



Definición de Software

Software, en general, es un set de programas y la documentación que acompaña.

- Existen tres tipos básicos de software. Estos son:
 - System software
 - Utilitarios
 - Software de Aplicación

Margaret Hamilton, lead software engineer, Project Apollo. Mostrando su código fuente..
<https://medium.com/@3fingeredfox/margaret-hamilton-lead-software-engineer-project-apollo-158754170da8>



¿Y dónde encontramos software?

3

5 Razones para no comparar software y manufactura

- ✓ El software es menos predecible
- ✓ No hay producción en masa, casi ningún producto de software es igual a otro.
- ✓ No todas las fallas son errores
- ✓ El software no se gasta
- ✓ El software no está gobernado por las leyes de la física ☺

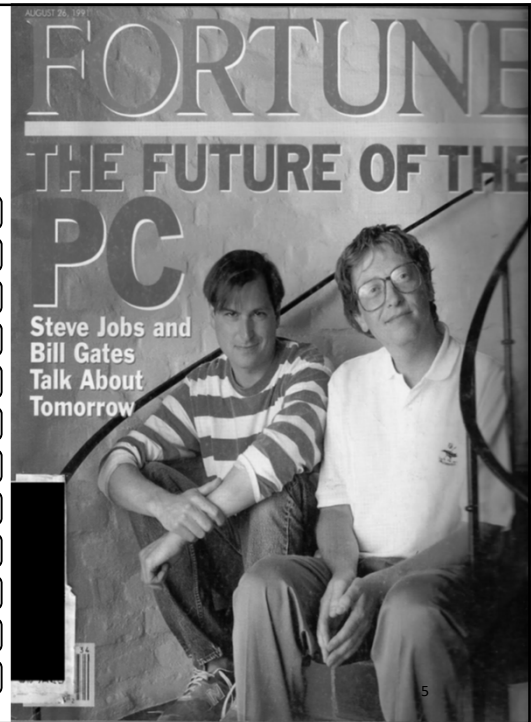
"The creation of software is an intellectual human endeavor. Creating good software relies on the personalities and the intellects of the members of the teams that create it. When applied to a different team of developers a process that delivers great software for one team of developers may fail to deliver anything at all for another team."

-- *The Practical Guide to S/W Arch.*



Un poco de historia

1968	• Nace el termino – conferencia de la NATO
1975	• The Mythical Man-Month – Frederick Brooks
1978	• Tom DeMarco introduce Structured Analysis
80	• Primeros grandes errores de software conocidos
1987	• No silver bullet (Brooks). Características esenciales del software
1989	• Managing the Software Process – Watts Humphrey
1990	• Internet / Object Oriented
1991	• CMM 1.0
1993	• CMM 1.1
2000	• CMMI 1.0
2001	• Agile Manifiesto
2003	• Lean Software Development



Algunos problemitas con el desarrollo de software

- La versión final del producto no satisface las necesidades del cliente.
- No es fácil extenderlo y/o adaptarlo. Agregar más funcional en otra versión es casi una misión imposible
- Mala documentación
- Mala calidad.
- Más tiempos y costos que los presupuestados

Y sigue
habiendo
errores...

Toyota deberá revisar 1,9 millones de autos Prius por problemas de software

Un error en el software que controla el funcionamiento del motor del auto podría hacer que se apague o funcione, pero a baja velocidad

Comentá 22

Tweet

Me gusta 28

Share

S+1



Toyota deberá afrontar la revisión de 1,9 millones de autos Prius por un error en el software que los controla. Foto: EFE

Toyota emitió ayer un llamado a la revisión de 1,9 millones de autos Prius de la tercera generación vendidos debido a un problema de software que provoca la caída de su sistema de control.

Alrededor de la mitad de los autos afectados están en Argentina y 713.000 en el resto del mundo. No se han reportado accidentes relacionados al problema.

