

IEEE - SWEBOK

ROSAS TRINIDAD SERGIO ANTONIO

PRACTICAS



CAPITULOS

CAPITULO 1	2
CAPITULO 2	2
CAPITULO 3	3
CAPITULO 4	4
CAPITULO 5	5
CAPITULO 6	5
CAPITULO 7	6
CAPITULO 8	7
CAPITULO 9	8
CAPITULO 10	8
CAPITULO 11	9
CAPITULO 12	9
CAPITULO 13	10
CAPITULO 14	10
CAPITULO 15	11



CAPITULO 1

En este capítulo introduce al lector un ambiente general del paradigma experimental aplicado a la Ingeniería de Software (IS), así como presentan algunas investigaciones hechas en México que han utilizado este enfoque de investigación. También se comentan los enfoques ingenieriles en el desarrollo y mantenimiento de productos software datan desde la década de los años 60s, ya que hasta en la década de los 80s que la comunidad académica comenzó a adoptar y emplear enfoques de investigación para estudiar de manera más rigurosa los diferentes aspectos y problemáticas involucradas en el desarrollo de software.

Desde sus inicios en la década de 1940, escribir software ha evolucionado hasta convertirse en una profesión que se ocupa de cómo crear software y mejorar su calidad. La calidad puede referirse a cuán mantenible es el software, su estabilidad, velocidad, usabilidad, con probabilidad, legibilidad, tamaño, costo, seguridad y número de fallas o "bugs", así como, entre muchos otros atributos, a cualidades menos medibles como elegancia, concisión y satisfacción del cliente.

Puedo decir que la investigación en ingeniería de software tiene como fin generar conocimiento nuevo en el ámbito de esta disciplina, así como modificarlo de tal manera que este pueda ser incorporado en la práctica diaria del desarrollo de software. Esto con el fin de mejorar los plazos de entrega, la ejecución del presupuesto asignado, así como mejorar la calidad del producto software a desarrollar o mantener.

CAPITULO 2

En el área del conocimiento de los requisitos de software (KA), se refiere al análisis, especificación y validación de los requisitos de software. Donde nos están demostrando ampliamente que el simple hecho de no hacer las cosas bien este proceso puede traer consecuencias malas en el desarrollo de cualquier producto de software.

Uno de los requisitos del software es una característica que se debe exhibir para solucionar un cierto problema en el mundo real. Se convierte en una combinación compleja de requisitos entregados por parte de los usuarios implicados dentro del desarrollo de la solución, teniendo en cuenta que pueden corresponder a diferentes niveles jerárquicos, ambientes e intereses. Es importante también que cada requisito los podamos comprobar, pensando también en las implicaciones que esto puede llevar.



También nos comentan de todo lo que necesita el software para funcionar desde el punto de vista de hardware, software, recurso humano, información, instalaciones, servicios, etc.

Se dice que se tiene que tener ciertas habilidades con las que debe contar el ingeniero de software, apoyado en herramientas que le ayuden a contextualizar el software.

También la Negociación de los requisitos que llega a esta etapa cuando hay incompatibilidad en dos o más requisitos y los solicitantes son distintos dice que el ingeniero de software debe reunirlos y tomar la decisión más conveniente para el correcto funcionamiento de la solución.

Existen medios que utiliza el ingeniero de software para manifestar lo que entendió y para facilitar la corrección, eliminación o adición de requisitos. Como se evidencia, la ventaja de hacerlo es grande pero el costo puede ser alto.

Hay Pruebas de Aceptación Son aquellas que se le aplican a cada requisito para determinar si el producto final lo satisface o no.

Debe haber total planeación para aplicar estas pruebas y hacerlo de manera cuantitativa. Como conclusión para este capítulo puede afirmarse que los requisitos de software deben tomarse como una práctica seria, asignándole el tiempo y los recursos necesarios que permitan continuar con un proceso ordenado para llegar a obtener un software de calidad.

