

Ingeniería de Software



Universidad Tecnológica Nacional
Facultad Regional Córdoba
Cátedra de Ingeniería de Software
Docentes: Judith Meles – Laura Covaro

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>

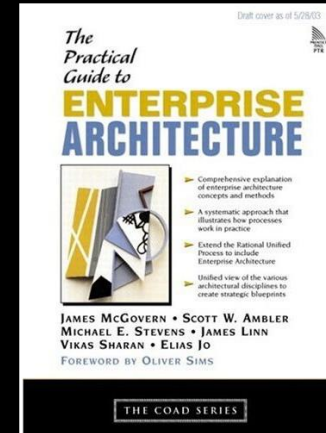


¿Y dónde encontramos software?

5 Razones para no comparar software y manufactura

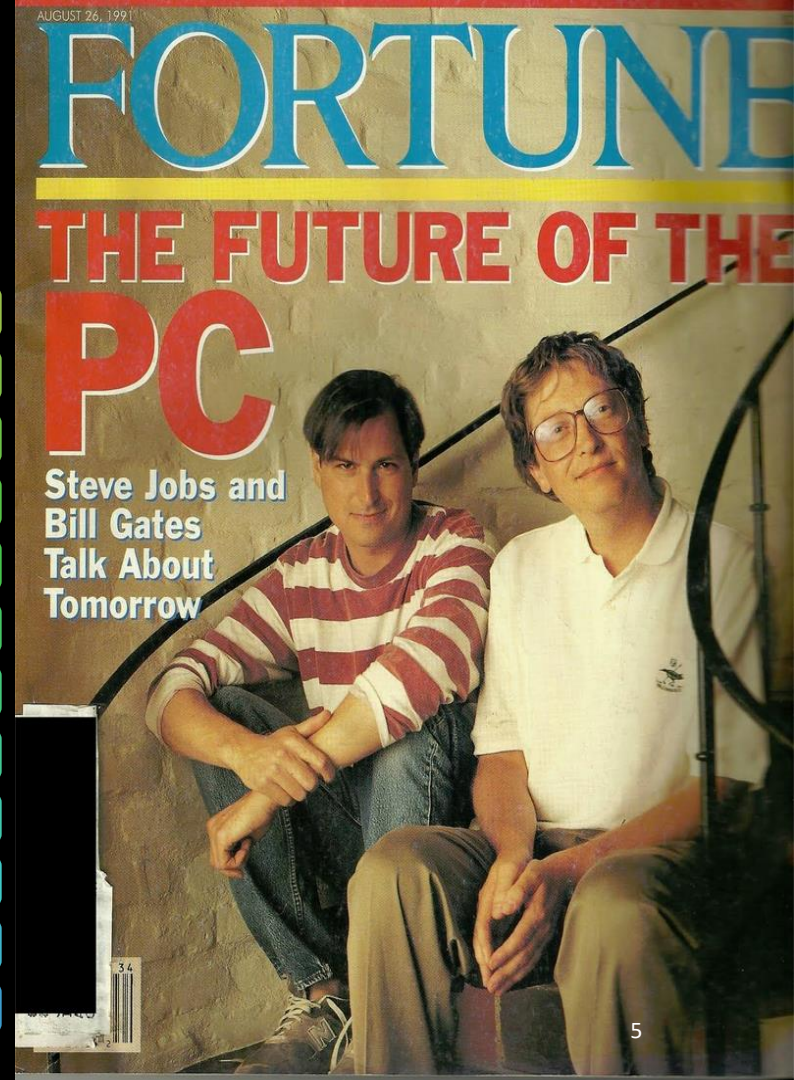
- ✓ El software es menos predecible
- ✓ No hay producción en masa, casi ningún producto de software es igual a otro.
- ✓ No todas las fallas son errores
- ✓ El software no se gasta
- ✓ El software no esta gobernado por las leyes de la física 😊

"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



GLOBAL TOP 5 BRANDS | TECH ASCENDANCY

RANK

1

2

3

4

5

2006



2012



2017



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...

Jueves 13 de febrero de 2014 | 01:44

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 25

Share

+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 comunicado en el que anunció que 1,9 millones de autos Prius de la tercera generación vendidos en los Estados Unidos debido a un problema de software que provoca la caída de su sistema de control de velocidad.

Alrededor de la mitad de los autos afectados están en Japón y 713.000 en los Estados Unidos, dijo un portavoz de Toyota. No se han reportado accidentes relacionados al problema.

