Metodologías de Desarrollo 620510

Introducción

Elizabeth Grandón Toledo, Ph.D.
Departamento de Sistemas de Información
2021 - 1

UNIDAD 1: INTRODUCCIÓN

- Generalidades - Sw
- Ingeniería vs. Artesanía
- Métodos y modelos de desarrollo

SOFTWARE

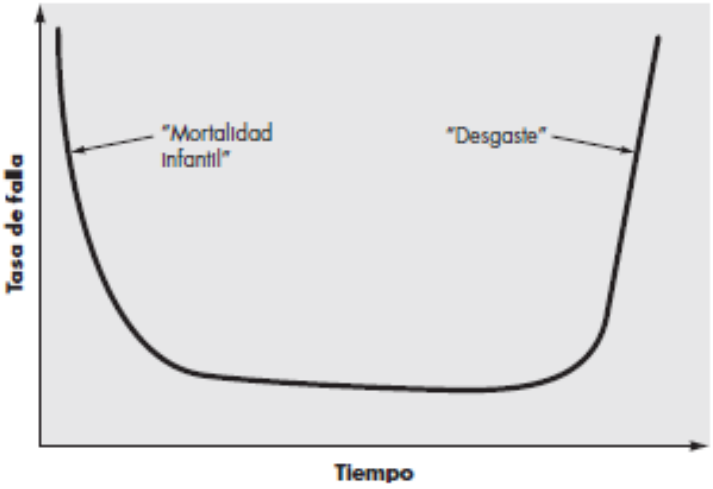
a) Creación intelectual que comprende los programas, los procedimientos, las reglas y cualquier documentación perteneciente a la operación del sistema de procesamiento de datos (ISO 9000-3: 1991,3.1).

b) Conjunto completo de programas, procedimientos, documentación y datos asociados, diseñados para entregar al usuario (ISO 9000-3: 1991, 3.2).

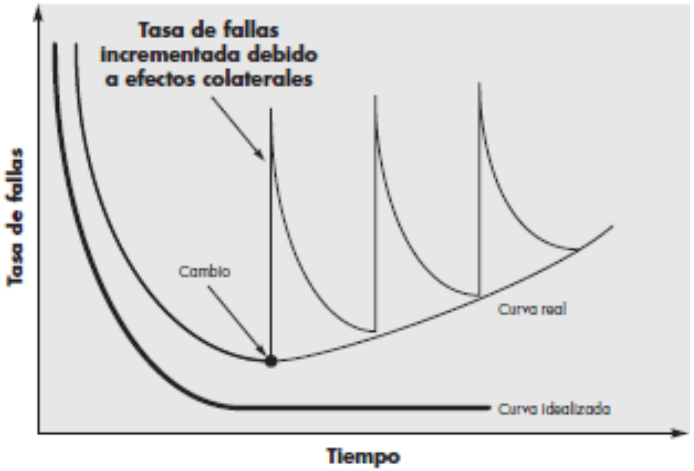
c) 1) Instrucciones (programas de cómputo) que cuando se ejecutan proporcionan las características, función y desempeño buscados. 2) Estructuras de datos que permiten que los programas manipulen en forma adecuada la información, y 3) información descriptiva tanto en papel como en formas virtuales que describen la operación y uso de los programas (Pressman, 2010)

SOFTWARE VS HARDWARE

El Sw se desarrolla con el intelecto; no se manufactura en el sentido clásico
El sw no se desgasta
El sw se está moviendo hacia la reutilización de componentes



Curva de fallas del Hw



Curva de fallas del Sw

Fuente: Pressman (2010)

- Algunos dominios de aplicación del Sw

a) Software de sistema:

Es todo aquello que se denomina "sistema operativo". Entre ellos se encuentran Windows, Mac OS, Ubuntu (entre otras distribuciones de Linux), Unix, Fedora y Solaris, entre otros.
Es el software principal de un sistema informático y se encarga de gestionar tanto los recursos de hardware como los programas de aplicación.