CAPITULO 3

Nos especifican el término construcción de software que se refiere a la creación detallada de software funcional mediante una combinación de codificación, verificación, pruebas unitarias, pruebas de integración y depuración. El área de conocimiento de Construcción de software (KA) está vinculada a todos los demás KA, pero está más fuertemente vinculada al Diseño de software y las Pruebas de software porque el proceso de construcción de software implica un diseño y pruebas de software importantes.

Comenta sobre el proceso que utiliza la salida del diseño y proporciona una entrada para las pruebas. Dice que los límites entre el diseño, la construcción y las pruebas variarán según los procesos del ciclo de vida del software que se utilicen en un proyecto, aunque se puede realizar algún diseño detallado antes de la construcción, gran parte del trabajo de diseño se realiza durante la actividad de



construcción. Por lo tanto, el KA de construcción de software está estrechamente relacionado con el KA de diseño de software.

Durante la construcción, los ingenieros de software, tanto la prueba unitaria como la integración, prueban su trabajo. Por lo tanto, el KA de construcción de software también está estrechamente relacionado con el KA de pruebas de software.

Es importante saber que la construcción de software generalmente produce la mayor cantidad de elementos de configuración que deben administrarse en un proyecto de software por lo tanto, el KA de construcción de software también está estrechamente vinculado al KA de gestión de configuración de software. Si bien la calidad del software es importante en todos los KA, el código es el resultado final de un proyecto de software y, por lo tanto, el KA de calidad del software está estrechamente relacionado con el KA de construcción de software.

También está relacionado con la gestión de proyectos, en la medida en que la gestión de la construcción puede presentar desafíos considerables.

CAPITULO 4

Se señala que una actividad que permite evaluar y mejorar la calidad del producto con el fin de detectar fallas y corregir errores. Las pruebas del software consisten en verificar el comportamiento de un programa dinámicamente a través de un grupo finito de casos de prueba, debidamente seleccionados.

Se ha ido cambiando la percepción de que las pruebas de software se realizan únicamente al final del proceso de creación de código fuente, siendo muy útil hacerlo en todas las etapas del desarrollo del software; esto permite corregir errores y detectar fallas de fondo, a tiempo.

En el capítulo se dice que en los últimos años, la visión de las pruebas de software se ha convertido en una visión constructiva. Las pruebas ya no se consideran una actividad que comienza solo después de que se completa la fase de codificación con el propósito limitado de detectar fallas.

Las pruebas de software son, o deberían ser, generalizadas durante todo el ciclo de vida de desarrollo y mantenimiento. De hecho, la planificación de las pruebas de software debe comenzar con las primeras etapas del proceso de requisitos de software, y los planes y procedimientos de prueba deben desarrollarse sistemática y continuamente, y posiblemente refinarse, a medida que avanza el



desarrollo de software y estas actividades de planificación y diseño de pruebas proporcionan información útil para los diseñadores de software y ayudan a resaltar posibles debilidades, como descuidos / contradicciones de diseño u omisiones / ambigüedades en la documentación.

CAPITULO 5

El proceso de desarrollo de software debe satisfacer los requerimientos planteados, una vez en operación el proceso de cubrimiento de defectos, operación y cambio de ambiente debe darse en esta etapa, la fase de mantenimiento empieza con un periodo de garantía y de soporte post-implementación, pero el mantenimiento del software ocurre mucho antes. Aunque la etapa de mantenimiento del software no ha tenido el grado de atención que se debe este tipo de desarrollo de software ya está empezando a cambiar ya que muchos errores graves han ocurrido por no prestarle la atención que se merece.

Al igual los esfuerzos de desarrollo de software dan como resultado la entrega de un producto de software que satisface los requisitos del usuario.

Como consecuencia, el producto de software debe cambiar o evolucionar. Una vez en funcionamiento, se descubren defectos, los entornos operativos cambian y surgen nuevos requisitos de usuario. La fase de mantenimiento del ciclo de vida comienza después de un período de garantía o de la entrega de soporte posterior a la implementación, pero las actividades de mantenimiento ocurren mucho antes.