Toyota dijo que el problema afecta a un módulo que forma parte del sistema de control de velocidad.

"La configuración del software podría causar una mayor tensión térmica en el 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 posiblemente 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...



The HP ElitePad Rugged Tablet with Intel Inside®
HP Mobility. Built for tough conditions.

Buy now



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



by NAPIER LOPEZ Tweet — 3d ago in INSIDER



<http://tnw.to/c4k1w>

0 Comments

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/>

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 %

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 %

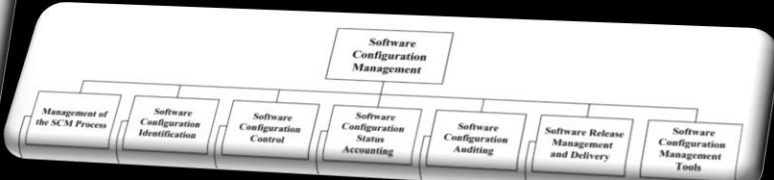
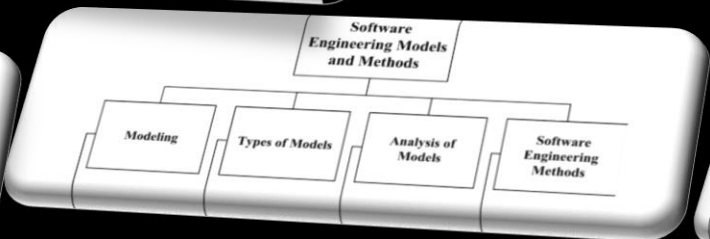
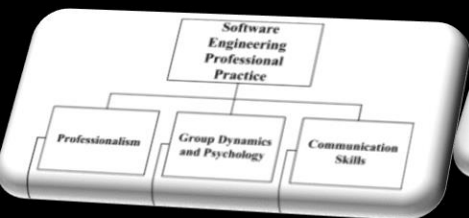
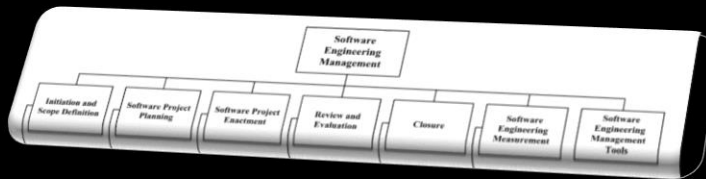
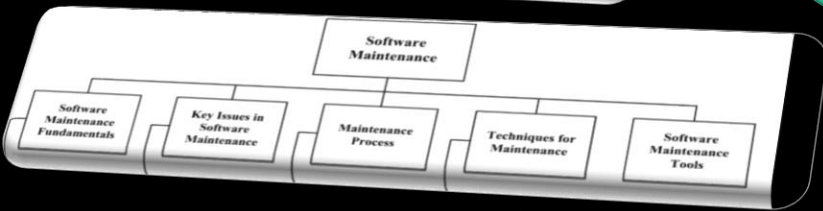
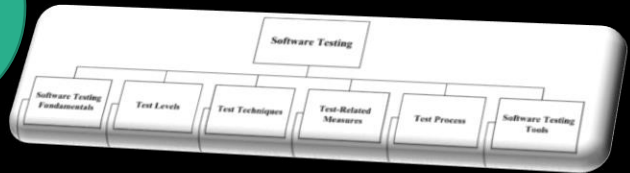
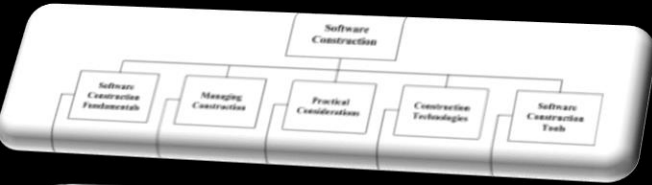
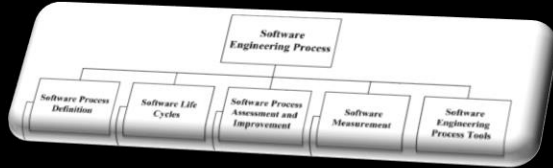
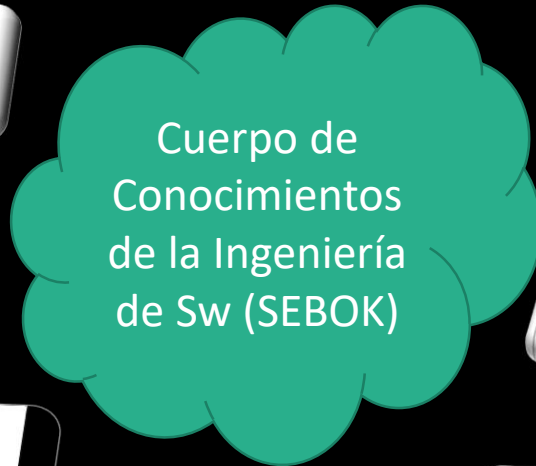
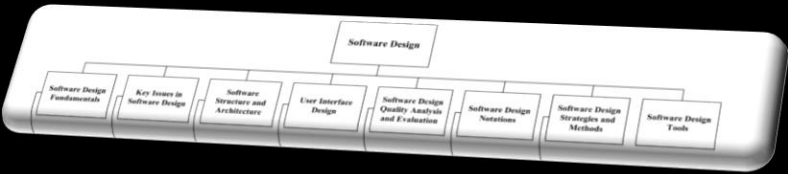
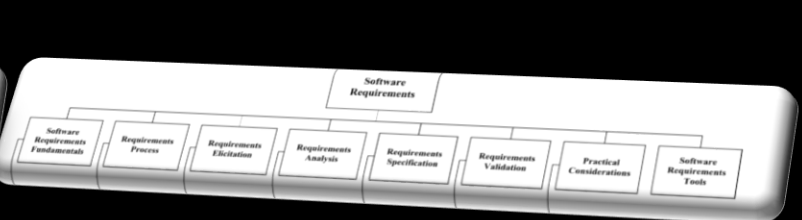
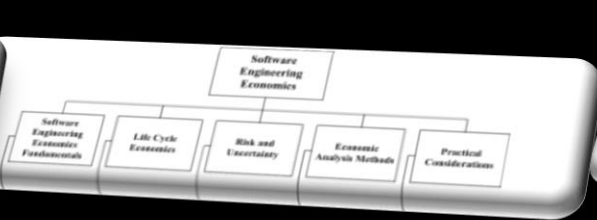