El software de sistema es el software principal de un sistema informático.

Fuente: https://www.caracteristicas.co/software/#

TIPOS DE SW (cont.)

b) Software de programación:

Son aquellas aplicaciones y herramientas que utilizan los programadores para desarrollar nuevo software. Entre ellos se encuentran editores de texto, compiladores, intérpretes, enlazadores, depuradores y entornos de desarrollo integrados. Cada una de estas herramientas de programación puede ser utilizada con uno o más lenguajes de programación.

Fuente: https://www.caracteristicas.co/software/#

TIPOS DE SW (cont.)

c) Software de aplicación:

Es un tipo de sw de computadora diseñado para realizar un grupo de funciones, tareas o actividades coordinadas para el beneficio del usuario. Programas aislados que resuelven una necesidad específica de negocios. Las aplicaciones en esta área procesan datos comerciales o técnicos en una forma que facilita las operaciones de negocios o la toma de decisiones administrativas o técnicas. Ejemplos: procesador de textos, una hoja de cálculo, una aplicación de contabilidad, un navegador web, un reproductor multimedia, un simulador de vuelo aeronáutico, una consola de juegos o un editor de fotografías.

Interacción

¿Qué experiencia ha tenido con el desarrollo de alguna aplicación o sistema?

Evolución:

- Primera Era: Años 50
  - Programas con ensamblador
  - Funciones matemáticas
- Epoca de transición : 60
  - Crisis del software
- Segunda Era : 70's
  - Aparición de computadoras más potentes
  - Software de uso general, fuerte mantenimiento
  - No existe un conocimiento detallado de la estructura interna de los programas

Evolución (cont.):


La IBM PC original (1981)

- Tercera Era: 80's
  - Marcada por PC's
  - Disminución de precios
  - Programación estructurada
  - Reducción del mantenimiento
- Cuarta Era
  - Lenguajes de cuarta generación (ej. SQL de Oracle)
  - Prog. concurrente con más de un procesador
  - Lenguajes orientados a objetos
  - Mejores herramientas, pero mayor complejidad

Actividad 1 (opcional)

Visualice los siguientes videos acerca del software y desarrolle una línea de tiempo de su evolución

Historia y evolución del Software

https://www.youtube.com/watch?v=oSssWHD1oSI (3:50 min)

Historia del Software

https://www.youtube.com/watch?v=hUtJ0FIbZFU (3:36)

Características deseables de un producto de Software

Corrección. Que cumpla con su objetivo

Usabilidad. Que sea fácil de aprender

Seguridad. Que sea resistente a ataques externos

Flexibilidad. Que pueda ser modificado por los desarrolladores

Portabilidad. Que pueda ser utilizado en diversos equipos (hw)

Fuente: https://www.caracteristicas.co/software/#ixzz6Kn0tQn21

A pesar de las características deseables,

el Sw falla…..

Dinero: Knight Capital

En agosto de 2012, un error de programa casi provocó la quiebra de la empresa de inversión Knight Capital. La compañía perdió 500 millones de dólares en media hora debido a que sus computadoras comenzaron a comprar y vender millones de acciones sin ningún tipo de control humano. Como resultado, el precio de las acciones de Knight Capital cayó un 75% en dos días.

Medicina: radioterapia

Un error de programación de la unidad de control de la máquina de radioterapia Therac-25 causó entre 1985 y 1987 al menos seis accidentes en los que los pacientes recibieron sobredosis masivas de radiación. Al menos tres de estos pacientes fallecieron como consecuencia directa del exceso de radiación. Los expertos creen que el fallo fue causado por un error en el código que obligó al programa a realizar la misma acción varias veces.

