**Sistema de alarmas para tutelas y derechos de petición de la universidad del Quindío**

**Quality Assurance Plan**

**Version 2.0**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 08/03/2016 | 1.0 | Creación inicial del documento | Daniel Ramiro García  Jhon Alexander Lara |
| 12/04/2016 | 2.0 | Actualización del documento | Jhon Alexander Lara |
|  |  |  |  |
|  |  |  |  |

Confidential <Company Name>, 2014 Page 2 of

**Table of Contents**

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

2. Quality Objectives

3. Management

3.1 Organization

3.2 Tasks and Responsibilities

4. Documentation

5. Standards and Guidelines

6. Metrics

7. Review and Audit Plan

8. Evaluation and Test

9. Problem Resolution and Corrective Action

10. Tools, Techniques, and Methodologies

11. Configuration Management

12. Supplier and Subcontractor Controls

13. Quality Records

14. Training

15. Risk Management

**Quality Assurance Plan**

**1. Introduction**

Para garantizar que en el proyecto habrá un aseguramiento de la calidad, este estará basado en los estándares fijados por la norma internacional ISO 9001 que aplica para los sistemas de gestión de calidad de la adquisición, desarrollo, operación y mantenimiento de software. Identificando temas no solo de tecnología, sino de estructura organizacional, ciclos de vida, procesos, secuencia de las actividades, entre otros.

**1.1 Purpose**

Esta plantilla tiene como propósito el aseguramiento de la calidad, con este documento se supervisara los requisitos propios del proyecto. Además que se definirán los estándares legales con los cuales se garantizara la utilización de las normas de calidad adecuadas para dichos requisitos.

**1.2 Scope**

El alcance del plan de aseguramiento de calidad es llevar un controlar al detalle los requisitos del proyecto, para así dar cumplimiento con el nivel de calidad y confiabilidad que se le desea brindar al cliente respecto al software que se entregara.

**1.3 Definitions, Acronyms, and Abbreviations**

ISO: Organización Internacional para la Estandarización

SGC: Sistema de Gestión de Calidad

SEPA: Software Engineering Process Authority

RUP: Rational Unified Process

Norma ISO 9001: “Es la base del sistema de gestión de la calidad ya que es una norma internacional y que se centra en todos los elementos de administración de calidad con los que una empresa debe contar para tener un sistema efectivo que le permita administrar y mejorar la calidad de sus productos o servicios.”[1].

**1.4 References**

[[1] Que es ISO 9001?, 2011, http://www.normas9000.com/que­es­iso­9000.html.](http://www.normas9000.com/que)

**1.5 Overview**

La visión general del plan de aseguramiento de calidad, es dar a conocer los objetivos, estándares y guías, las mediciones que se tomarán, los planes, evaluaciones, herramientas, metodologías, riesgos, necesarias para que el módulo cuente con la calidad que se ha propuesto inicialmente en el proyecto.

**1. Quality Objectives**

Para cumplir con el plan de aseguramiento de calidad se requiere especificar los requisitos del software a medir y aplicar control de calidad. Los requisitos de software especificados son:

­ Diseñar y dar soporte a una plataforma que asegure que la información de las tutelas y los derechos de peticiones se pueda controlar, manipular y procesar.

­ La funcionalidad de la plataforma sea amigable con el usuario, evitando que la experiencia de usuario sea compleja.

-La realización de auditorías, para dar un correcto seguimiento a el proyecto.

**2. Management**

**1.6 Organization**

La organización se basará en lo recomendado por el RUP, la SEPA, que permite que se inicie y evalúe periódicamente los procesos del proyecto, conociendo a fondo el contexto del proyecto y plan de futuras mejoras por realizar, modificaciones o agregaciones al proyecto. La SEPA puede ser una sola persona o un equipo, que representa una autoridad sobre el proyecto.

**1.7 Tasks and Responsibilities**

­ Basarse en métodos y herramientas donde se puedan analizar, diseñar, programar y realizar pruebas.

­ Contar con una documentación del software desarrollado y las versiones que han sido modificadas.

­ Verificaciones y validación de técnicas en los pasos de los procesos de software.

­ Ajustarse a los estándares de calidad.

­ Tomar atributos de medición del software, controlando el proyecto.

­ Realización de informes y evaluaciones de gestión.

**3. Documentation**

La documentación en la que se hace referencia en el proyecto es:

• Plan de desarrollo de software

• Plan de pruebas

• Planes de iteración

• Requisitos de Software

• Documento de Arquitectura de Software

**4. Standards and Guidelines**

Como normas y directrices del proyecto se tiene:

­ Acompañamiento al cliente, conocimiento de las necesidades y los beneficios de los interesados en el proyecto, casos de desarrollo.

­ Modelamiento de procesos, realización del modelado de negocio.

­ Documentación de todos los procesos realizados, documentación del negocio y documentación del

sistema; interfaz de usuario, modelado de casos de uso, las pautas de diseño, programación de la plataforma, evaluaciones.

**5. Metrics**

Las métricas de calidad del proyecto se basaran en atributos internos como la correcta codificación, las pruebas y el diseño. Además de atributos externos dirigidos al usuario como manuales de uso, soporte y adecuaciones.

### Cumplir con la totalidad de los requisitos funcionales y no funcionales.

Es importante porque cumplir con ellos a cabalidad garantiza el cumplimiento de los objetivos planteados en el proyecto.

Dicha métrica se calificara teniendo en cuenta el avance del proyecto, el cumplimiento de los distintos módulos y la verificación del cumplimiento y no cumplimiento de los distintos requisitos funcionales.