Conclusión....

Saber programar
NO es
suficiente!!!!

Ingeniería de Software

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





- 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

Ingeniería de Software: la materia en contexto



Disciplinas Técnicas

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



Disciplinas de Gestión

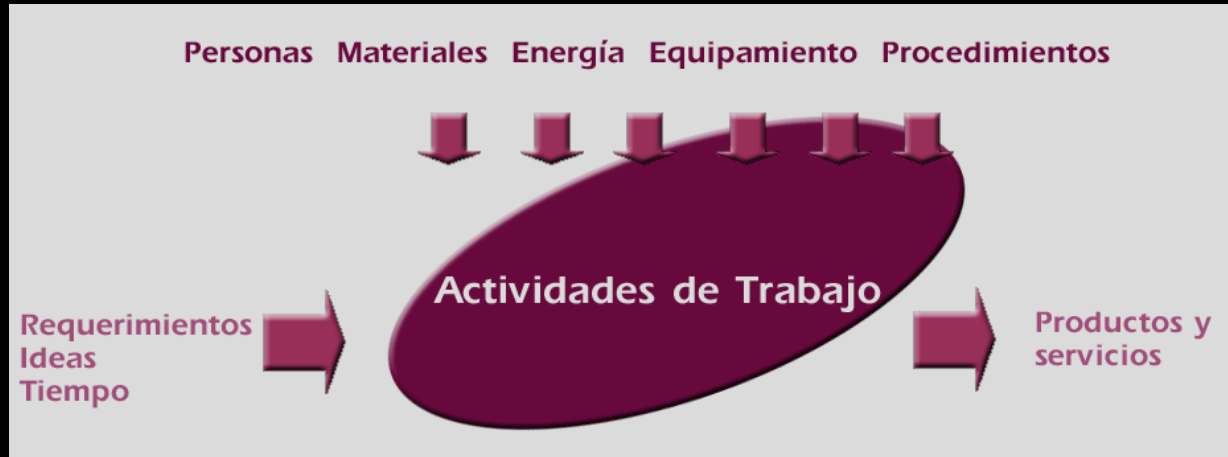
- **Planificación de Proyecto**
- **Monitoreo y Control de Proyectos**