- Consecuencias catastróficas de falla de Sw

Infraestructura: apagón en el noreste de EE.UU.
En agosto del 2003 varios estados del noreste de EE.UU. y la provincia canadiense de Ontario se quedaron sin luz debido a un corte de energía resultado de un accidente local. El accidente pasó desapercibido a causa de un fallo del software de vigilancia del funcionamiento de General Electric Energy y provocó una cadena de errores.

- Consecuencias catastróficas de falla de Sw

Transporte: Aerolínea American Airlines

En 2013, un error de programación provocó el caos en la compañía American Airlines. La unión de dos sistemas como resultado de la fusión de varias compañías aéreas originó un fallo en el sistema de reserva de pasajes. Concretamente, el problema surgió con toda probabilidad cuando se intentó unificar plataformas escritas en diferentes lenguajes de programación.

Interacción

Mencione algún otro ejemplo de falla de sw que usted conozca

Jamboard

Actividades

1. En clase: Lluvia de ideas - Cuáles cree usted son los factores críticos de éxito de un proyecto de desarrollo de sw? Utilice el Jamboard indicado por la profesora

2. Lea el Informe de Caos y escriba un resumen (opcional- disponible en Moodle)

Porqué la crisis del Sw? (caos?)

- Problemas para estimar tiempo, esfuerzo y costos de los sistemas
- Insatisfacción y desconfianza del cliente, debido a la falta de control de calidad
- No hay mantenimiento. Detección de un fallo implica sustitución de módulos ➔ altos costos de mantención
- No hay documentación

Y esto no es nuevo….

Según un estudio sobre proyectos realizado en 1979 en USA, se demostró que:

- 2% funcionaron
- 3% funcionaron después de algunas correcciones
- 45% fueron entregados y nunca pudieron ser usados
- 20% se usaron después de ser casi devueltos a hacer o fueron abandonados
- 30% fueron pagados y nunca entregados

Qué está pasando hoy día?
The Caos Report 2015 – The Standish Group

- El Reporte se publica cada dos años desde 1994 para mostrar el estado del desarrollo de software en la industria
- El 2015 incluye una muestra de 50.000 proyectos en todo el mundo
- Incluye proyectos que involucran pequeñas mejoras a sistemas masivos
- Los factores críticos de éxito se relacionan con el tiempo de desarrollo (on time) y el presupuesto (on budget) con resultados satisfactorios.

Fuente: https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

The Caos Report 2015

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS

The Caos Report 2015

CHAOS RESOLUTION BY PROJECT SIZE

	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	**100%**	**100%**	**100%**

The resolution of all software projects by size from FY2011–2015 within the new CHAOS database.

Interacción

¿Qué puede deducir respecto de las fallas y éxito de los proyectos de sw dependiendo de su tamaño?