Toyota dijo que el problema se relaciona con un módulo que forma parte del sistema de control del motor.

"La configuración del software podría causar una mayor tensión térmica en el interior del convertidor de elevación, y estos transistores podrían deformarse como consecuencia", dijo Toyota.

"Esto resultará en el encendido de varias luces de advertencia y problemas de funcionamiento del vehículo entre en su modalidad protegida", sostuvo la automotriz.

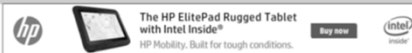
Agregó que en esa modalidad el coche puede conducirse pero a menor potencia.

En pocos casos, el sistema híbrido podría apagarse, lo que causaría que el vehículo se detenga momentáneamente mientras esté siendo conducido, dijo Toyota.

Esta reciente medida sería el tercer llamado a revisión para el Prius actual, después de un llamado en junio del año pasado debido a problemas de frenado. El Prius es uno de los modelos más vendidos de Toyota y se ha convertido en un sinónimo de la tecnología híbrida de bajo consumo.

Agencia Reuters

ANNOUNCEMENT The Next Web is coming back to New York. Learn more here.



Chrysler is recalling 1.4 million vehicles after a Jeep was remotely hacked

by NAPIER LOPEZ 3d ago in INSIDER



All Time Favorites

- The complete guide to backing up your computer properly
- How to use personal drones legally: A beginner's guide
- 11 Android apps to make notifications more interesting
- How to raise prices without ruining your business
- How mobile is bridging brick and mortar's data gap

<http://thenextweb.com/insider/2015/07/24/chrysler-is-recalling-1-4-million-vehicles-after-a-jeep-was-remotely-hacked/>

<http://www.lanacion.com.ar/1663594-toyota-debera-revisar-19-millones-de-autos-prius-por-problemas-de-software>

7

Y cuando nos va bien es por...

1. Involucramiento del usuario 15.9 %
2. Apoyo de la Gerencia 13.0 %
3. Enunciado claro de los requerimientos 9.6 %
4. Planeamiento adecuado 8.2 %
5. Expectativas realistas 7.7 %
6. Hitos intermedios 7.7 %
7. Personas involucradas competentes 7.2 %

8

Y cuando nos va mal es por...

1. Requerimientos incompletos 13.1 %
2. Falta de involucramiento del usuario 12.4 %
3. Falta de recursos 10.6 %
4. Expectativas poco realistas 9.3 %
5. Falta de apoyo de la Gerencia 8.7 %
6. Requerimientos cambiantes 8.1 %



