Retrospectiva

Sistema de Diagnóstico y Agendamiento de Citas (SDAC)

Versión 1

Jairo Daniel Bautista Castro
Miguel de Oliveira Dias Gonçalves

Índice

Línea de tiempo del proyecto	2
Plantilla de retrospectiva	
El proyecto en general	
Lo que salió bien	
·	
Lo que hay que mejorar	
Lo que hacer a continuación	6

Línea de tiempo del proyecto

- 1 de Octubre Inicio de las clases de la asignatura de PGPI
- 9 de Noviembre Entrega de la Práctica 1
- 12 de Noviembre Entrega de las Prácticas 2 y 3
- 18 de Noviembre Entrega de la Práctica 4
- 2 de Diciembre Entrega de las Prácticas 5 y 6
- 23 de Diciembre Entrega de las Prácticas 7 y 8 / Conclusión del proyecto

Plantilla de retrospectiva

Para esta retrospectiva final del proyecto SDAC, vamos a seleccionar un modelo de retrospectiva utilizado en las metodologías ágiles. Este modelo tiene 3 aspectos principales, además del propio proyecto en general:

- Lo que salió bien
- Lo que hay que mejorar
- Lo que hacer a continuación

El proyecto en general

¿Cuál era el plan al comienzo del proyecto? ¿Cómo cambió a lo largo de su ejecución?

El objetivo del proyecto se definió al principio como el gestionar la creación e implementación de un sistema de diagnóstico y agendamiento de citas destinado al Hospital PTS y a sus usuarios, integrado en sus sistemas informáticos. Ese objetivo se mantuvo en el resto de las prácticas.

Se planeó al principio que el proyecto se desarrollaría en un tiempo estimado de 6 meses hasta la entrega el MVP (Minimum Viable Product) al cliente. Sin embargo al estimar con el modelo de COCOMO II nos dio un resultado de 2 años.

¿Qué herramientas se utilizaron durante el proyecto?

Se contempló utilizar mayormente herramientas de desarrollo libre:

- Para el back-end se eligió Node.js, junto con el framework Express y el IDE Visual Studio Code
- 2. Para la capa de base de datos se eligió la base de datos no relacional MongoDB y el IDE Robo 3T

3. Para el front-end se eligió el framework Angular 5 y también el IDE Visual Studio Code

- 4. Como herramienta de despliegue se eligió Jenkins
- 5. Para automatizar la infraestructura se eligió Chef
- 6. Para el rastreo de errores se eligió Bugzilla
- 7. Para el control de versiones no solo se eligió sino también se utilizó en las prácticas GitHub
- 8. Para el manejo de paquetes se eligió Gradle
- Para la gestión del proyecto se utilizó Microsoft Project y Project Libre
- 10. Para la estimación del esfuerzo de ejecución se utilizó principalmente COCOMO

¿Cuál fue el proceso utilizado a lo largo del proyecto?

Se utilizó un ciclo iterativo e incremental, basado en las metodologías ágiles como Scrum y Kanban.

¿Cuáles son los aspectos más relevantes de este proyecto que resaltarían para compartirlos con el gestor del proyecto y el resto de su equipo de cara a proyectos futuros?

Los aspectos más relevantes son tanto el tiempo de ejecución como el presupuesto previstos al final de las prácticas y su diferencia comparando con el tiempo de ejecución y presupuesto previstos inicialmente.

¿Está satisfecho con los resultados del proyecto? Si lo está, ¿qué fue lo mejor? Y si no lo está, ¿en qué se puede mejorar?

No, pues en la planeación no se debió utilizar modelos como COCOMO II para calcular el tiempo de ejecución del proyecto, dado que esos modelos utilizan medidas como el número de SLOC para calcular el tiempo de desarrollo, las cuales no son fiables con lenguajes modernos.

¿Qué sabe ahora que le hubiese gustado saber al comienzo del proyecto? ¿Cómo habría cambiado el proyecto de haberlo sabido antes?

Nos gustaría haber sabido que tendríamos que utilizar modelos como COCOMO II más adelante en el proyecto, para que pudiéramos utilizarlos al inicio y así reducir la enorme diferencia (las predicciones de COCOMO II son 4 veces más grandes que nuestras predicciones iniciales) entre nuestras predicciones iniciales y las predicciones de dichos modelos, que posteriormente aceptamos para el proyecto.

¿Qué decisiones importantes se tomaron a lo largo del proyecto?

La decisión más importante y con impacto más grande tomada al largo del proyecto fue cambiar nuestra estimación de esfuerzo para la obtenida utilizando el modelo COCOMO II, la cual resultó en una estimación cerca de 4 veces superior a la inicial además de traer un gran aumento tanto al tiempo de ejecución del proyecto como también al presupuesto.

Lo que salió bien

¿Qué aspectos fueron especialmente bien durante el proyecto? ¿Por qué?

Creemos que dividir el trabajo en tres células ágiles para trabajar en paralelo desde un principio ayudará a realizar el trabajo en paralelo y mejorar la eficiencia.

Sobre la realización de las prácticas, creemos que las reuniones entre los miembros del grupo fueron uno de los aspectos que mejor funcionó, por haber sido donde se realizó la mayor parte del trabajo.

De las herramientas que se utilizaron en el proyecto, ¿Para usted cuales funcionaron bien?

En las prácticas, la herramienta GitHub fue bastante útil para nosotros al permitir no solo almacenar sino también compartir el trabajo.