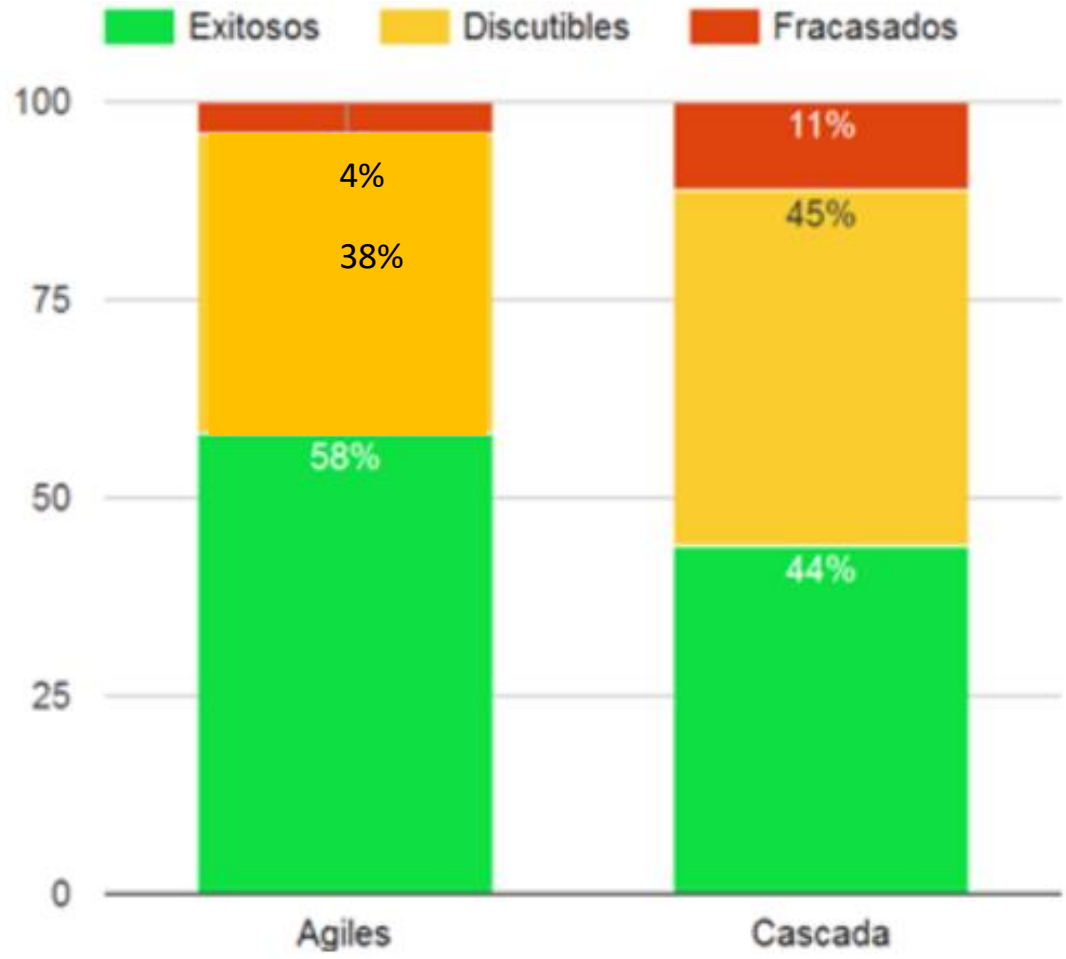
The Caos Report 2015

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

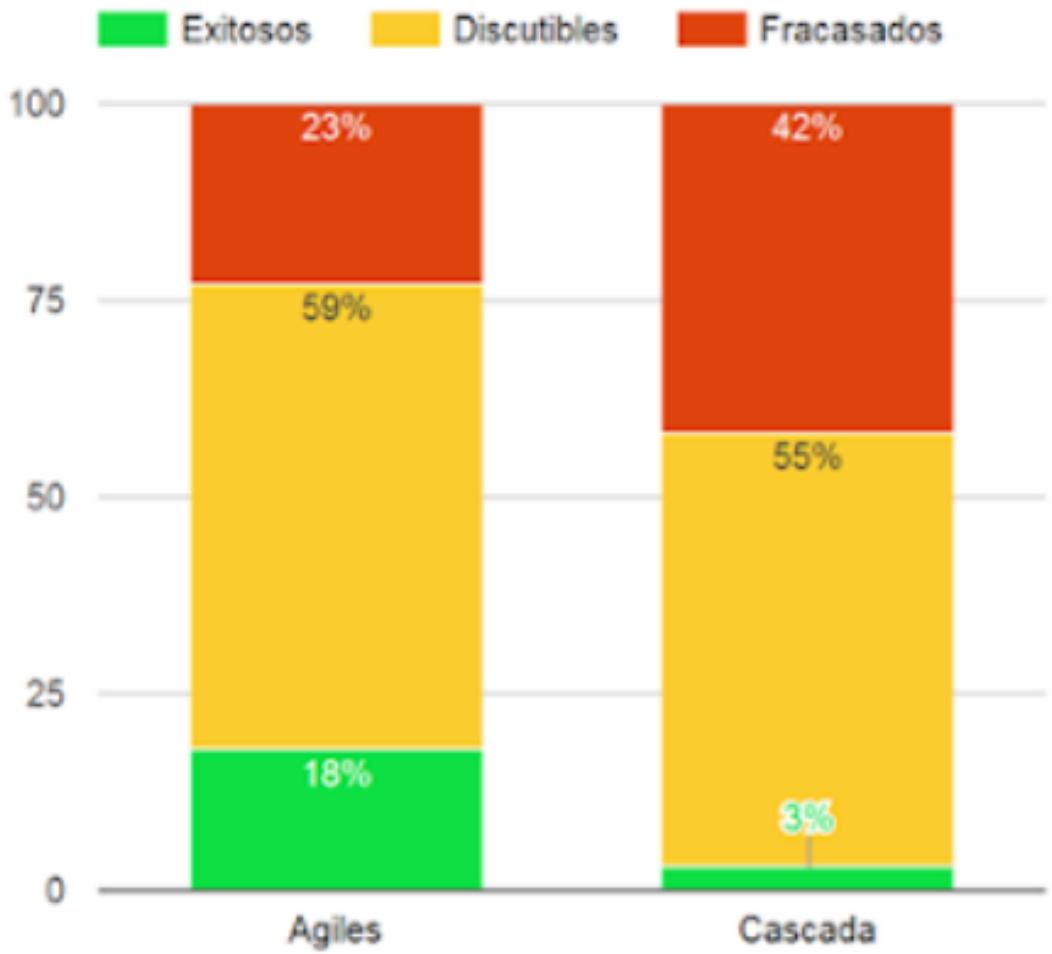
SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000

