Universidad ORT Uruguay

Facultad de ingeniería

Obligatorio 1

Entregado como requisito obligatorio para la materia Ingeniería de Software 2

Maximiliano García - 146323

Felipe Cordoves - 175155

Docente: Dr. Gerardo Maturro

2017

# **Abstract**

En este trabajo,

# **Declaratoria de autoría**

*Mediante el presente manifestamos, los escritores firmantes del trabajo, Felipe Cordoves y Maximiliano García, afirman que son los únicos autores del documento*.

Firmas de los autores:

|  |  |
| --- | --- |
| Felipe Cordoves  14-Abr-2017 | Maximiliano García  14-Abr-2017 |

# **Palabras claves**

# **Índice**

[Abstract 2](#_Toc481016994)

[Declaratoria de autoría 3](#_Toc481016995)

[Palabras claves 4](#_Toc481016996)

[Índice 5](#_Toc481016997)

[Plan de Aseguramiento de la Calidad 6](#_Toc481016998)

[Evaluación del Estado de Calidad del Software 10](#_Toc481016999)

[Análisis de Impacto de los Cambios a Realizar 11](#_Toc481017000)

[Implementación de los Cambios 12](#_Toc481017001)

[Evaluación de la Calidad del Resultado Final 13](#_Toc481017002)

[Lecciones aprendidas 14](#_Toc481017003)

[Referencias bibliográficas 15](#_Toc481017004)

# **Plan de Aseguramiento de la Calidad**

Propósito: El propósito de este plan es especificar las actividades que se realizarán para asegurar la calidad del software a mantener. En él se detallan los productos que se van a revisar y los estándares, normas o métodos a aplicar; los métodos y procedimientos que se utilizarán para revisar que la elaboración de las funcionalidades se realice como lo establece el modelo de ciclo de vida del proyecto; y procedimientos para informar a los responsables de los productos los defectos encontrados y realizar un seguimiento de dichos defectos hasta su corrección.

Descripción del producto: El producto es un software de juego llamado “Monopoly”

Objetivos y criterios de calidad:

Uso del plan de calidad por cada uno de los roles:

El modelo de proceso que sigue este proyecto de mantenimiento es el incremental (*Esto no va como actividad preventiva es correcto?*), donde el plan de SQA pretende cubrir las tareas de diseño detallado, construcción y pruebas. Por otra parte, los requerimientos están claramente especificados como parte de la letra del proyecto, y por temas de alcance y costos no sería posible modificar el diseño arquitectónico de la solución.

Actividades Preventivas

* Revisión de a pares, dado que consume poco tiempo y recursos, dos elementos escasos en este proyecto. Se realiza de acuerdo a la siguiente lista de verificación *(ARMAR Y ANEXAR LISTAS, Buscar Herramientas y Técnicas, Cuales Herramientas hay?),* y los resultados se documentan.
* Programación de a pares, optimizando tiempos entre idear la solución y asegurar el cumplimiento de los estándares de codificación y el éxito en las pruebas unitarias.
* Inspecciones (Buscar alguna
  + Dinámicas
  + Manuales
* Seguimiento de un estándar de codificación
* Metodología TDD: el enfoque de entrelazar el diseño de pruebas en conjunto con el desarrollo de la nueva funcionalidad conlleva a ventajas como facilitar en pruebas de regresión automatizadas para conocer los problemas que introduzca todo nuevo código y al mismo tiempo poder analizar el impacto del mismo sobre la funcionalidad ya existente.

Actividades Correctivas

*Definimos mas adelante*

Roles y Responsabilidades

*Tendríamos en este capítulo del documento especificar que roles van a integrar el Plan de SQA y cuáles de las actividades (Preventivas y Correctivas) son de su responsabilidad?*

Plan de Métricas:

Para idear dicho plan utilizamos la técnica GQM (Goal-Question-Metric) que es un paradigma para desarrollar y mantener un significativo programa de métricas que ayudan a:

* Alinear las Métricas con los negocios de la organización y las metas técnicas.
* Mejorar el proceso del software
* Mejorar la calidad del producto

