

APUNTES SEMANALES INDIVIDUALES

La ingeniería de software es la aplicación sistemática de enfoques de ingeniería para el desarrollo de software. La ingeniería de software es una disciplina informática.

Señor que se llama Fridrik Powers quién fue director del comité de ciencia de la OTAN en el año 67 y fue que en el año 67 fue quien propuso ese término de ingeniería de eso son peras o son manzanas lo importante es que ya en los años sesentas ya el desarrollo de software o lo que más bien se usaba como programación ya la gente se dio cuenta que se necesitan ciertas formas mejor organizadas sobre todo que en desarrollo de software empezaron para proyectos más grandes como el proyecto de apoyo por ejemplo se necesitaba el trabajo de varias personas varios programadores que tenían que organizarse y trabajar como equipos y entonces los que tenían la experiencia en trabajo en equipo para construir cosas fueron ingenieros de otras disciplinas entonces dijeron por qué no aplicar formas más organizadas más sistemáticas de hacer las cosas que hacen en otras ingenierías y aplicarlo para el software.

se define ingeniería de software como una aplicación sistemática de conocimientos científicos y tecnológicos métodos y experiencia al diseño e implementación prueba y documentación de software entonces aquí pueden ver que tenemos varios elementos por un lado y las ingenierías tratan de aplicar de manera sistemática el conocimiento de cada una de las áreas de conocimiento científico apoyándose con conocimientos tecnológicos para organizar el trabajo de manera sistemática por un lado se aplican métodos pero por otro lado la gente usa mucho su propia experiencia lo que se genera como producto es algo el software que se tiene que diseñar implementar probar y tiene que estar documentado entonces estos son como elementos básicos que se como Weigand para para ver si tenemos lo que hacemos en el desarrollo de software se acerca más o menos a una ingeniería en sentido bueno parecido a otras ingenierías y el software hay otra de bueno hay definición también del software mismo cómo está tu eres intangible entonces con mi amiga y colega una profesora también de la facultad de ciencias para nuestros alumnos hemos inventado la representación del software en forma de este de este duendecito la definición del el stand como la definición más estandarizada del software es el conjunto de programas de cómputo procedimientos reglas documentación y datos asociados que forman parte de las obras ah pero no tenemos gente entonces en esa definición porque bueno aunque lo básico son los programas pero los programas tienen que ir acompañado con esos otros elementos que aquí se mencionan para que realmente los programas puedan operar sobre un sistema de cómputo.

el contexto de ingeniería de software y de tecnología de información yo veo un sistema computacional o como aquí se tradujo como de cómputo como un mate un tercio por un lado son todas las cosas físicas todo el hardware si redes y etcétera lo que es físico por el otro lado es la parte intangible de software que permite mover este físico y por tercer lado somos los seres humanos que lo echamos a andar y lo aprovechamos entonces un sistema de cómputo o sistema computacional o sistema de software.

El desarrollo en espiral es un modelo de ciclo de vida del software definido por primera vez por Barry Boehm en 1986, utilizado generalmente en la ingeniería de software.

Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. Las actividades no están fijadas a ninguna prioridad, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior.

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.

En el video nos muestra la trayectoria de la Dr. Hanna Oktaba que llegó a México con su esposo como un intercambio de maestros en la UNAM.

Se ha quedado en México porque le gusta nuestro país y su cultura.

De ahí nos comparte respecto a la computación de hace como 60 años en México.

En 1958 un ing. Llamado Sergio Beltrán trajo la primera computadora de cálculo electrónico de la facultad de ciencias.

Era una IBM 650. Y pesaba como 900 Kg.

Fue fundador de la maestría en Informática en la UNAM.

Antecedentes del término de ingeniería de software:

Anthony Oettinger en 1966 habló de la profesión de la ingeniería de software.

Pero dicen que Margaret Hamilton fue la que usó el término de ingeniería de software para distinguirse de los otros trabajos en donde estaban haciendo cosas para el Apolo.

Hay otro que se llama Friedrich Ludwic propuso el término de ingeniería de software.

¿El software que es? Es una aplicación sistemática de conocimientos científicos y tecnológicos, métodos y experiencias al diseño, implementación, prueba y documentación de software.

Otra definición puede ser el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones d un sistema de cómputo.

La crisis de software se habló por primera vez del conjunto de dificultades o errores ocurridos en:

- Calidad de software
- Costos
- Gestión
- Profesión
- Proceso de desarrollo de software

Modelo espiral: surgió en los 80's por Barry Bohem, ciclos /iteraciones incrementales como objetivo, continuidad basada en análisis de riesgos.

Proceso Unificado surgido en los 90's Crearon Rational Unified Process Rup (tool) y OMG standard.

Y fue comprada por IBM en 2003.

Hay uno que se llama Modelo Ágil que fue creada en el 2000.

Como Scrum en el 86 (manufactura) y en 95 (software)

LIBRO

Panorama de la computación en México capítulo 1

Como va pasando el tiempo, la computación se va haciendo más grande en desarrollo, ha sido una tecnológica que hoy en día convivimos mucho con ella.

La tecnología es omnipresente y su importancia en todo el mundo.

En México a finales de los 50's su desarrollo ha girado principalmente en entorno las lógicas comerciales del país.

Siempre era el hardware y después empezó con el software hasta llegar al mercado que hoy tenemos.