CAPITULO 6

Este capítulo menciona que un sistema puede definirse como la combinación de elementos que interactúan organizados para lograr uno o más propósitos establecidos como la configuración de un sistema son las características funcionales y físicas del hardware o software o es lo que según se establece en la documentación técnica o se logra en un producto, también consideran como una colección de versiones específicas de hardware o elementos de software con procedimientos de construcción específicos para cumplir un propósito en particular.

La gestión de la configuración (CM), entonces, es la disciplina de identificar la configuración de un sistema en distintos puntos en el tiempo con el fin de controlar sistemáticamente los cambios en la configuración y mantener la integridad y trazabilidad de la configuración a lo largo del ciclo de vida del



sistema y una disciplina que aplica la dirección y vigilancia técnica y administrativa para: identificar y documentar las características funcionales y físicas de un elemento de configuración, controlar los cambios en esas características, registrar e informar el estado de implementación y procesamiento de cambios y verificar el cumplimiento de los requisitos especificados.

La gestión de la configuración del software es un proceso del ciclo de vida del software de apoyo que beneficia la gestión de proyectos, las actividades de desarrollo y mantenimiento, las actividades de garantía de calidad, así como a los clientes y usuarios del producto final. Los conceptos de gestión de la configuración se aplican a todos los elementos que se van a controlar, aunque existen algunas diferencias en la implementación entre la que brindan garantía de que los productos y procesos de software en el ciclo de vida del proyecto cumplen con sus requisitos especificados mediante la planificación, la implementación y la realización de un conjunto de actividades para proporcionar la confianza adecuada de que la calidad es incorporado en el software.

Las actividades de SCM son la administración y planificación del proceso de SCM, la identificación de la configuración del software, el control de la configuración del software, la contabilidad del estado de la configuración del software, la auditoría de la configuración del software y la administración y entrega de versiones del software.

CAPITULO 7

Este capítulo puede ser definido como las actividades de gestión de la aplicación, planeación, coordinación, medición, monitorización, control y reportes para asegurar el desarrollo y mantenimiento del software como sistemático, disciplinado y cuantificable ya que esto es un aspecto muy importante en la administración y medición de la ingeniería del software.

Aquí se destacan tres niveles: Gestión y organización de la infraestructura, gestión de proyectos, y control y planeación del programa de medidas. Los aspectos de la gestión de la organización son importantes en términos del impacto en la ingeniería del software y en las políticas de gestión, esas políticas pueden ser influenciadas por los requerimientos de un software efectivo, mantenimiento y desarrollo. Un número de políticas específicas deben ser establecidas para una efectiva gestión en la ingeniería del software y el nivel organizacional.



Se comenta acerca de evaluar las mediciones de este proceso ya que se debe llevar a cabo con un criterio de evaluación específico para determinar las fuerzas y debilidades de los productos, se puede hacer por medio de un proceso de auditoría interna o externa y debe incluir una retroalimentación a los usuarios. Se deben identificar las mejoras potenciales y costos y beneficios de estas, comunicarlás a la persona encargada para su revisión y aprobación.

CAPITULO 8

En el capítulo se examinan dos niveles desde lo técnico y desde el meta-nivel o nivel de implementación, valoración, medición y gestión también existen varios significados sobre el proceso de la ingeniería del software puede verse como una sola manera de realizar el proceso o como muchas maneras que es lo que se quiere y se debe llegar a hacer ya que el proceso de la ingeniería abarca muchos factores, finalmente un tercer significado se refiere al conjunto actual de actividades realizadas dentro de una organización que se puede ver como un solo proceso. Los procesos de ingeniería del software son considerados de gran importancia, el objetivo es gestionar nuevos y mejores procesos.

También comentan acerca de la medición de los procesos y los productos, aunque puede resultar compleja la medición a la ingeniería del software existen varios aspectos que son fundamentales para la medición y análisis, estas mediciones se pueden realizar para apoyar los procesos de implementación y cambio o evaluar consecuencias de estos.

La medición de un producto software incluye principalmente, la medición del tamaño del producto, la estructura del producto y la calidad del producto.

También es importante tener en cuenta la medición del tamaño, de la estructura y de la calidad. Para garantizar la calidad en los resultados de la medición es necesaria la medición efectiva de los programas para proveer resultados exitosos.



