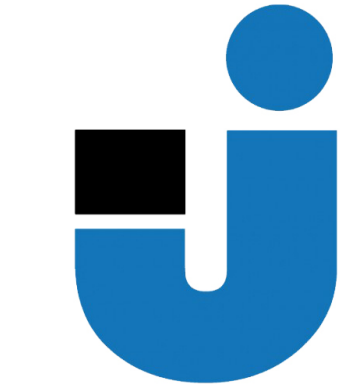

INGENIERÍA DE SOFTWARE I

TRABAJO PRÁCTICO Nº 1

Introducción y Conceptos

Autor
Emiliano SALVATORI

Profesor
Dr. Sergio Daniel CONDE

7 de septiembre de 2020

Índice

1. Introducción

En el siguiente informe se detalla lo realizado como parte introductoria de la materia **Ingeniería de Software I** para la **Comisión nº 1**. El presente trabajo se basa en los conceptos fundamentales acerca de la Ingeniería de Software.

2. Cuestionario

2.1. ¿Qué no es el Software?. Defina en función de los conceptos vistos en clase, lo que NO podría definirse como una pieza de software

Aquello que no es software, sería lo que no estaría alcanzado por la definición que propone por ejemplo, Roger S. Pressman, es decir, aquellos productos que no son realizados por ingenieros de software, ni tampoco pensados ni documentados para ser ejecutados dentro de una computadora, Por ejemplo, la metodología Kanban no es un software, si bien es empleado por los desarrolladores para poder gestionar de forma armónica y transparente la producción, no está alcanzado por la definición provista por Pressman [2].

2.2. Realice una definición de Ingeniería del Software y justifique su respuesta

La ingeniería de software se puede definir como un conjunto interdisciplinario que brinda los conocimientos técnicos y teóricos para poder producir y mantener tanto programas que se ejecutan dentro de sistemas de computo, como su correspondiente documentación. No sólo se deben producir el software que permita el funcionamiento requerido, sino también es necesario mantenerlo durante todo su ciclo de vida, con sus modificaciones y agregados. Asimismo, es necesario también dar respaldo de este producto, suministrando documentación que explique sus funcionalidades, sus modos de operar, y sus articulaciones con otros módulos y programas.

2.3. Defina los siguientes conceptos justificando cada una de las respuestas: Proceso de desarrollo de software y Ciclo de vida del software

Tanto el proceso de Desarrollo como el Ciclo de Vida del software, se encuentran emparentados, íntimamente ligados. El primero es la forma que adopta tanto el diseño como su construcción, que suele dividirse en diferentes tareas, tales como la planificación, el análisis de requerimientos, el testeo de sus funcionalidades, etc. Para todas ellas existen diferentes modelos que permiten llevar a cabo un proyecto de software, como por ejemplo las metodologías ágiles o la producción en cascada (o algunas veces

llamado *Ciclo de vida clásico*[2]). En general, se puede definir como las distintas fases por los cuales pasa un producto de software, definido como una solución a una problemática del mundo real, y que concluye con la puesta en producción de la solución.

Algunas de las etapas pueden definirse como:

- **Modelaje**
- **Desarrollo**
- **Pruebas**
- **Depurado**
- **Diseño**

En cambio, a todo el conjunto de etapas que conforman la totalidad del proceso de desarrollo se lo conoce como *ciclo de vida del software*, etapas que pueden ser definidas como:

- **Inicio**
- **Planificación**
- **Seguimiento**
- **Cierre**

2.4. Defina detalladamente por intermedio de una tabla las características del Software

Las características del Software se pueden visualizar en el siguiente cuadro:

Características	Definición
Mantenibilidad	Es la facilidad con la que el software existente puede corregirse, adaptarse o aumentarse.
Confiabilidad	Es la probabilidad que tiene un programa de cómputo de operar sin fallas en un ambiente específico por un tiempo específico. Esto está asociado con la opción de recuperación que tenga el usuario ante una falla inesperada.
Eficiencia	Su construcción debe ser de tal manera que no desgaste los recursos del sistema como memoria o procesador. Asociada a la idea de respuesta ante las exigencias del usuario y los recursos del sistema.
Usabilidad	El uso del software debe ser de fácil acceso para el usuario final para el que fue diseñado. Si bien el sistema puede ser amigable, cuanto más amigable es al usuario final, más complejo se vuelve mantenerlo [3]
Corrección	Se necesita cumplir con el objetivo principal para el que fue diseñado y desarrollado, en base a las especificaciones diagramadas por el cliente.

