



Instituto Politécnico Nacional



Escuela Superior de Cómputo

Tarea 1.- Ciclo de Vida del Software

Materia:

Ingeniería de Software

Grupo:

3CM13

Profesor:

Méndez Segundo Laura

Alumno:

Castro Cruces Jorge Eduardo

Fecha:

martes, 1 de febrero de 2022

INTRODUCCIÓN

El ciclo de vida del software con frecuencia se usa como otro nombre para el proceso de software; Originalmente se acuñó para referirse modelo de cascada del proceso de software.[1]

Un proceso de software es una serie de actividades relacionadas que conduce a la elaboración de un producto de software. Estas actividades pueden incluir el desarrollo de software desde cero en un lenguaje de programación estándar como Java o C. Sin embargo, las aplicaciones de negocios no se desarrollan precisamente de esta forma. El nuevo software empresarial con frecuencia ahora se desarrolla extendiendo y modificando los sistemas existentes, o configurando e integrando el software comercial o componentes del sistema. [1]

Existen muchos diferentes procesos de software, pero todos deben incluir cuatro actividades que son fundamentales para la ingeniería de software:

1. *Especificación del software* Tienen que definirse tanto la funcionalidad del software como las restricciones de su operación.
2. *Diseño e implementación del software* Debe desarrollarse el software para cumplir con las especificaciones.
3. *Validación del software* Hay que validar el software para asegurarse de que cumple lo que el cliente quiere.
4. *Evolución del software* El software tiene que evolucionar para satisfacer las necesidades cambiantes del cliente[1]

En cierta forma, tales actividades forman parte de todos los procesos de software. [1]

Por supuesto, en la práctica éstas son actividades complejas en sí mismas e incluyen subactividades tales como la validación de requerimientos, el diseño arquitectónico, la prueba de unidad, etcétera. También existen actividades de soporte al proceso, como la documentación y el manejo de la configuración del software. [1]

Cuando los procesos se discuten y describen, por lo general se habla de actividades como especificar un modelo de datos, diseñar una interfaz de usuario, etcétera, así como del orden de dichas actividades. Sin embargo, al igual que las actividades, también las descripciones de los procesos deben incluir:

1. Productos, que son los resultados de una actividad del proceso. Por ejemplo, el resultado de la actividad del diseño arquitectónico es un modelo de la arquitectura de software.
2. Roles, que reflejan las responsabilidades de la gente que interviene en el proceso. Ejemplos de roles: gerente de proyecto, gerente de configuración, programador, etcétera.
3. Precondiciones y postcondiciones, que son declaraciones válidas antes y después de que se realice una actividad del proceso o se cree un producto. Por ejemplo, antes de comenzar el diseño arquitectónico, una precondición es que el cliente haya aprobado

todos los requerimientos; después de terminar esta actividad, una postcondición podría ser que se revisen aquellos modelos UML que describen la arquitectura. [1]

EL CICLO DE VIDA DEL DESARROLLO DE SISTEMAS

En este capítulo hemos hecho referencia a la metodología sistemática con la que los analistas llevan a cabo el análisis y diseño de los sistemas de información. Gran parte de ello se expresa en lo que conocemos como el ciclo de vida del desarrollo de sistemas (SDLC). El SDLC es una metodología en fases para el análisis y diseño, de acuerdo con la cual los sistemas se desarrollan mejor al utilizar un ciclo específico de actividades del analista y los usuarios.[2]

Los analistas no se han puesto de acuerdo sobre la cantidad de fases que hay en el SDLC, pero por lo general alaban su metodología organizada. En este libro vamos a dividir el ciclo en siete fases, como se muestra en la *figura 1*. Aunque cada fase se presenta de manera discreta, en realidad nunca se puede llevar a cabo como un paso separado, sino que varias actividades pueden ocurrir al mismo tiempo, e incluso se pueden repetir. [2]