Disciplinas de Soporte

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

El proceso de Software

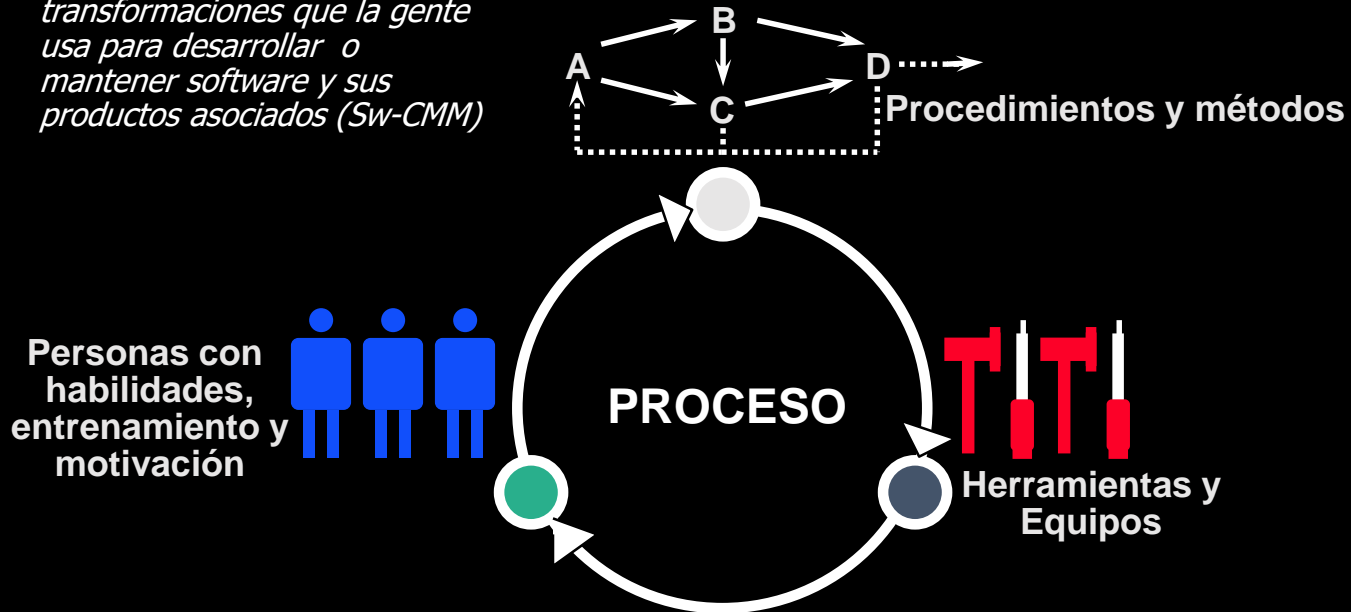


- Conjunto estructurado de actividades para desarrollar un sistema de software
- Estas actividades varían dependiendo de la organización y el tipo de sistema que debe desarrollarse.
- Debe ser explícitamente modelado si va a ser administrado.

Definición de un Proceso de Software

Proceso: La secuencia de pasos ejecutados para un propósito dado (IEEE)

Proceso de Software: Un conjunto de actividades, métodos, prácticas, y transformaciones que la gente usa para desarrollar o mantener software y sus productos asociados (Sw-CMM)



Definido vs. Empírico



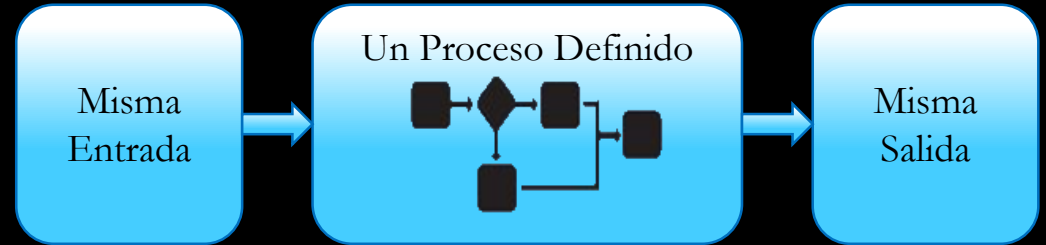
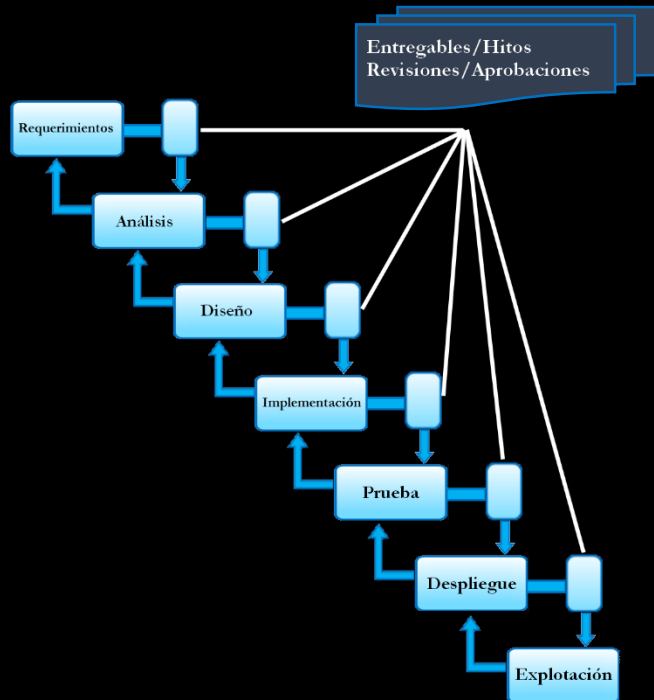
En la Universidad de California, en Irvine, simplemente sembraron pasto y esperaron 1 año, luego de eso se fijaron donde la gente había hecho “caminito”, entonces ahí construyeron las sendas peatonales