Después a partir de algunas plataformas como PEMEX, IMSS, etc., empezaron a instalar sus centros de cómputo, los bancos y las aseguradoras principalmente.

El ecosistema de la computación en México siempre ha sido de dos tipos, tecnología propia como el desarrollo de dispositivos y sistemas, así también como SO, software de comunicación y el otro sería los recursos humanos que se enfoca principalmente en desarrollo y aplicaciones finales de la tecnología.

Esto de la computación llegó por medio de IBM y la UNAM las cuales fueron las que aportaron la tecnología.

Después hubo los comercios de equipos y sistemas que requerían de una fuerza de ventas, constituida por recursos humanos nacionales. En esta línea de comercialización los compradores eran el gobierno en diversos sectores.

Otra línea de comercialización es la venta de productos y sistemas directamente al consumidor final. La mayor parte se va a los vendedores y administradores, pero estamos conscientes de la lógica del ecosistema.

Un elemento esencial sería la educación en México que está orientada a satisfacer las demandas del ecosistema.

Lamentablemente la riqueza que se genera con este tipo de ecosistema no se va hacia la sociedad. Si no que se va al quien la crea, ya que el que la vende no genera mucha riqueza.

Apuntes semana 6.

02/NOVIEMBRE/2020

El software nos rodea. Está por todas partes, incluso ahora mismo, en esta habitación, puedes verla si miras por la ventana o al encender la televisión. Puedes sentirlo, cuando vas a trabajar, cuando vas al supermercado, cuando pagas tus impuestos. Es el mundo tecnológico que ha sido puesto ante tus ojos para ocultarte la verdad: que eres un esclavo del software. Sobre todo del mal software.

En principio la calidad del software no tiene una metodología estándar donde se pueda certificar. Son los procedimientos para desarrollar ese software los que se certifican y los que se pueden normalizar. La normativa iso 9000 Moprosoft, SW-CMM son algunos de ejemplos.

Es imposible hacer una medición correcta ya que las métricas deberían medir posibles situaciones como el número de errores que se generan durante un periodo de tiempo, las líneas de código, la velocidad de ejecución, la facilidad para modificar el código o añadir nuevas funcionalidades. Algunas de estas no implican una mejora en la calidad del código como por ejemplo que un programa tenga menos líneas de código no implica que tenga más calidad tanto en cuanto la modificación se convierta en un problema.

La conclusión inicial es que la medición de la calidad es algo subjetivo y depende de los parámetros que se midan un software puede ser más o menos de calidad.

Sin las medidas sólidas que muestren la eficacia de los distintos métodos los equipos de desarrollo van a tener problemas a la hora de seleccionar el método más efectivo.

La industria del software necesita mejores pruebas de la efectividad de los distintos métodos de desarrollo. Esto lleva al punto final en el que se necesita estandarizar las métricas del software. La presencia de puntuaciones de métricas incompatibles sin ningún ratio de conversión es en sí misma un ejemplo de innovación negativa ya que no aporta beneficio sino todo lo contrario.

SEMANA 13

PROYECTOS DE CALIDAD COMIENZAN CON REQUISITOS DE CALIDAD

El 47 por ciento de los fracasos en proyectos se deben a la mala gestión de requisitos

El 20 por ciento de los defectos tienen su origen en requisitos

Encontrar y corregir defectos en el software después de entregarlo es 100 por ciento más costoso que hacerlo en la fase de requisitos

La calidad es el grado en conjunto de características inherentes cumple con los requisitos ISO-9000

Un requisito de software es una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo

condición o capacidad que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato estándar, especificación u otra documentación formalmente impuesta

Representación documentada de una condición o capacidad como en 1 o 2

Las especificaciones y requisitos:

La especificación de requisitos ayuda a los clientes a describir con precisión lo que desean obtener de un software, ayuda a los desarrolladores a entender exactamente lo que quiere el cliente.

El rol de la especificación

ser un contrato entre cliente y desarrolladores no se debe enfocar en aspectos de diseño o implementación, además debe ser detallado, debe promover la comunicación entre dos partes, el nivel de confianza entre dos partes determina el nivel de detalle, teniendo clientes altamente involucrados, desarrolladores con experiencia considerable en el asunto del problema, existen antecedentes disponibles por ejemplo reingeniería, una

solución de paquete será utilizada, contexto más orientado a los cambios esto sera en menos detalle,

en más detalle tendremos un desarrollo, miembros del equipo del proyecto geográficamente dispersos, pruebas basadas en los requisitos, se requiere estimaciones con más precisión, se requiere trazabilidad de los requisitos

Los criterios de calidad:

- correcta: cada requisito satisface la necesidad o demanda legítima del negocio, es decir, debe trazar alguna necesidad o requisito de negocio.

- no contiene requisitos superfluos

- menos riesgo de scope creep y gold plating

Clara sin ambigüedad:

Tiene una interpretación única para todo público, el lenguaje natural casi siempre usado para describir requisitos, es inherente ambiguo

Consistente: No existen contradicciones entre los documentos de requisitos, sea en un mismo nivel o niveles diferentes.

La temporalidad REQ-03 indica que el evento A procede al evento B REQ-12 indica que los eventos A y B son simultáneos.

Dos requisitos utilizan diferentes nombres para el mismo objeto del mundo real

A menudo, la inconsistencia surge de solicitudes de cambios asimilados en la especificación.