2.5. Elija un Software y analice detalladamente las características del software, justificando su Respuesta

Software Analizado:

Skype

- **Mantenibilidad**: Microsoft (empresa propietaria de Skype desde el año 2011) ha tratado de igualar las actualizaciones existentes en las plataformas de escritorio como las de los teléfonos móviles, aunque en este último los errores han sido erradicados de manera más paulatina que en la primera. Uno de los principales problemas que tuvo Skype al enfrentar esta problemática, fue la expansión y soportes a distintas plataformas que se vio obligada a presentar ante su compra por parte de la empresa de Gates. La integración junto a Windows Live Messenger, luego a la red social Facebook y las consecuentes redes sociales en las que está inmersa el gigante de Mark Zuckerberg, provocó que el mantenimiento no sea para nada escalable. Esto también generó la respuesta de la empresa en tratar tanto a las versiones on line, como las de escritorio y las de los teléfonos móviles como algo único, por lo que en el año 2018 muchas de las nuevas funcionalidades se unificaron. Aún así, los constantes cambios en los que se ve inmerso el software, hace que sea muy inestable a la hora de poder corregirlo y se vuelva adaptativo. El que haya dejado de ser en el 2011 software libre, complica mucho más aún las correcciones para poder erradicar errores [1].

- **Confiabilidad**: Una de las desventajas con las que corrió Skype, es que su código pasó de ser abierto, a completamente cerrado en el 2011, tras la adquisición por parte de Microsoft. Esto dificulta mucho la visualización de las problemáticas con respecto a seguridad y confiabilidad, y que cabe destacar, en el último decenio se han ido intensificando a partir de los escándalos a raíz de lo publicado por Edward Snowden en el año 2013 [1].

- **Eficiencia**: Dadas las mejoras a las que se ha ido dotando a Skype en las distintas plataformas desde el año 2011 hasta la fecha, la eficiencia ha sido afectada progresivamente. Bajo la modalidad de escritorio, el software ha ido mejorando su capacidad y velocidad, aunque muchas veces en detrimento de los recursos consumidos.

  En la plataforma móvil, las diferencias son notables ante la merma de la aplicación bajo Android, que bajo iOS. En el primero, los recursos consumidos son visiblemente mayores, provocando que no sea una opción para el usuario final, a la hora de optar por un programa de comunicación TCP/IP.

  Bajo iOS, el comportamiento es mucho mejor, los recursos consumidos no son tantos, por lo que no se ve afectada la eficacia.

- **Usabilidad**: Al igual que el punto anterior, dependiendo de la plataforma seleccionada, la usabilidad cambia drásticamente. Bajo la plataforma de escritorio, bajo sistemas operativos de tipo GNU/Linux, la aplicación es muy poco amigable, sus íconos han quedado arcaicos y muchas veces ya no hay soportes para versiones no tan antiguas del núcleo Linux. Para sistemas de tipo Windows, la aplicación ha sido mucho más amigable, pero aún así ha sufrido distintos embates en lo que respecta a experiencia de usuarios y diseño; lo que da como resultado una ventana temporal muy pequeña para que el usuario pueda adaptarse tanto a las nuevas prestaciones, como también a los cambios diseños que trae aparejado cada nueva característica.

