



UNIVERSIDAD DE GRANADA

MANUAL DE COORDINACIÓN - VERSIÓN 2

Autores

Juan Manuel Castillo Nievas
Víctor Torres de la Torre



MÁSTER PROFESIONAL EN INGENIERÍA INFORMÁTICA

Granada, 16 de noviembre de 2020

Servicio de mantenimiento del Sistema de Información de Unidad de Cuidados críticos en el Hospital Universitario Regional de Málaga

1. Ciclo de vida

El proyecto tendrá un ciclo de vida iterativo. Esta decisión ha sido tomada en base a sus notables beneficios. En primer lugar nos proporciona un *feedback* muy potente con el Hospital Universitario, ya que implica una gran retroalimentación en todo el ciclo de vida del proyecto. También nos permite una mejor gestión de riesgo, asumiendo con menos impacto los posibles cambios imprevistos. Por otro lado se adapta perfectamente a la metodología que será descrita en la Sección 2.

Con esta decisión se pretende incentivar la orientación al cliente.

2. Metodología desarrollo

Se ha decidido proponer la metodología **Scrum** como metodología de desarrollo debido a su gran potencial. Primero destacar que **Scrum** nos ayudará a reducir tiempo y costes gracias al sistema de planificación de tareas que nos brinda. Debemos tener en cuenta que **Scrum** es una metodología adaptativa que nos permitirá adecuarla a las necesidades del proyecto en cuestión. Por último, destacar que es una metodología de manejo simple que nos ayudará a ser eficientes en cuanto al desarrollo se refiere.

3. Recursos software desarrollo - Se añade el lenguaje de programación

Como lenguaje de programación se va a usar **Python** debido a sus conocimientos y experiencia en el desarrollo de interfaces usando dicho lenguaje.

Como entorno de desarrollo se va a usar el editor **Sublime Text** debido a su gran versatilidad para adaptarse a cualquier tipo de proyecto, y que además incluye diversos atajos y corrección sintáctica de **Python**. Ofrece una gran cantidad de paquetes para ofrecer un flujo de trabajo que se adapta a las necesidades de trabajo de cada persona, y que además es muy personalizable a los atajos de teclado que use cada uno.

Para las herramientas de dibujo se va a usar **Visual Paradigm** por la experiencia que se ha tenido con este programa a lo largo de los años de carrera y por su facilidad de uso. Se pueden hacer diagramas de todo tipo, incluyendo plantillas con las que el usuario puede empezar a realizar los diagramas sin necesidad de crear un diagrama completamente desde el inicio.

Para la documentación se va a usar **Google Docs** de forma que se pueda hacer una documentación totalmente colaborativa en el equipo de trabajo. Este documento será una especie de borrador, puesto que una vez redactado todo el documento se usará **Overleaf** para pasar todo lo redactado a \LaTeX , y de esa forma ofrecer una presentación mucho más limpia y ordenada.

4. Organización del equipo de trabajo

Al ser un equipo de solamente tres personas, se necesitaría la contratación de más personal (no más de tres) para las tareas que impliquen la programación de funcionalidades. También se podría necesitar la contratación de un traductor para la posible traducción de los documentos a inglés y las reuniones en caso de que en este proyecto se cuente con personas que no dominen el español.

5. Herramientas para comunicaciones en el equipo de trabajo

Se va a usar **Slack** y **Outlook** como medios principales de comunicación. **Slack** es una muy buena opción para trabajar en equipo y además mantener al tanto de todo lo que se ha avanzado a los clientes.

Como opción de reserva se va a tener en cuenta **Zoom** y **Telegram**.

6. Relaciones con el cliente

El cliente puede contactar con nosotros en cualquier momento si hay alguna pregunta abierta o si quiere preguntar sobre el estado del proyecto. Aparte de eso, el plan va a llevar a cabo los siguientes puntos:

- Entrevista con el cliente antes de la licitación.
- Reunión con el cliente después de la aprobación del proyecto y visita *in situ* al sitio (reunión **kick off**).
- Intercambio continuo con el cliente para satisfacer todos sus requerimientos.
- Reunión con el cliente para revisiones del proyecto después de su finalización.

7. Estándares de documentación - Se añaden estándares de documentación

Para los requisitos, en su documentación se va a seguir el estándar **IEEE830**. Habrá un índice inicial en el que se mostrarán todos los puntos que abarca toda la documentación. Los márgenes laterales van a ser un poco más grandes de los habituales en un documento.

Para la documentación y gestión del *software*, se va a seguir el estándar **SO/IEC 18019-2004**.

Para el manual de usuario, se va a documentar siguiendo una estructura por puntos en el que cada punto abarcará una tarea de la interfaz. En cada punto se aportarán imágenes explicativas.

Todo lo que se haga se va a documentar. La documentación reflejará el estado actual del proyecto

en cualquier momento. Al final del proyecto se entregará la documentación en duplicado.

8. Estándares de código - Cambiado el punto de comentarios

- El código sigue altos estándares y siempre se prestará atención a los códigos limpios y la tabulación siempre correcta.
- Se siguen las mejores prácticas para identificadores.
- Convención de nombres: los identificadores serán siempre en español.
- Comentarios: se usará **pydoc** que genera una documentación **HTML** con los comentarios de cada función incrustados en el código.
- Para todo el código habrá una revisión por pares para asegurar la buena calidad.

9. Control de versiones

Como sistema de control de versiones (VCS) usaremos **Git** y **Github** por la experiencia que se tiene en su uso.

Como ya se ha comentado anteriormente, nuestro método incluye el sistema de revisión por pares, por lo que cualquier actualización de la versión publicada debe ser revisada primero por un socio.

10. Gestión de calidad - Cambiado

Se realizarán validaciones cruzadas para asegurar que el código sigue los estándares indicados y poder detectar fallos en códigos ajenos.

Se realizarán reuniones propias de la metodología de trabajo en las que se incluye el seguimiento del desarrollo y cada uno de los detalles del proyecto.

Por otro lado, se van a crear nuevas pruebas durante todo el proceso. Las pruebas se realizan automáticamente y así se podrá comprobar la calidad del proyecto en cualquier momento. Como herramienta de pruebas se usará **Travis CI**, debido a la experiencia que se tiene en su uso.