know it today, is a demanding area of business with its fast-changing customer requirements, pressures of an ever shorter timeto- market, and unpredictability of market. With the shift towards modern continuous deployment pipelines, releasing new software versions early and often has become a concrete option also for an ever growing number of practitioners.

Continuous delivery is a software development practice where new features are made available to end users as soon as they have been implemented and tested. In such a setting, a key technical piece of infrastructure is the development pipeline that consists of various tools and databases, where features ow from development to deployment and then further to use.

1.1 Purpose

The objective of the project is to put in place a Continues Integration framework for product development activities of Tech11 Software.This would enable the Tech11 team to rapidly bring a product change or feature to production gaining market advantage. This activities of this project will involve accessing different CI integration approaches and solutions available,identify the feasibility of those solution by doing POCs and demos,fine tune the final solution and set up the CI infrastructures,educate the developers on CI culture.

1.2 Document Conventions

As this document is to be viewed by different class of people, to avoid ambiguity and to keep a standard, it has been documented in ieee format. This document have been organized into different sections and each section have been thoroughly analyzed and is presented here.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for the usage by various level of people, such as from a novice, who is skilled with only basic understanding of a computer to high level programmers. This document is created for the use of various people like developers, project managers, testers, users, documentation writers etc. This document gives a detail on what our proposed system is what does it do, how it is implemented, for what it can be used, advantages over the existing systems. This document also specifies the functional and non functional requirements required by the user.

As we already said, this document is created with the intention of understanding this project by people from different levels of qualification and expertise and of different category. For users they need to understand only the services provided to them, so in the srs , they only need to go through introduction, product functions, user documentation, user interfaces and other non functional requirements. Designers need to go only through the interface requirements, product functions. Thus each category of people needs to proceed only through those section they require. This is achieved through well organizing of this document.

1.4 Product Scope

The scope of the system is to help software developers to ensure new features are made available as soon as the program has been implemented and tested. This product also helps in reducing the time needed to develop a software and also acts a guideline for future software developments

Chapter 2

OVERALL DESCRIPTION

2.1 Product Perspective

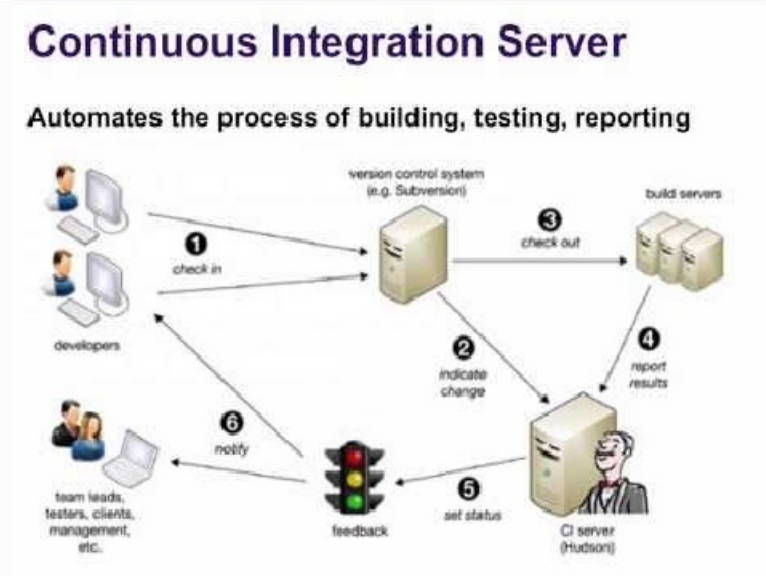
Continuous Integration is based on continuous performance of acts of integration of source code, testing, building and deployment in response to each change to the source code of the project submitted by the developer and for the use of tools for support of the development and testing by compliance with the established procedure automatically. The proposed system directs the user (developers) for time and cost effective production and solves majority of the Integration hell problem.

Integration Hell refers to the point in production when members on a delivery team integrate their individual code. In traditional software development environments, this integration process is rarely smooth and seamless, instead resulting in hours or perhaps days of fixing the code so that it can finally integrate. Continuous Integration (CI) aims to avoid this completely by enabling and encouraging team members to integrate frequently (e.g., hourly, or at least daily).

2.2 Product Functions

The CI frame work continuously monitors each development phase, automates the process of building, testing and reporting.

The developer will push changes into the repository, example GitHub repository. The CI



server some how notified these changes and pull the repository to check if there is any changes in the repository. Then the CI server clone the changes into the server and instruct the build server to check its correctness and build. The developers will be notified via email or other means.