  Bajo la modalidad móvil, las características se emparejan bastante ante Android y iOS. Si bien en el primero los modificaciones son más drásticos en cada actualización, siempre sacrificando accesibilidad para el usuario, para iOS los cambios son más bien estables (presentado ante todo por las duras pruebas de aceptabilidad que tiene Apple) en favor del usuario final. Salvando estas dos diferencias, se puede asegurar que Skype en el 2020 pudo mantener una estabilidad ante la constante mutabilidad exigida por la pandemia. Aún así, competidores con menos tiempo en el mercado, han ido quitándole terreno, como puede ser la aplicación de moda Zoom [1]

- **Corrección**: Desde el año 2018, donde se integraron todas las funcionalidades dispersas en distintos módulos bajo una misma plataforma transversal, puede decirse que ha ido cumpliendo de forma muy poco fiable esta función. Se registraron varios problemas de integración a comienzo de ese mismo año, aunque para mediados del 2019, la aplicación llegó estable a las distintas propuestas.

[1] Para más información, se puede consultar: `https://www.theverge.com/2020/3/31/21200844/microsoft-skype-zoom-houseparty-coronavirus-pandemic-usage-growth-competition`

Aún así, el propósito de comunicación completamente bajo TCP/IP, tiene muchas articulaciones distintas según el dispositivo que se utilice, lo que vuelve complejo abordar las problemáticas que van surgiendo continuamente.

2.6. Desarrolle cinco ejemplos donde lo aplicaría conceptos de Ingeniería de Software en forma detallada, justificando su respuesta

En primer lugar, se necesitaría realizar un análisis de los distintos requerimientos estableciendo para ello los lineamientos generales con el cliente. Se deberá de tener conocer mediante el análisis funcional, toda la envergadura del negocio, como también tener en cuenta los requisitos no funcionales que puedan surgir.

Habrá que coordinar con el cliente cuáles serán los estándares de calidad que un sistema de comunicación totalmente basado en IP (como Skype) deberá de cumplir; como también los recursos físicos con los que se cuenta para implementar la solución deseada.

Se debe tener presente que los requerimientos planteados por el cliente, pueden estar sujetos a cambios de último momento provistos a lo largo del Proceso de Producción del software, dado que se se deberá de trabajar con Metodologías ágiles.

En segundo lugar, se necesitará realizar un diseño asequible de abordar tanto por el cliente o *Product Owner* como por el equipo de desarrollo. En este diseño se deberán expresar las relaciones existentes entre cada una de las entidades que vayan estructurando a la solución deseada. Esto puede ser fácilmente reconocible en los diagramas de Entidad-Relación, subrayando las entidades como *emisor*, *receptor* o *mensaje*.

A partir de este diseño, se pueden ir diagramando también los primeros bocetos para la interfaz de usuario (*front-end*) como también en las conexiones que habrá que elaborar para poder mantener los datos de forma consistente (*back-end*), pudiendo delinear el tipo de base de datos que se deberá utilizar.

Una vez bosquejados estos primeros pasos, es posible comenzar a evaluar los tiempos de desarrollo, sus costos y también delinear la arquitectura donde se cimentarán los pasos descriptos anteriormente. Skype utiliza una novedosa arquitectura de red denominada híbrida, dado que los clientes hacen de servidores y viceversa.

La metodología a emplear también podrá ser escogida en período, ya que el cliente tendrá a la vista los recursos y las horas hombre a emplear. Quizás un procedimiento lineal como el desarrollo en cascada (*waterfall model*) no podría ser útil, ya que si bien el cliente puede tener todos sus requerimientos detallados y la problemática a dar solución bien delineada, sería difícil poder seguir la dinámica de cambio bajo este tipo de método.

Por último, será necesario abordar los distintos equipos que integren el desarrollo: Analistas, Desarrolladores, Testers, personal de UX, etc; clasificar según jerarquía, especialidad y antiguedad en proyectos similares y comenzar con las primeras reuniones para planificar el calendario de entregas y sucesivamente, las iteraciones a tener en cuenta.