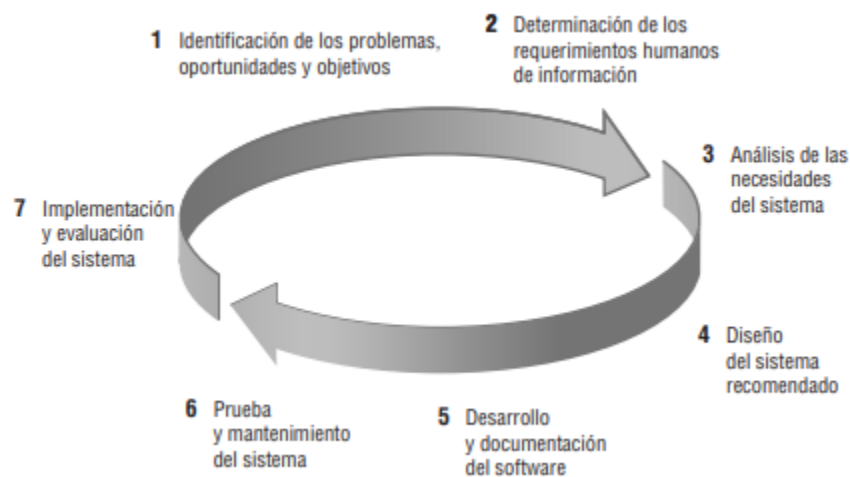


FIGURA 1 Las siete fases del ciclo de desarrollo de sistemas (SDLC). [2]

1. Identificación de los problemas, oportunidades y objetivos

En esta primera fase del ciclo de vida del desarrollo de sistemas, el analista se encarga de identificar correctamente los problemas, las oportunidades y los objetivos.

En la primera fase el analista debe analizar con honestidad lo que está ocurriendo en la empresa. [2]

La identificación de los objetivos también es un componente importante de la primera fase. El analista debe descubrir primero qué trata de hacer la empresa. [2]

2. Determinación de los requerimientos de información del factor humano

La siguiente fase a la que entra el analista es determinar las necesidades de los usuarios involucrados, mediante el uso de varias herramientas, para comprender la forma en que interactúan en el contexto laboral con sus sistemas de información actuales. [2]

El analista utilizará estos métodos para plantear y responder muchas preguntas relacionadas con la interacción humano-computadora (HCI), incluyendo preguntas tales como: “¿Cuáles

son las fortalezas y limitaciones físicas de los usuarios?”, o dicho en otras palabras, “¿qué hay que hacer para que el sistema sea perceptible, legible y seguro?”, “¿cómo puede diseñarse el nuevo sistema para que sea fácil de usar, aprender y recordar?”, “¿cómo puede el sistema ser agradable o incluso divertido de usar?”, “¿cómo puede el sistema apoyar las tareas laborales individuales de un usuario y buscar nuevas formas de hacerlas más productivas?”. [2]

3. Análisis de las necesidades del sistema

La siguiente fase que debe llevar a cabo el analista de sistemas involucra el análisis de las necesidades del sistema. Aquí también hay herramientas y técnicas especiales que ayudan al analista a realizar las determinaciones de los requerimientos. Las herramientas como los diagramas de flujo de datos (DFD) para graficar la entrada, los procesos y la salida de las funciones de la empresa, o los diagramas de actividad o de secuencia para mostrar la secuencia de los eventos, sirven para ilustrar a los sistemas de una manera estructurada y gráfica. A partir de los diagramas de flujo de datos, de secuencia u otros tipos de diagramas se debe desarrollar un diccionario de datos para enlistar todos los elementos de datos utilizados en el sistema, así como sus especificaciones. [2]

4. Diseño del sistema recomendado

En la fase de diseño del SDLC, el analista de sistemas utiliza la información recolectada antes para realizar el diseño lógico del sistema de información. El analista diseña los procedimientos para ayudar a que los usuarios introduzcan los datos con precisión, de manera que los datos que entren al sistema de información sean los correctos. Además, el analista debe ayudar a que los usuarios completen la entrada de datos efectiva al sistema de información mediante el uso de las técnicas del buen diseño de formularios y páginas Web o pantallas. [2]