Proporciona una manera útil para definir mediciones tanto del proceso como de los resultados de un proyecto. Considera que un programa de medición puede ser más satisfactorio si es diseñado teniendo en mente las metas (objetivo perseguido). Las preguntas ayudaran a medir si se está alcanzando en forma exitosa la meta definida por esta razón se consideraran preguntas potencialmente medibles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo | Propósito | Asunto | Elemento | Punto de Vista |
| O1 | Mejorar | la usabilidad | de la interfaz gráfica | Del usuario que juega |
| O2 | Mejorar | la calidad | del código | Los Desarrolladores |
| O3 | Generar | un proceso software | proyecto de mantenimiento | Interno al equipo |
|  |  |  |  |  |
|  |  |  |  |  |

*Tendríamos que incluir las 10 heurísticas para poder medir la usabilidad actual o solo enfocarnos en algunas, por ejemplo las que más nos interesarían como diseño de interfaz, reconocimiento en vez de recuerdo, consistencia y estándar de términos, control y libertad del usuario (Ejemplos)?*

|  |  |
| --- | --- |
| **Objetivo GQM - O1** | |
| **Analizar** | Cumplimiento de Heurísticas de Nielsen en UI |
| **Con el propósito de** | Mejorar |
| **Con respecto a** | La interfaz gráfica |
| **Desde el punto de vista de** | Los usuarios que juegan |
| **En el contexto de** |  |
| **Preguntas** | |
| **Pregunta 1** | El diseño de la interfaz sigue algún parámetro o estándar de estética? |
| **Pregunta 2** | El sistema ayuda a los usuarios a reconocer antes que recordar objetos y acciones? |
| **Pregunta 3** | El sistema es consistente y cumple un estándar semántico? |
| **Métricas** | |
| **Pregunta 1** |  |
|  |
|  |
| **Pregunta 2** |  |
|  |
|  |

|  |  |
| --- | --- |
| **Objetivo GQM – O2** | |
| **Analizar** | Cumplimiento de Estándares de Codificación |
| **Con el propósito de** | Mejorar |
| **Con respecto a** | La calidad del código |
| **Desde el punto de vista de** | Los desarrolladores |
| **En el contexto de** |  |
| **Preguntas** | |
| **Pregunta 1** |  |
| **Pregunta 2** |  |
| **Pregunta 3** |  |
| **Métricas** | |
| **Pregunta 1** |  |
|  |
|  |
| **Pregunta 2** |  |
|  |
|  |

Gestión de Cambios

Se considera un cambio toda solicitud que modifique el estado actual del sistema Monopoly, ya sea agregando nueva funcionalidad o corrigiendo defectos detectados o reportados. Para la gestión de versiones de los elementos de configuración de software se usará GIT como repositorio aprovechando las ventajas que tiene esta herramienta de tercera generación en cuanto al manejo distribuido del repositorio, además de ser gratis.

Objetivos de la gestión del cambio:

* Revisar, analizar y aprobar las solicitudes de cambio
* Gestionar los cambios aprobados
* Mantener la integridad de la línea base, incorporando solo los cambios aprobados
* Revisar, aprobar, rechazar todas las acciones preventivas y correctivas recomendadas
* Coordinar los cambios de acuerdo al ciclo de vida
* Documentar el impacto de los cambios

Gestión de Defectos

Para el registro de solicitudes y defectos se utiliza GitHub al estar ya integrada al manejo del repositorio. Descartamos otras herramientas como mantis, redmine o bugzilla ya que requería un tiempo mayor de investigación y aprendizaje.

*\* Duda: Las solicitudes que ya se hicieron en la letra también las registramos como Issues en la herramienta? Cuando entreguemos el obligatorio tenemos que entregar algunos printscreen de lista de Issues?*

*\* Alcanza para estos puntos de Gestión de Defectos y Gestión de Cambios el marco teórico que le dimos y la justificación de que herramientas estamos usando o hace falta algo más?*

# **Evaluación del Estado de Calidad del Software**

# **Análisis de Impacto de los Cambios a Realizar**

# **Implementación de los Cambios**

# **Evaluación de la Calidad del Resultado Final**

# **Lecciones aprendidas**

# **Referencias bibliográficas**