

Jogo Rubynho

Informe de Revisión Técnica Formal (RTF)

Versión 1.0

Historia de revisiones

Fecha	Versión	Descripción	Autor
19/09/2014	1.0	Creación del documento	Mauricio Calcagno, Fernando Banchemo

Contenido

- Producto revisado.....3**
 - Nombre y versión del producto revisado.....3**
 - Participantes de la revisión.....3**
 - Técnica utilizada.....3**
- Objetivos de la RTF.....3**
- Problemas detectados.....3**
- Evaluación.....5**
 - Estado actual del producto.....5**
 - Acciones a tomar.....5**
 - Próxima revisión del producto.....5**

1. Producto revisado

1.1. Nombre y Versión del Producto revisado

Se revisó el documento de descripción de la arquitectura versión 1.0 de la línea de trabajo diseño.

1.2. Participantes de la revisión

Fernando Banchemo – Responsable de SQA, Asistente de Verificación

Mauricio Calcagno – Arquitecto

1.3. Técnica utilizada

La técnica utilizada para la realización de la revisión corresponde a la de listas de comprobación. En particular se utilizó una checklist correspondiente a aspectos arquitectónicos que aparece en el Plan de Calidad.

2. Objetivos de la RTF

El objetivo es verificar la arquitectura descripta, y en particular verificar que para la elaboración de la misma se haya tomado en cuenta las historias más importantes y que se mitiguen los riesgos de mayor importancia.

3. Problemas detectados

¿La organización total del programa está clara e incluye una buena vista de la arquitectura y su justificación?

- Medianamente, la arquitectura está avanzada pero sigue sufriendo cambios a medida que se avanza en el desarrollo y análisis de las historias de usuario. La única documentación existente es el documento Descripción de Arquitectura que no está actualizado.

¿La arquitectura está compuesta de capas?

- No directamente, la arquitectura está basada en un diseño MVC y no separa más en capas de lo que hace el MVC y el framework (Ruby on Rails).

¿Los paquetes están bien definidos, incluyen su funcionalidad e interfaz para otros módulos?

- No todos, se está definiendo informalmente las interfaces con el simulador.

¿Todas las estructuras de datos importantes están descriptas y justificadas?

- Hay un punto de arranque en todas y a medida que son necesarias de implementar cuando entran al backlog se modifican. No existe documentación.

¿Todas las estructuras de datos importantes están ocultas mediante funciones?

- Nada más allá de lo que de por sí provee el framework, muchas no han entrado al backlog.

¿Todas las estructuras de datos importantes están ocultas mediante funciones de acceso?

- Nada más allá de lo que de por sí provee el framework, muchas no han entrado al backlog

¿Todos los algoritmos principales están descriptos y justificados?

- Solamente en parte el de simulación. No existe documentación

¿Todos los objetos importantes están descriptos y justificados?

- Hay un punto de arranque para todos y a medida que se implementen al entrar en el backlog se modifican. No existe documentación

¿La interfaz con el usuario está modularizada tal que los cambios en ella no afecten el resto del programa?

- Nada más allá del MVC

¿Los aspectos claves de la interfaz de usuario están definidos?

- No, en negociación con el cliente

¿Se ha estimado el uso de memoria, se ha descrito y justificado una estrategia para la gestión de la memoria?

- No

¿Se ha incluido una estrategia coherente de manejo de errores?

- No

¿Se ha definido un nivel de robustez?

- No

¿La arquitectura está diseñada para acomodar cambios?

- Si

¿Los objetivos principales del sistema están claramente establecidos?

- Medianamente

¿Identificó las historias de usuario y actores más significantes para definir la Arquitectura del sistema (tomando en cuenta los más importantes para el cliente y que mitiguen los riesgos más serios)?

- Si

¿Realizó un diagrama mostrando la descomposición del Modelo de Diseño en Subsistemas, sus interfaces y las dependencias entre ellos, mostrando para cada Subsistema la Capa de la Arquitectura en la que se encuentra?

- No

¿Identificó las Clases de Diseño claves del Sistema?

- Si

¿Describió las clases del Diseño y los subsistemas que corresponden a la Arquitectura del Diseño de cada Historia?

- No

¿Especificó cómo están organizados los componentes de acuerdo a la estructura y mecanismos de modularización disponibles en el ambiente de implementación y en el(los) lenguaje(s) de programación, y cuáles son las dependencias entre los distintos componentes?

- Medianamente

¿Muestra la correspondencia entre la Arquitectura del Software y la Arquitectura del Hardware?

- No

¿Se tienen en cuenta todas las propiedades de calidad que debe tener el sistema?

- Medianamente

¿Se ha alcanzado un grado adecuado de modularidad?

- Se ha alcanzado un grado aceptable

¿Se ha diseñado para el cambio?

- Si

4. Evaluación

4.1. Estado actual del Producto

La arquitectura aun no está definida completamente, esta seguirá modificándose. El trabajo sobre ella se ha centrado en lo necesario para la construcción de las historias a realizar en este sprint.

4.2. Acciones a tomar

Se tomaran en cuenta los resultados de la checklist para continuar con la construcción del producto.

4.3. Próxima Revisión del Producto

La próxima revisión del producto se realizará al final de este sprint.