La fase de diseño también incluye el diseño de bases de datos que almacenarán gran parte de los datos necesarios para los encargados de tomar las decisiones en la organización. Los usuarios se benefician de una base de datos bien organizada que sea lógica para ellos y se corresponda con la forma en que ven su trabajo. En esta fase, el analista también trabaja con los usuarios para diseñar una salida (ya sea en pantalla o impresa) que cumpla con sus necesidades de información. [2]

5. Desarrollo y documentación del software

En la quinta fase del SDLC, el analista trabaja con los programadores para desarrollar el software original requerido. Durante ella, el analista desarrolla junto con los usuarios una documentación efectiva para el software, incluyendo manuales de procedimientos, ayuda en línea, sitios Web con preguntas frecuentes (FAQ) y archivos Léame (Read Me) para incluir con el nuevo software. Como los usuarios están involucrados desde el principio, la fase de documentación debe lidiar con las preguntas que hicieron y resolvieron junto con el analista. [2]

La documentación indica a los usuarios cómo deben usar el software y qué deben hacer en caso de que ocurran problemas. Los programadores desempeñan un rol clave en esta fase, ya

que diseñan, codifican y eliminan los errores sintácticos de los programas de computadora. Para asegurar la calidad, un programador puede llevar a cabo un recorrido por el diseño o por el código para explicar las porciones complejas del programa a un equipo formado por otros programadores. [2]

6. Prueba y mantenimiento del sistema

Antes de utilizar el sistema de información, se debe probar. Es mucho menos costoso detectar los problemas antes de entregar el sistema a los usuarios. Una parte del procedimiento de prueba es llevado a cabo por los programadores solos; la otra la realizan junto con los analistas de sistemas. Primero se completa una serie de pruebas para señalar los problemas con datos de muestra y después se utilizan datos reales del sistema actual. A menudo, los planes de prueba se crean en las primeras etapas del SDLC y se refinan a medida que el proyecto progresa. [2]

El mantenimiento del sistema y la documentación de este mantenimiento empieza en esta fase y se lleva a cabo de manera rutinaria durante toda la vida del sistema de información. [2]

7. Implementación y evaluación del sistema

En esta última fase del desarrollo de sistemas, el analista ayuda a implementar el sistema de información. En esta fase hay que capacitar a los usuarios para operar el sistema. Los distribuidores se encargan de una parte de la capacitación, pero la supervisión de la capacitación es responsabilidad del analista de sistemas. Además, el analista necesita planear una conversión sin problemas del sistema antiguo al nuevo. Este proceso incluye convertir los archivos de los formatos anteriores a los nuevos, o crear una base de datos, instalar equipo y llevar el nuevo sistema a producción. [2]

CONCLUSIONES

Podemos decir que El ciclo de vida del desarrollo de software (en inglés: SDLC – Systems Development Life Cycle) es la estructura que contiene los procesos, actividades y tareas relacionadas con el desarrollo y mantenimiento de un producto de software, abarcando la vida completa del sistema, desde la definición de los requisitos hasta la finalización de su uso.

Se trata de evitar los costes de rectificar errores de implementación mediante un método que permita a los programadores adelantarse para mejorar sus resultados finales.

Este sistema de desarrollo (o ciclo de vida del proceso de software), necesita de varios pasos imprescindibles para garantizar que los programas ofrezcan una buena experiencia al usuario, seguridad, eficiencia, estabilidad y fiabilidad de uso.

BIBLIOGRAFÍA

- [1] Sommerville, I. (2012). *Ingeniería de software (Spanish Edition) by Ian Sommerville* (9.^a ed.). Pearson (México). https://sistemamid.com/panel/uploads/biblioteca/2018-06-11_03-37-12144643.pdf
- [2] Kendall, K. E., & Kendall, J. E. (2011). *Análisis y diseño de sistemas* (8.^a ed.). PRENTICE HALL/PEARSON. https://ccie.com.mx/wp-content/uploads/2020/03/Id-Analisis-y-Diseno-de-Sistemas_Kendall-8va.pdf