Proyectos pequeños por tipo de Metodología



Proyectos grandes por tipo de Metodología



Interacción

¿Qué puede deducir respecto de las fallas y éxito de los proyectos de sw dependiendo del tipo de metodología empleada en su desarrollo?

¿Y los factores críticos de éxito?

¿Qué arroja el Informe de Caos?

- GENERALIDADES

The Caos Report 2015

CHAOS FACTORS OF SUCCESS

FACTORS OF SUCCESS	POINTS	INVESTMENT
Executive Sponsorship	15	15%
Emotional Maturity	15	15%
User Involvement	15	15%
Optimization	15	15%
Skilled Resources	10	10%
Standard Architecture	8	8%
Agile Process	7	7%
Modest Execution	6	6%
Project Management Expertise	5	5%
Clear Business Objectives	4	4%

PRINCIPALES FACTORES PARA EL ÉXITO DE LOS PROYECTOS

- **Apoyo Ejecutivo:** el apoyo del equipo ejecutivo debe incluir tanto aspectos financieros como emocionales. Este apoyo es clave para estimular al equipo y lograr el éxito del proyecto.
- **Madurez Emocional**: la definición psicológica que se le da a la madurez emocional es el "ser capaz de aceptar la realidad de las personas y cosas tal cual son". En cualquier grupo, equipo u organización los resultados dependen de las habilidades individuales pero también de los vínculos que se establecen en el equipo.
- **Implicación de los Usuarios**: se trata de lograr la implicación de los usuarios tanto en la toma de decisión como en los procesos de toma de información. Incluye la incorporación de feedback, revisión de requisitos, prototipado, etc.

Executive Support: when an executive or group of executives agrees to provide both financial and emotional backing. The executive or executives will encourage and assist in the successful completion of the project.

Emotional maturity is the collection of basic behaviors of how people work together. In any group, organization, or company it is both the sum of their skills and the weakest link that determine the level of emotional maturity.

User Involvement: takes place when users are involved in the project decision-making and information-gathering process. This also includes user feedback, requirements review, basic research, prototyping, and other consensus-building tools.

Optimization is a structured means of improving business effectiveness and optimizing a collection of many small projects or major requirements. Optimization starts with managing scope based on relative business value.

