Interpool

Estimaciones y Mediciones

Versión 6.0

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 15/09/10 | 6.0 | Inicio | Juan Ghiringhelli |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contenido

[1. Estimaciones y Mediciones de Esfuerzo del Proyecto 3](#_Toc272414480)

[1.1. Mediciones de Datos históricos utilizados 3](#_Toc272414481)

[1.1.1. Total de Horas por Línea de Trabajo para grupos de años anteriores 3](#_Toc272414482)

[1.2. Estimación reconciliada de Esfuerzo 3](#_Toc272414483)

[1.3. Estimación según juicio de expertos: 4](#_Toc272414484)

[2. Estimaciones y Mediciones de Tamaño del producto a desarrollar 7](#_Toc272414485)

[2.1. LOC (Lines of Code) 7](#_Toc272414486)

[Mediciones de Tamaño 7](#_Toc272414487)

[2.1.1 Fase de Elaboración – Iteración II - Histórico 7](#_Toc272414488)

[Fase de Elaboración – Iteración II – Nuestro grupo 7](#_Toc272414489)

[2.1.1 Fase de Transición – Iteración I – Histórico 7](#_Toc272414490)

[Fase de Transición – Iteración I – Nuestro proyecto (Estimado) 7](#_Toc272414491)

[3. Resumen y Conclusiones 8](#_Toc272414492)

1. Estimaciones y Mediciones de Esfuerzo del Proyecto
   1. Mediciones de Datos históricos utilizados

Usamos los históricos del equipo 2 del 2009, que realizó un juego para Windows Mobile 6.5 con el mismo cliente, Marcelo Guerra. Utilizamos la cantidad de líneas de código, a cotejar contra la cantidad a medida que vayamos construyendo el producto, y la cantidad de horas por rol y línea de trabajo. Decidimos esto ya que las similitudes de proyecto son muchas, y ese equipo hizo un gran trabajo de rastreo de esfuerzo.

* + 1. Total de Horas por Línea de Trabajo para grupos de años anteriores

Mediciones de Esfuerzo (horas).

Modificamos esta estimación, ya que resultaba redundante.

* + - 1. Horas por fase histórico/interpool

Esto indica que al menos en la primera fase tuvimos un esfuerzo casi idéntico al del grupo del año pasado.

* 1. Estimación reconciliada de Esfuerzo

La técnica de suponer que la carga horaria por áreas se mantiene no tiene sentido, en este momento tenemos 0 horas de implementación y muchas de análisis, eso cambiará con seguridad en el futuro. Estimamos si la cantidad en bruto de horas.

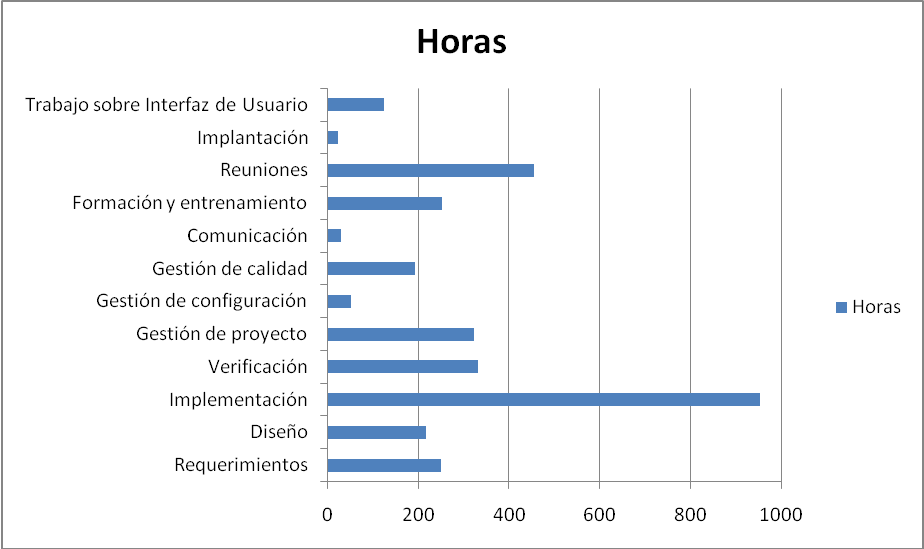
Carga horaria estimada:

Si ponemos 12\*15\*2 (integrantes \* dedicación semanal \* semanas) da 180, cercano a las 200 que registramos.

200\*14 = 2800.

Respecto al trabajo registrado en los históricos:





Por lo que el grupo anterior terminó necesitando 3200 horas de esfuerzo. Viendo el promedio final de horas es razonable, ya que todos pasaban las 20, y algunos las 25. Por ahora conservaremos este estimado, ya que es bastante similar.

Cotejado con nuestras horas y con el MUM, al final de la iteración deberíamos alcanzar un nivel similar de horas de trabajo sobre la interfaz, en formación y entrenamiento, en diseño y requerimientos. En posteriores iteraciones seguiremos ajustando.