Definido (inspirados en las líneas de producción)

- Asume que podemos repetir el mismo proceso una y otra vez, indefinidamente, y obtener los mismos resultados.
- La administración y control provienen de la predictibilidad del proceso definido.



Procesos Definidos



Procesos Empíricos

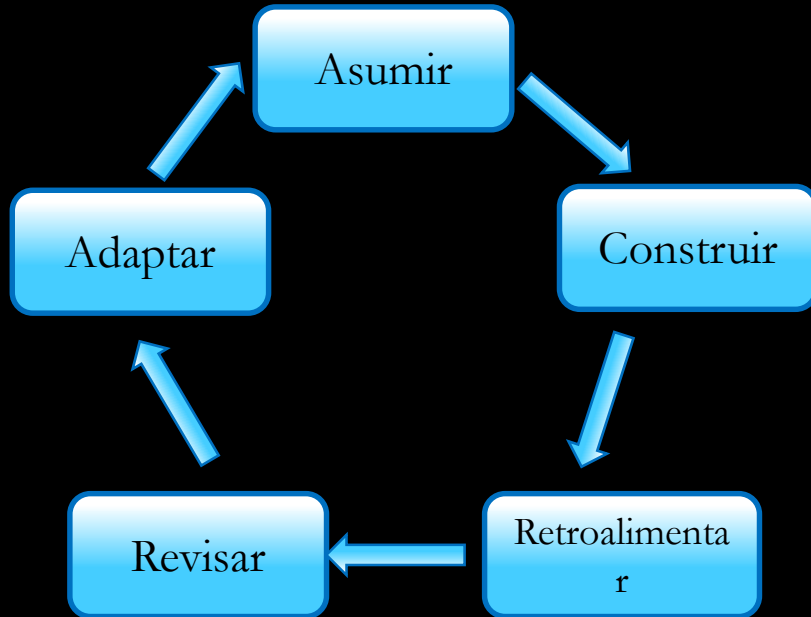


Empírico

- Asume procesos complicados con variables cambiantes. Cuando se repite el proceso, se pueden llegar a obtener resultados diferentes.
- La administración y control es a través de inspecciones frecuentes y adaptaciones
- Son procesos que trabajan bien con procesos creativos y complejos. (a que suena??)



Patrón de conocimiento en procesos empíricos



Ciclos de vida

La serie de pasos a través de los cuales el producto o proyecto progresa

Ciclos de Vida

- Un ciclo de vida de un proyecto software es un representación de un proceso. Grafica una descripción del proceso desde una perspectiva particular
- Los modelos especifican
 - Las fases de proceso.
 - Ejemplo: requerimientos, especificación, diseño...
 - El orden en el cual se llevan a cabo

Clasificación de los ciclos de vida

- Hay tres tipos básicos de Ciclos de Vida
 - Secuencial
 - Iterativo/Incremental
 - Recursivo (sólo se lo nombra, no vamos a profundizar)

De ciclos de vida hay más?

- Si! Capítulo 7 de Desarrollos de proyectos informáticos (Rapid Development) de Mcconell

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.stevemcconnell.com/psd/07-orphanspreferred.htm>)
- No Silver Bullet (<http://www.virtualschool.edu/mon/SoftwareEngineering/BrooksNoSilverBullet.html>)
- Software's Ten Essentials (<http://www.stevemcconnell.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/>