2.7. Compare el contenido del Caso Presentado y realice la relación detallada de cada uno de los principios del código de ética de Ingeniería de Software por intermedio de una tabla

1. **Sociedad**: Se deberán resguardar cada uno de los datos ingresados en el sistema de reservas, siendo esto algo básico para poder sortear cualquier prueba de calidad a la que se vea evaluada nuestra solución. Estos datos pueden ser tanto personales, como también de entidades jurídicas, que afectan no sólo a una persona sino también a un grupo más grande. No hay que perder de vista que la cordialidad con el cliente es una parte fundamental para la buena comunicación y desarrollo del equipo de trabajo.

2. **Cliente y Empresario**: No sólo es importante la seguridad descripta en el punto anterior, sino también es importante la confidencialidad de la información por parte del equipo, empresa o persona que tomó para sí, las tareas de desarrollo de las soluciones. El cliente asimismo debe dar su consentimiento y aprobar las acciones que se estarán realizando durante el período de implementación, si así se requiere. Por ejemplo, si se requiere consumir datos externos y cruzarlos con datos propios para una eventual aprobación de pagos, será importante que el cliente esté enterado de esto como también lo apruebe.

3. **Producto**: Será necesario contar con la documentación detallada tanto de los requerimientos funcionales como también de los diseños técnicos, que hayan ido produciéndose a lo largo de la vida del proyecto. Será necesario tener en cuenta la escalabilidad de la solución en caso de la necesidad de acoplar nuevos módulos de negocios y el posible impacto que esto suministrará.

4. **Juicio**: Cabe destacar como en cualquier proyecto profesional, siempre mantener una objetividad técnica ante cualquier problemática que pueda acaecer. Es importante mantener una objetividad y distancia con los problemas que puedan presentarse diariamente, para mantener los valores y el trabajo en equipo durante todo el ciclo de productividad, como para el de documentación. Para ello existen técnicas muy bien valoradas en el ámbito profesional como pueden ser las denominadas *retrospectivas*.

5. **Administración**: Es conveniente utilizar procedimientos de desarrollo que permitan en alcanzar los estándares de calidad buscados por el cliente, en tiempos sensatos y preestablecidos. Es importante tener en cuenta las prácticas que da prioridad del cliente y a cuáles no.

6. **Profesión**: Durante la implementación de las soluciones alcanzadas por el equipo, los errores son parte inevitable del ciclo de vida del proyecto, por lo que es importante en estos casos detectarlos, reportarlos y finalmente corregirlos. Un punto que es contrario al Código de Ética Profesional, es el de ignorar alguna irregularidad grave y presentar la solución con características no viables para poder cumplir tiempos y fechas preestablecidos, teniendo como contrariedad un

aumento de tiempo considerable y de trabajo mal gastado en solucionarlo una vez productivo que el haberlo modificado de haberlo detectado a tiempo.

7. **Colegas**: Es de suma importancia acudir a los profesionales de mayor jerarquía y de mayor experiencia cuando las soluciones no parecen viables o cuando no se tienen conocimientos sobre cómo abordar determinadas temáticas nebulosas, ambiguas o difíciles de acometer. Es por ello que el equipo debe estar dotado de profesionales de las más diversas experiencias y áreas, ya sean de seguridad, infraestructura, desarrollo y hasta de operaciones, según la necesidad del proyecto.

   Revisar el desempeño de todo el equipo cada cierto plazo de tiempo, mejorará la adaptación de las necesidades del cliente, y a sus tiempos.

8. **Personal**: Al finalizar un proyecto, una fase o un sprint una buena práctica para todo el equipo es realizar una instrospección sobre, los problemas acaecidos, inquirir acerca del desempeño a nivel individual y grupal, preguntarse si el cliente está o fue satisfecho con los resultados, aprender de estos errores y logros y seguir adelante.

Bibliografía

[1] Esteban Magnani. *Tension en la red*. Adruki, 2014.

[2] Roger S. Pressman. *Ingenieria de Software*. McGraw Hill, 2010.

[3] Neal Stephenson. *In the beginning... was the command line*. Avon, 1999.