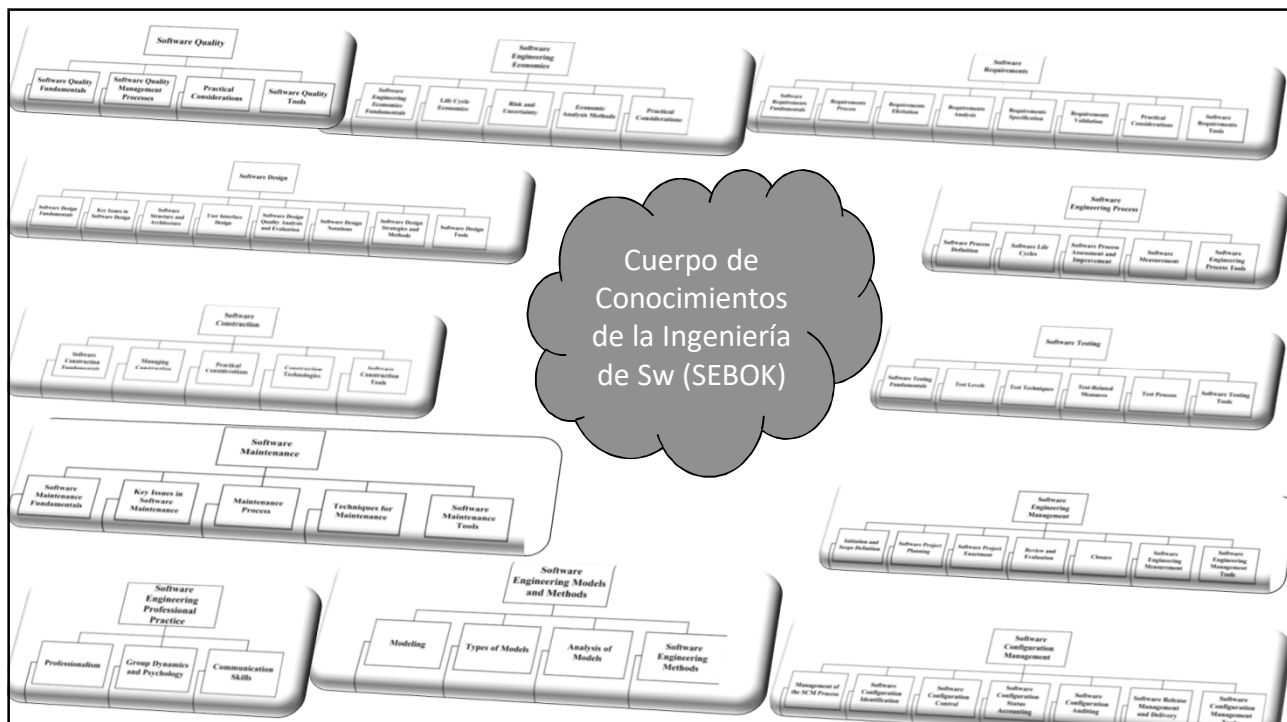
Conclusión....

**Saber programar
NO es
suficiente!!!!**

Ingeniería de Software

- Parmas [1987] definió a la ingeniería en software como “multi-person construction of multi-version software”

11





- Cuerpo de Conocimiento de la Ingeniería de Software
- Versión 3.0 del 2014 de la IEEE
- Está conformado por 15 áreas de conocimiento

Software Requirements
Software Design
Software Construction
Software Testing
Software Maintenance
Software Configuration Management
Software Engineering Management
Software Engineering Process
Software Engineering Models and Methods
Software Quality
Software Engineering Professional Practice
Software Engineering Economics
Computing Foundations
Mathematical Foundations
Engineering Foundations

13

Ingeniería de Software: la materia en contexto



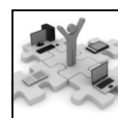
Disciplinas Técnicas

- Requerimientos
- Análisis y Diseño
- Construcción
- Prueba
- Despliegue



Disciplinas de Gestión

- Planificación de Proyecto
- Monitoreo y Control de Proyectos



Disciplinas de Soporte

- Gestión de Configuración de Software
- Aseguramiento de Calidad
- Métricas

14

Referencias Bibliográficas Principales

■ Libros:

- Sommerville, Ian - INGENIERÍA DE SOFTWARE - Novena Edición (Editorial Addison-Wesley Año 2011). -
- Brooks, Frederick The mythical man-month (anniversary ed.), 1995 Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©1995
- Steve Mc Connell - Rapid Development Redmond, Wa.: Microsoft Press,
- Cohn, Mike – Agile Estimation and Planning – Editorial Prentice Hall 2006
- Cohn, Mike - User Stories Applied – Editorial Addison Wesley 2004.

■ Papers:

- Orphans Preferred (<http://www.stevemccconnell.com/psd/07-orphanspreferred.htm>)
- No Silver Bullet (<http://www.virtualschool.edu/mon/SoftwareEngineering/BrooksNoSilverBullet.html>)
- Software's Ten Essentials (<http://www.stevemccconnell.com/ieeesoftware/10Essentials.pdf>)
- <http://www.scrumguides.org/download.html>
- Dean Leffingwell and Pete Behrens – A user story primer (2009)
- Manifiesto Ágil <http://agilemanifesto.org/iso/es/>