Skilled staff are people who understand both the business and the technology. A skilled staff is highly proficient in the execution of the project's requirements and deliver of the project or product.

- GENERALIDADES

SAME is Standard Architectural Management Environment. The Standish Group defines SAME as a consistent group of integrated practices, services, and products for developing, implementing, and operating software applications.

Agile proficiency means that the agile team and the product owner are skilled in the agile process. Agile proficiency is the difference between good agile outcomes and bad agile outcomes.

Modest execution is having a process with few moving parts, and those parts are automated and streamlined. Modest execution also means using project management tools sparingly and only a very few features.

Project management expertise is the application of knowledge, skills, and techniques to project activities in order to meet or exceed stakeholder expectations and produce value for the organization.

Clear Business Objectives is the understanding of all stakeholders and participants in the business purpose for executing the project. Clear Business Objectives could also mean the project is aligning to the organization's goals and strategy.

Entonces…

- La industria del software no ha acabado de salir de la fase artesanal (costos concentrados en mantención)
- Dedicamos nuestros esfuerzos de hoy a arreglar lo que se hizo mal ayer
- ”Prisa patológica”, estimaciones no realistas (falta de planificación), desorganización nos lleva:
  - Procesos software normalmente improvisados
  - Si se han especificado, no se siguen rigurosamente
  - Organización reactiva (resolver crisis inmediatas)
  - Recorte de revisiones , pruebas y verificaciones del software
- Si hay plazos rígidos $\Rightarrow$ se sacrifican funcionalidad y calidad del producto para satisfacer el plan
- No existen bases objetivas para juzgar la calidad del producto

GENERALIDADES

Entonces…

- El 90 % de los proyectos no alcanzan los objetivos
- El 40 % fracasan por completo
- El 29 % no se entregan nunca
- Costos de producción del software
  - 10 % en especificar...
  - 30 % en desarrollar...
  - 60 % en PROBAR Y MANTENER!!

INGENIERÍA VS. ARTESANIA

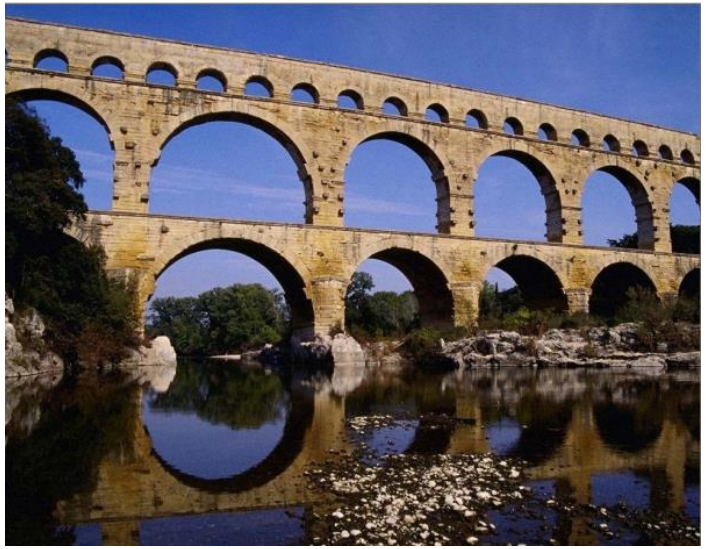
Que se requiere…

- Producir software de calidad
- Reutilización de experiencia acumulada de proyectos
- Enfasis en la adopción de las ”mejores prácticas” reconocidas por la industria, incorporándolas a los procesos de desarrollo

Artesano → Ingeniero

INGENIERÍA VS. ARTESANIA

Entonces…



"The Roman bridges of antiquity were very inefficient structures. By modern standards, they used too much stone, and as a result, far too much labour to build. Over the years we have learned to build bridges more efficiently, using fewer materials and less labour to perform the same task."