CAPITULO 9

En este capítulo el tema principal son los instrumentos asistidos por ordenador que se requieren para ayudar a los procesos de ciclo de vida del software y nos ayuda a reducir la carga cognoscitiva enfocándonos más en los aspectos creativos del proceso.

También los métodos imponen la estructura a la actividad de la ingeniería del software con el objetivo de hacerla sistemática, por lo general proporcionan notación y vocabulario para comprobar tanto el proceso como el producto, aunque existen numerosos materiales sobre los instrumentos de apoyo en la ingeniería del software, los textos sobre las técnicas sobre estos instrumentos son relativamente escasos, una dificultad es el alto costo que representa un cambio de instrumento de software en general, los instrumentos y métodos cubren ciclos de vida completos por esto es tan complicado hacer un cambio.

También se cubren métodos que implican el prototipado de software y es subdivida en estilos de prototipado, objetivos y técnicas de evaluación. Se deben incluir aspectos como estilos de prototipado, objetivos del prototipado, y las técnicas para la evaluación del prototipo propuesto.

CAPITULO 10

En este capítulo se abordan los aspectos relativos a la calidad del software los cuales trascienden en cualquier ciclo de vida, en esta guía se describen un conjunto de métodos para alcanzar la calidad, en este caso se tratarán las técnicas estáticas es decir aquellas que no requieren la ejecución del software para su evaluación mientras las dinámicas cubren los aspectos de calidad en las pruebas del software como los Fundamentos de Calidad del Software que aquí se definen formalmente los aspectos a tratar y la manera como un ingeniero de software debería entender y adoptar los conceptos y características de calidad y su relevancia en el desarrollo o mantenimiento de software.

Se comentan algunos aspectos de calidad que deben estar inherentes desde el momento mismo de los requerimientos, así como la medición y criterios de aceptación que evalúan estas características.

Modelos y Características de Calidad usada en las características que cada una posee niveles jerárquicos diferentes, hay tres modelos relacionados con la calidad en un producto de software. La mejora de calidad da a la tarea de ser mejorada cada vez más gracias a un proceso iterativo de mejora continua que requiere control de dirección, control y retroalimentación de muchos procesos.



CAPITULO 11

Este capítulo se centra en relacionar cada una de las disciplinas con la Ingeniería del Software.

Es específico en lo que respecta a determinar las temáticas de cada una de ellas en este capítulo se centra en relacionar cada una de las disciplinas con la Ingeniería del Software.

Es específico en lo que respecta a determinar las temáticas de cada una de ellas ya que el área de conocimiento de la práctica profesional de ingeniería de software se ocupa del conocimiento, las habilidades y las actitudes que los ingenieros de software deben poseer para practicar la ingeniería de software de manera profesional, responsable y ética. Debido a las aplicaciones generalizadas de los productos de software en la vida social y personal, la calidad de los productos de software puede tener un impacto profundo en nuestro bienestar personal y armonía social. Los ingenieros de software deben manejar problemas de ingeniería únicos, produciendo software con características y confiabilidad conocidas. Este requisito requiere ingenieros de software que posean un conjunto adecuado de conocimientos, habilidades, capacitación y experiencia en la práctica profesional.

CAPITULO 12

Yo pienso que respecto a la economía de la ingeniería de software trata de tomar decisiones relacionadas con la ingeniería de software en un contexto empresarial.

Se dice que el éxito de un producto, servicio y solución de software depende de una buena gestión empresarial, sin embargo, en muchas empresas y organizaciones, las relaciones comerciales de software con el desarrollo y la ingeniería de software siguen siendo bajas. Esta área de conocimiento proporciona una descripción general de la economía de la ingeniería de software.

Se comenta que la economía es el estudio del valor, los costos, los recursos y su relación en un contexto o situación determinados. En la disciplina de la ingeniería de software, las actividades tienen costos, pero el software resultante en sí mismo tiene atributos económicos también y la economía de la ingeniería de software proporciona una forma de estudiar los atributos del software y los procesos de software de una manera sistemática que los relaciona con medidas económicas. Estas medidas



económicas pueden pensarse y analizarse al tomar decisiones que están dentro del alcance de una organización de software y aquellas dentro del alcance integrado de todo un negocio de producción o adquisición.