Se verificara cada semana el avance del proyecto y los logros alcanzados para determinar el nivel de cumplimiento, exigiendo eficiencia en el proceso de desarrollo del software y el cumplimiento de actividades.

### Cumplir con la integración entre los módulos del software.

Es importante porque la aplicación debe de garantizar un óptimo funcionamiento y debe de estar bien acoplada.

Para la verificación de esta métrica se utilizara la integración continua para verificar periódicamente el correcto funcionamiento de cada módulo que se va desarrollando y su acoplamiento con el resto de la aplicación.

El servidor de integración generara un build para la verificación de la aplicación cada que se haga un commit en la línea principal de código del repositorio de almacenamiento del proyecto.

### Permitir al usuario interactuar de una forma fácil para cumplir sus necesidades, sin atrofiar el sistema en el que navega.

Es importante que la aplicación sea amigable para el usuario y le permita optimizar su proceso de manera que el manejo del software sea intuitivo.

Para verificar esta métrica se diseñara prototipos a medida que avanza el proyecto para que el usuario interactúe con ellos y determinar la facilidad con la que se desenvuelve en su uso.

Para cumplir esta métrica se segregaran cada uno de los módulos para poder desarrollar una aplicación trasparente y cuyas interfaces no agobien al usuario, permitiéndole determinar en cada pantalla cuál es su uso y finalidad sin confundirlo con herramientas innecesarias.

### Cumplir con la funcionalidad de almacenamiento en cada uno de los usuarios.

Es de vital importancia persistir la información de cada uno de los usuarios, ya que el proyecto gira alrededor la información de estos.

Se debe de monitorear si en cada una de las funcionalidades del proyecto donde se inserte información, en el método de persistencia que se use verificar que la información que se ingrese y quede dentro de este.

Cada vez que se inserte un dato, se verificara el correcto almacenaje de esta.

### Cumplir con un sistema básico, el cual sea de un uso que no sea complejo.

Es muy importante que la aplicación no cuente con una complejidad de uso, ya que le costara mucho al usuario aprender a interactuar de manera correcta con ella. Esto podría generar un descontento de parte del usuario.

Haciendo adecuados manuales de uso, se hará que la experiencia del usuario sea un poco más amigable y acompañada.

Cada vez que se haga una entrega al usuario se verificara que cuente con su debido manual de uso y se dará soporte para garantizar el acompañamiento del usuario.

**6. Review and Audit Plan**

­Tareas. Con cada iteración durante el proyecto se revisarán las tareas realizadas que han sido encargadas a cada uno de los integrantes del grupo de trabajo, dándoles una respectiva observación y sugerencia de mejoras.

­ Horario. Está relacionado con las iteraciones que se realizarán periódicamente en una fecha establecida previamente en el cronograma, sin llegar a incumplir con lo establecido.

­ Organizaciones y responsabilidades. El arquitecto, gerente del proyecto y el cliente serán los que realizarán las auditorías, revisaran el cumplimiento de las tareas y emitirán la opinión sobre los avances o cambios necesarios durante el proyecto.

­ Resolución de problemas y acciones correctivas. La acciones correctivas se referencian en el documento

Problem Resolution Plan.

­ Herramientas, técnicas y metodologías. Esto se revisará según lo descrito en el RUP, haciendo la revisión de requisitos, arquitectura, diseño, configuración fisica, configuración funcional, procesos y revisiones gerenciales, también descritos en el Plan de Evaluación.

**7. Evaluation and Test**

Se evaluaran durante la ejecución del proyecto las métricas, en cada entregable se le dará un porcentaje de evaluación de cada métrica.

Se hará por medio de una lista de chequeo, evaluándola por porcentajes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Porcentaje | Realizado |  |
| Items | 25% | 50% | 75% | 100% |
| 1. Cumplir con la totalidad de los requisitos funcionales y  no funcionales. | | X |  |  |
| 2. Cumplir con la integración entre los módulos del software. |  |  |  | X |
| 3. Permitir al usuario interactuar de una forma fácil para cumplir sus necesidades, sin atrofiar el sistema en el que navega. | | X |  |  |
| 4. Cumplir con la funcionalidad de almacenamiento en  cada uno de los usuarios. | | X |  |  |
| 5. Cumplir con un sistema básico, el cual sea de un uso  que no sea complejo. | | X |  |  |

**8. Problem Resolution and Corrective Action**

**9. Tools, Techniques, and Methodologies**

Las herramientas que se utilizaran para la programación serán el XAMPP, ya que el lenguaje en el que se desarrollará la plataforma será en PHP; project libre para el manejo del cronograma. Como framework se utilizara CodeIgniter y la metodología se basara en artefactos del RUP.

**10. Configuration Management**

**11. Supplier and Subcontractor Controls**

**12. Quality Records**

Con la realización de las revisiones por iteración establecidas en el cronograma, y las observaciones documentadas en el Plan de Evaluación, Plan de Pruebas, en el Plan de Resolución de Problemas, se tomarán las intervenciones necesarias y se hará un seguimiento a los requisitos de calidad determinados desde un comienzo del proyecto.

**13. Training**

El equipo de trabajo deberá tener un conocimiento previo para trabajar en el proyecto, con capacidades para el desarrollo en el lenguaje determinado y capacidades para gestionar todo lo relacionado con la gerencia de proyectos. Además de conocer a cabalidad la aplicación antes construida para dar un buen soporte.

**14. Risk Management**