-Tom Clancy (The Sum of All Fears)

Porqué los puentes no se caen y se fabrican on-time y on-budget…

- Se realiza un proceso de diseño extremadamente detallado
- El diseño se "congela" y el constructor no tiene ninguna flexibilidad de cambiar las especificaciones
- Sin embargo, hoy día en el mundo de los negocios cada vez más rápido no se puede "congelar" un diseño ➔ flexibilidad
- Además de los 3000 años de experiencia en construcción de puentes, si un puente se cae se investigan las causas y se genera un informe ad-hoc. Esto no sucede en la industria del sw!

Entonces…

La ingeniería de software es: 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software.
2) El estudio de enfoques según el punto 1 (IEEE, citada en Pressman (2010)

Ingeniería de Software: área de la informática o ciencias de la computación que ofrece métodos y técnicas para desarrollar y mantener software de **calidad** para todo tipo de sistemas.

Fuente: Pressman (2010)

Un *proceso* es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo.

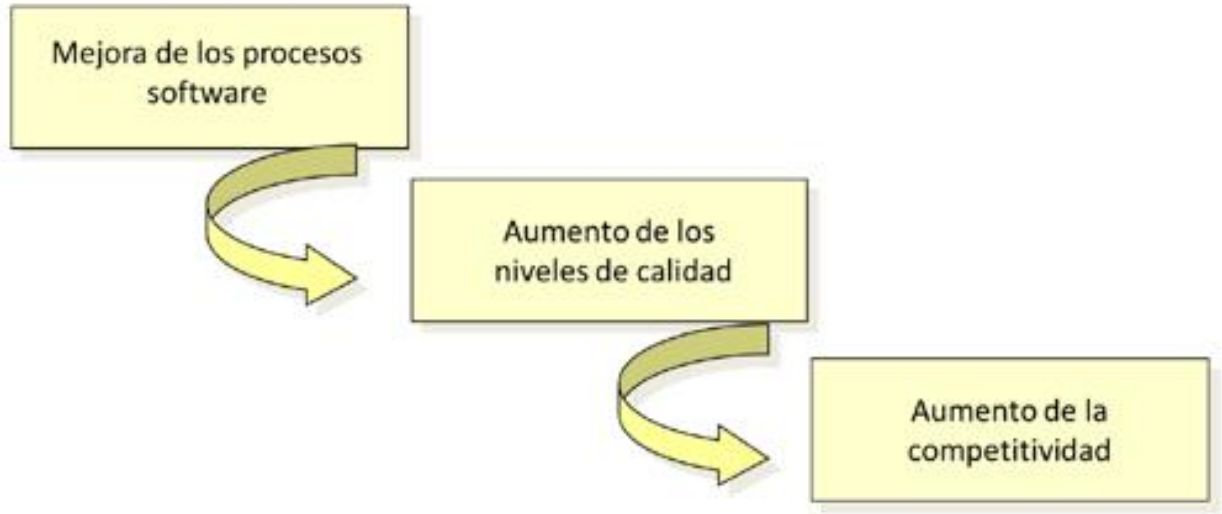
Una ***actividad*** busca lograr un objetivo amplio (por ejemplo, comunicación con los participantes) y se desarrolla sin importar el dominio de la aplicación, tamaño del proyecto, complejidad del esfuerzo o grado de rigor con el que se usará la ingeniería de software.
Una ***acción*** (diseño de la arquitectura) es un conjunto de tareas que producen un producto importante del trabajo (por ejemplo, un modelo del diseño de la arquitectura).
Una ***tarea*** se centra en un objetivo pequeño pero bien definido (por ejemplo, realizar una prueba unitaria) que produce un resultado tangible.

En el contexto de la ingeniería de software, un proceso *no* es una prescripción rígida de cómo elaborar software. Por el contrario, es un enfoque adaptable que permite que las personas que hacen el trabajo (el equipo de software) busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usarán.

Fuente: Pressman (2010)

Qué debemos hacer?

