

Ingeniería de software

Lectura 2



Unidad III : Planificación del proyecto de software.

PRESENTAN:

Ricardo Gomez Martínez.

DOCENTE:

EDUARDO FLORES GALLEGOS

50 Años de la Ingeniería de Software Problemas, Logros, Tendencias y Retos.

En este documento el autor nos habla de los principales problemas que se identificaron y a su vez los mayores logros alcanzados durante estos 50 años.

Problemas y Logros

Proceso de desarrollo de software

Problemas en 1968

- Falta una comprensión más completa del proceso de diseño del programa.
- Construimos sistemas como los hermanos Wright construyeron aviones:
“construye todo, empújalo por el precipicio, déjalo caer y vuelve a empezar.”

Logros hasta 2018

Durante los últimos 50 años hemos transitado por diferentes modelos: cascada, espiral, iterativo-incremental o ágil tipo Scrum. Hemos generado múltiples estándares para apoyar el desarrollo de software: CMM/CMMI, TSP/PSP, ISO/IEC 12207, 29110, 15504, MoProSoft, mps.Br o ESSENCE.

Calidad de software

Problemas en 1968

- Dificultades para lograr una fiabilidad suficiente en los sistemas, que cada vez son más integrados en las actividades centrales de la sociedad.
- Particularmente alarmante son las fallas, aparentemente inevitables de un software, que pueden impactar una cuestión de vida o muerte.

Logros hasta 2018

La calidad de software tiene dos vertientes: del proceso y del producto. En la primera hemos avanzado introduciendo conceptos y técnicas de verificación y validación. El aseguramiento de calidad y la mejora de procesos también son un avance.

En mi opinión esta parte es fundamental y va siendo más importante conforme pasan los años, esto se debe a que cada vez más cosas se pueden hacer con software y al usuario no le gustaría un error al momento tratarse de dinero involucrado.

Costos

Problemas en 1968

- T.J. Watson dijo que OS / 360 le costó a IBM más de 50 millones de dólares al año durante su preparación, y una inversión de al menos 5,000 años-hombre.
- Los costos de desarrollo del software estaban integrados con los costos de desarrollo del hardware. No se cuantificaba el valor y costo del software como tal.

Logros hasta 2018

Para justificar el costo del software tuvimos que empezar a medir el tamaño del software. El primer intento fue usar el número de líneas de código (LOCs), pero fue criticado porque software con la misma funcionalidad puede tener diferentes tamaños LOCs dependiendo del lenguaje e incluso estilo de programación. Luego tuvimos una propuesta más sensata con los Puntos de Función ajustados a ciertos

parámetros de complejidad, que eventualmente se convirtieron en el estándar ISO/IEC 14143, llamado de forma estelar “Puntos Funcionales Cosmic” :)

El problema de intangibilidad de software sigue causando estragos en la estimación de costos. La forma más popular es definir el precio en términos del costo por hora/hombre, pero hay gran discrepancia sobre cuál es la tarifa justa. Todavía es común encontrarse con quienes cuestionan por qué cuesta tanto el tiempo de las personas que desarrollan software, “si solo usan su cabeza y una computadora”.

Como nos dice el autor aun es un problema con el que nos topamos los programadores el querer estimar un costo justo para ambas partes, esta complicación aún se sigue.

En colusión los problemas y logros evolucionan conforme evolucionan las tecnologías de la información, con esto me refiero a que en cierto que en estos 50 años han sido solucionados grandes problemas, pero también han surgido más es como al programar al tratar de solucionar un error generamos 2 más.