Las herramientas de gestión de proyectos fueron también útiles, al permitirnos no solo hacer una calendarización y estimación de costes adecuadas del proyecto sino también estimar nuevas después de los cambios en la estimación de esfuerzo del proyecto derivados del uso del COCOMO II.

Sobre el proceso utilizado a lo largo del proyecto, ¿En qué aspectos funcionó razonablemente bien?

Las reuniones de nuestro grupo fueron positivas, ya que no solo eran frecuentes sino servían también para compartir ideas para el proyecto, además de corregirse y mejorar nuestro trabajo previo.

Lo que hay que mejorar

¿Qué aspectos salieron rematadamente mal durante el proyecto? ¿Por qué?

La aplicación de métodos tradicionales de estimación de esfuerzo como el COCOMO II o Modelo de Putnam para planear un proyecto ágil ha tenido un impacto negativo en el proyecto, pues ha aumentado inmensamente su

esfuerzo estimado y por ende sus costes. Tanto las herramientas como los lenguajes empleados son recientes, y por eso el número de SLOC tiene poca relevancia como medida de esfuerzo.

¿Cuáles son los cuellos de botella que usted considera que existieron en este proyecto?

Creemos que el cuello de botella más grande fue nuestra falta de tiempo debido a la cantidad de esfuerzo exigida en otros proyectos que teníamos aparte. Juntos fueron factores determinantes en la calidad del proyecto desarrollado.

¿Qué fases o etapas del proyecto habrían necesitado más tiempo para poder ejecutarse de una forma más adecuada, teniendo en cuenta las restricciones temporales ya impuestas?

De haber alguna, creemos que sería la fase de desarrollo y la fase de pruebas al sistema.

En las prácticas, ninguna fue particularmente larga temporalmente y el posponer de algunas fechas también nos ha ayudado.

¿En qué fases se tuvieron que repetir tareas ya realizadas (i.e. "rework" considerado innecesario)? ¿Cómo podría haberse evitado?

Creemos que esa situación no ocurrió mucho durante las prácticas. Es verdad que tanto la estimación con modelos como posteriormente el análisis de los riesgos hicieron cambiar la planificación del proyecto, pero el trabajo realizado con la planificación inicial del proyecto no se repitió, al menos totalmente, dado que no fue necesario planificar en detalle los meses adicionales sino sobre todo decir cuantos eran.

De las herramientas que se utilizaron en el proyecto, ¿para usted cuales funcionaron mal?

Creemos que el Modelo de Putnam, uno de los modelos utilizados para predecir el esfuerzo del proyecto, funcionó mal al estimar el esfuerzo del proyecto en más de 15.000 personas-mes, mientras las otras herramientas utilizadas lo estimaron en menos de 400 personas-mes. Creemos que eso se debe al reducido número de parámetros del modelo.

Sobre el proceso utilizado a lo largo del proyecto, ¿qué aspectos del proceso le parece que hay que cambiar?

Creemos que el proceso se ha aplicado de una forma correcta. Hacer más retrospectivas y tenerlas en cuenta posteriormente parece ser el cambio más grande a implementar, en estas prácticas se hizo menos.

¿Qué le quita a usted el sueño por la noche? ¿Qué se quedó sin resolución durante el proyecto?

El presupuesto es la parte que nos quita más el sueño, ya que nos dio un valor demasiado alto (más de 2 millones de euros después de tenerse en cuenta el análisis de los riesgos).

Las fases del desarrollo del proyecto fueron cubiertas en la planificación inicial del proyecto, desde el reclutamiento del equipo hasta la retrospectiva final.

Sobre las prácticas, todas fueron entregadas a tiempo.

Lo que hacer a continuación

De las herramientas que se usaron en el proyecto, ¿Cómo cambiaría el uso de herramientas de cara a proyectos futuros?

Evitaremos el uso del Modelo de Putnam, dado que sus estimaciones fueron demasiado altas. También, si es necesario utilizar el COCOMO II de nuevo, entonces habrá que usarlo en la planificación inicial del proyecto para que no existan discrepancias tan grandes entre la planificación inicial v la de COCOMO II.

Según su opinión, ¿Cuáles serían las mejores estrategias para disminuir los cuellos de botella?

Escoger menos proyectos para su realización en paralelo con este proyecto, ya que los demás proyectos consumieron mucho de nuestro tiempo.

¿Qué lección ha sido aprendida en el transcurso del desarrollo del proyecto?

Este proyecto nos ha servido como una experiencia de planificación y gestión de proyectos informáticos. A diferencia de otras asignaturas, hemos estado en el lugar de gestores de un proyecto en vez de programadores del mismo, y de esa forma hemos vivido una nueva experiencia que ciertamente nos vendrá útil en un futuro en el cual tengamos un rol de gestores en un proyecto informático.

Una de las lecciones aprendidas fue que metodologías como COCOMO II y Modelo de Putnam para planificar un proyecto con lenguajes modernos como Javascript o Java no son la mejor elección. La cantidad de líneas de

código no es una medida fiable para concluir el esfuerzo de un programador. Por ejemplo, para aplicar un patrón de diseño la mayoría de las veces se reduce el número de líneas de código del proyecto, pero esto se torna mucho más difícil de hacer y lleva más tiempo que hacer solo una clase con más líneas de código. Sin embargo, si aplicamos los modelos como el COCOMO II, la clase que tiene más líneas de código llevaría mucho más trabajo que aplicar un patrón de diseño, cosa que no es verdad.

¿Que podríamos hacer mejor en cada entrega?

Hacer una retrospectiva de la práctica o de las prácticas anteriores, y tener en cuenta los resultados de esa retrospectiva en las prácticas siguientes.