Mejorar los procesos de desarrollo para obtener como resultado mejores productos de software



Qué necesitan las organizaciones?

- Definir las actividades necesarias en el desarrollo de aplicaciones (sistemas de información)
- Mantener una coherencia entre todos los proyectos de la organización
- Alinear los sistemas de información a las estrategias de la organización
- Introducir puntos de control para realizar revisiones y controles de calidad
- Investigación de paradigmas o modelos de desarrollo

Ciclo de vida del Software

"Marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de sw, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso" (Norma ISO 12207-1)

Definiciones

Método

- Es un proceso disciplinado para generar un conjunto de modelos que describen varios aspectos de un sw en desarrollo, utilizando una notación bien definida (Booch). Responde al Qué?

Técnica

- Es una forma particular de desarrollar una tarea. Se relaciona con la habilidad de hacer. La tecnología incorpora el conocimiento científico a la técnica. Responde al Cómo?

Herramienta

- Es el medio que permite aplicar una o más técnicas. Por ejemplo herramientas de diagramación, generadores de código, compiladores, depuradores, etc.

Metodología

- Es un conjunto coherente de métodos y técnicas que cubren mas de una etapa del ciclo de vida

Paradigmas o Modelos de Desarrollo:

- Los paradigmas o modelos de desarrollo de sw son estrategias de desarrollo para organizar las diversas etapas y actividades del ciclo de vida del sw (ej. lineales, iterativos, ágiles, etc.)
- Describe las transiciones entre etapas, especificando qué actividades desarrollar en cada momento
- La selección de un modelo o paradigma específico dependerá de la naturaleza del proyecto y/o aplicación, los métodos, las herramientas a utilizar, los controles y las entregas que se requieran

Fases GENERALES de los modelos de desarrollo:

El trabajo asociado a la ingeniería del Software puede dividirse en tres fases fundamentales, independientemente del área de aplicación:

- FASE DE DEFINICIÓN
- FASE DE DESARROLLO
- FASE DE MANTENIMIENTO

Los paradigmas de la ingeniería de software conservan estas fases generales, variando los métodos, las herramientas y procedimientos para aplicarlos

Fase de DEFINICIÓN:

- Qué información ha de ser procesada
- Qué función y rendimiento se desea
- Qué comportamiento del sistema
- Qué interfaces van a ser establecidas
- Qué restricciones de diseño existen
- Qué criterios de validación se necesitan para definir

Dependiendo del paradigma o modelo se definen un conjunto específico de actividades, pero las tareas principales serán: ingeniería de sistema o de información, planificación del proyecto del software, y análisis de los requisitos

Fase de DESARROLLO:

- Cómo han de diseñarse las estructuras de datos,
- Cómo ha de implementarse la función como una arquitectura del software
- Cómo han de caracterizarse las interfaces
- Cómo ha de traducirse el diseño en un lenguaje de programación
- Cómo ha de realizarse la prueba

Las tareas principales serán: diseño del software, generación de código y prueba del software

Fase de MANTENIMIENTO:

Fase centrada en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software, y a cambios producidos por los requisitos cambiantes del software.

Cuatro tipos de cambio: Corrección, Adaptación (Cambio de sistema Operativo, reglas de la empresa,etc.), Mejora (evolutivo), Prevención

Actividades a realizar: Gestión de riesgos, revisiones técnicas formales, mediciones, garantia de calidad del software, seguimiento y gestion del proyecto de software, gestión de reutilización.

Bibliografía

- PRESSMAN, R. Ingeniería del Software, un enfoque práctico. Editorial McGraw Hill. 7ma edición. 2010.
- SOMMERVILLE, I. Ingeniería de software. Editorial Pearson Educación, 8va edición. 2011
- The Chaos Report – Standish Group
- Uriarte, J. M. "Software Caracteristicas.co". Última edición: 13 de octubre de 2019. Disponible en: https://www.caracteristicas.co/software/