CAPITULO 13

Es importante saber el alcance del área de conocimiento de Fundamentos de la Computación ya que abarca el entorno operativo y de desarrollo en el que el software evoluciona y se ejecuta, debido a que ningún software puede existir en el vacío o ejecutarse sin una computadora, el núcleo de dicho entorno es la computadora y sus diversos componentes.

El conocimiento sobre la computadora y sus principios subyacentes de hardware y software sirve como marco sobre el que se ancla la ingeniería de software.

La mayoría de los temas discutidos en el capítulo también son temas de discusión en cursos básicos impartidos en programas de pregrado y posgrado en ciencias de la computación. Dichos cursos incluyen programación, estructura de datos, algoritmos, organización de computadoras, sistemas operativos, compiladores, bases de datos, redes, sistemas distribuidos, etc. Por lo tanto, al desglosar temas, puede ser tentador descomponer los Fundamentos de Computación de acuerdo con estas divisiones que se encuentran a menudo en los cursos relevantes.

CAPITULO 14

Se habla de los profesionales del software y como conviven con los programas. En un lenguaje muy simple, uno puede programar solo para algo que sigue una lógica bien entendida.

El área de conocimiento de Fundamentos matemáticos ayuda a los ingenieros de software a comprender esta lógica, que a su vez se traduce al código del lenguaje de programación. Las matemáticas que son el enfoque principal en este son bastante diferentes de la aritmética típica, donde los números se tratan y se discuten.

La lógica y el razonamiento son la esencia de las matemáticas que un ingeniero de software debe abordar.

Se comenta que las matemáticas, en cierto sentido, son el estudio de sistemas formales. La palabra "formal" está asociada a la precisión, por lo que no puede haber una interpretación ambigua o errónea del hecho. Por lo tanto, las matemáticas son el estudio de todas y cada una de las verdades ciertas



sobre cualquier concepto. Este concepto puede ser tanto de números como de símbolos, imágenes, sonidos, videos, casi cualquier cosa.

En resumen, no solo los números y las ecuaciones numéricas están sujetos a precisión si no por el contrario, un ingeniero de software necesita tener una abstracción precisa en un dominio de aplicación diverso.

El capítulo cubre técnicas básicas para identificar un conjunto de reglas para el razonamiento en el contexto del sistema en estudio. Todo lo que se pueda deducir siguiendo estas reglas es una certeza absoluta dentro del contexto de ese sistema. En este se definen y discuten técnicas que pueden representar y hacer avanzar el razonamiento y juicio de un ingeniero de software de una manera precisa y por lo tanto matemática.

El lenguaje y los métodos de lógica que se discuten aquí nos permiten describir pruebas matemáticas para inferir de manera concluyente la verdad absoluta de ciertos conceptos más allá de los números.

CAPITULO 15

IEEE define la ingeniería como “la aplicación de un enfoque sistemático, disciplinado y cuantificable a estructuras, máquinas, productos, sistemas o procesos”

En este capítulo describe algunas de las habilidades y técnicas fundamentales de ingeniería que son útiles para un ingeniero de software. La atención se centra en temas que apoyan a otros mientras se minimiza la duplicación de temas cubiertos en otras partes de este documento.

A medida que madura la teoría y la práctica de la ingeniería de software, es cada vez más evidente que la ingeniería de software es una disciplina de ingeniería que se basa en conocimientos y habilidades comunes a todas las disciplinas de ingeniería.

Esta área de conocimiento de Fundamentos de Ingeniería (KA) se ocupa de los fundamentos de ingeniería que se aplican a la ingeniería de software y otras disciplinas de ingeniería. Los temas de esta incluyen métodos y técnicas experimentales como análisis estadístico, medición, diseño de ingeniería, modelado, creación de prototipos y simulación, estándares, y análisis de la causa raíz. La aplicación de este conocimiento, según corresponda, permitirá a los ingenieros de software desarrollar y mantener el software de manera más eficiente y eficaz.