2.3 Operating Environment

The system is designed to work in various working implements.

2.4 Design and Implementation Constraints

Even though the system speeds up the development process, the end users needs to be familiarized with various tools recommended by the system.

2.5 User Documentation

- The system users will be provided with a user manual such that they can be familiarized with the product beforehand.

2.6 Assumptions and Dependencies

The Development environments should support repository plugging (recommended)and a sufficient speed Internet connection.

Chapter 3

EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interfaces

Main Screen:

It will have a login screen. The page will have each category listed clearly, with the selected candidate listed underneath its respective heading. At the very bottom of the page the user should see the ”Go Back” and ”Submit” buttons.

- Welcome screen
- Creating the Voters Database
- Modify the voters databases
- Delete the Voters database
- Creating the Election Instance
- The Voting on the Voters end

3.2 Hardware Interfaces

The additional Infrastructure used is Cloud Hosted Ubuntu system (IAAS). Herouku (PAAS)

3.3 Software Interfaces

- Documentation : LaTeX
- Version Control Framwork : Git,Svn
- Package Managment Framwork : Nexus
- Source Build Framwork : Jenkins
- Build Verification Framwork : PMD,CheckStyle,JUnit
- Review Framework- : Gerrit
- Verification and Review Tracking : BugZilla ,Trac
- Deployment Framwork :Puppet,Chef

Chapter 4

SYSTEM FEATURES

4.1 Frameworks

These are the high level system frameworks that has to be developed for the project

- Version Control Framework
- Package Management Framework
- Source Build Framework
- Build Verification Framework
- Review Framework
- Verification and Review Tracking Framework
- Deployment Framework

4.2 Functional Requirements

The following are the functional requirements of E-Co-Operative

REQ-1: Member Registration

In order to use the system the voters must register to system. This explains the registration process.

Normal flow of events:

1. Member enters the system homepage.
2. He clicks the ”register now” button.
3. The system prompts the application form.
4. He fills in the necessary information related with him in the application form.
5. He sends the request for registration by using ”send” button.

REQ-2: Approve Member

This describe how EA will approve the application form of voter and generate the new account to that voter Precondition.

Normal flow of events:

1. EA selects the online voter application form from list.
2. EA checks the information of the applicant a
3. If the the given information is correct
   (a) EA approves the form by pressing ”Approve” buton.
   (b) EA generates the new online account to this new voter
   (c) EA prepares the username and password send to member via message.
4. if the given information is not correct EA will inform voter about misinformation via message

REQ-3:Update Voters

This describe how EA updates online voters

Normal flow of events:

1. The system checks online voters with respect to upcoming elections voters list.

2. If the voter exists in the list, the system updates the voter with respect to official the voter information.

3. If the voter does not exist in the list, the system deletes that voter from database.

REQ-4:Login/Logout

This describes how the members log into the system

Normal flow of events:

1. The user enters his login id and password
   (a) If the login and password is valid, a session is opened.
   (b) The OTP password as security.
   (c) The specific page of every user is loaded
   (d) If the login or password is not valid, the login screen is redisplayed with an error message.
2. The user click on the logout button.
   (a) The session is terminated.
   (b) The login screen is displayed.

REQ-5: Voting

1. Display Ballot Paper containing candidates names.
2. Click favorable candidates to register vote.

REQ-6: View Election Results

This describes the process of how the voters view the election results by using the system.

Normal flow of events:

1. He clicks on the election results link.
2. He presses click on button ”show results”.
3. The system displays the required information according to the selected choices.

REQ-7: Message Generation

This describes the process of how the messages are send to the members.

1. If the membership is verified by admin, then send username and password to the corresponding members mobile number.
2. Send OTP password as security to the corresponding members mobile number.

Chapter 5

OTHER NONFUNCTIONAL REQUIREMENTS

5.1 Performance Requirements

- System should work smoothly with out lag.
- Devement should not br interrupted.
- System shall support simultaneous users.
- System shall give the user a user friendly interface.

5.2 Safety Requirements

- System use shall not cause any harm to human users.

5.3 Security Requirements

- The Users must have individual as well as group passwords for accessing the system.
- Data handling have to be secured.

5.4 Software Quality Attributes

- Maintainability.
- The system have to continuously updated frequently.

Chapter 6

Glossary

CI	:	Continous Integration
SRS	:	Software Requirements Specification