* 1. Estimación según juicio de expertos:



Esta tabla es un digerido de la lista de tareas, por casos de uso.

El trabajo total bruto es 120 + 491, = 611. Multiplicando por los porcentajes de completados, tenemos 390 horas de trabajo necesario, 221 horas de trabajo hecho.

Si ajustamos en +20% verificación + %20 buffer + %10 riesgos, tenemos (120+491)\*1,5 = 916,5 horas total, 585 horas necesarias de trabajo, 331 horas trabajadas.

Este estimado es sobre lo mínimo que deberíamos tener armado para el juego según lo definido en el alcance.

Si vemos que tenemos 7 implementadores, que dedican un promedio de 12 horas de las 15 por semana, y que empezaron a implementar los casos de uso en la fase de elaboración ( en la fase inicial se construyó el prototipo para ver la factibilidad y la gran mayoría de las funciones estaban hard-codeadas, pero la segunda iteración puede considerarse para calcular ya que se hizo mucho trabajo reflejado en los porcentajes de completados ), nos quedan 10 semanas de implementación de las fases, más una de implantación que vamos a tomar, dadas las características del proyecto. 7 implementadores \* 11 semanas \* 12, nos da 924. Tenemos una desviación de -0.99 %, o sea, tenemos un muy pequeño margen de sobre esfuerzo según las estimaciones. Si estimamos bien, podemos terminar el proyecto según el alcance definido, sin implementar nada más de los requerimientos opcionales. Llevamos cuatro semanas de trabajo se estas 11 semanas, 4/11 = 0,36. 924 \* 0,36 = 332 horas trabajadas. Coincide casi perfectamente lo que estimamos de trabajo hecho con lo que deberíamos haber hecho en este tiempo. Como estas estimaciones, tanto el porcentaje de completado como la cantidad de horas de trabajo son subjetivas, estamos sujetos a errores en ambos sentidos, pero al menos la estimación está bastante acotada.

1. Estimaciones y Mediciones de Tamaño del producto a desarrollar

No haremos puntos de función, el juego solo tiene dos interfaces, y el verdadero esfuerzo está en los algoritmos que van a correr sobre la “nube” del data warehouse. Otro punto por el cual no los utilizaremos es debido a que la complejidad, la usabilidad y la atractividad de la interfaz de usuario, tampoco se ven reflejados en los puntos de función. Si haremos estimación por líneas de código, basándonos en el prototipo, en una reunión con los especialistas técnicos. En la misma reunión estimaremos esfuerzo en horas de construcción, por medio de un listado de actividades necesarias para la construcción.

* 1. LOC (Lines of Code)

Mediciones de Tamaño

Usaremos los datos históricos e iremos cotejando con una regla de tres según el momento en que nos encontremos y el tamaño final. El histórico tiene registros en la iteración 2 de la fase de elaboración, en la iteración 1 de la fase de construcción, y al final de la fase de implantación. Estimaremos en este documento según el momento que estamos e iremos re estimado a medida que lleguemos a los demás.

2.1.1 Fase de Elaboración – Iteración II - Histórico

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cantidad de líneas de código | Cantidad de archivos de código | Líneas de código auto generado | Número de líneas en blanco | Cantidad de comentarios |
| 6.819 | 64 | 3.918 | 345 | 232 |

Fase de Elaboración – Iteración II – Nuestro grupo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cantidad de líneas de código | Cantidad de archivos de código | Líneas de código auto generado | Número de líneas en blanco | Cantidad de comentarios |
|  |  |  |  |  |

* + 1. Fase de Transición – Iteración I – Histórico

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cantidad de líneas de código | Cantidad de archivos de código | Líneas de código auto generado | Número de líneas en blanco | Cantidad de comentarios |
| 32.045 | 191 | 11.200 | 3.010 | 2.968 |

Fase de Transición – Iteración I – Nuestro proyecto (Estimado)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cantidad de líneas de código | Cantidad de archivos de código | Líneas de código auto generado | Número de líneas en blanco | Cantidad de comentarios |
|  |  |  |  |  |

1. Resumen y Conclusiones

En resumen, esperamos que en total el proyecto nos lleve entre el esfuerzo del equipo 2 del año 2009, 3200 horas, y el estimado por trabajo sostenido, 2800 horas. O sea unas 3000 horas. Respecto al esfuerzo de construcción en código, llevará aprox. 1000 horas, según lo estimado venimos bien en los requerimientos construidos según las semanas trabajadas, y trabajando lo mismo de aquí al fin de la fase de construcción, expandiéndola una semana más tomada de la fase de implantación, llegaríamos bien.

Las líneas de código debería estar entre 32.000 líneas de código y ( lo lleno el sábado ).

La gráfica que muestra el esfuerzo aproximado de horas por fase indica que estamos haciendo un esfuerzo similar al proyecto que tomamos de base.