



## Introducción

# Ingeniería de Software

María Antonieta Soto Ch.

2019

Fuente: Slides - Sommerville, Ian, Software Engineering. 10th ed., chapter 1, 2015



## Contenido

- ✧ Desarrollo de software profesional
  - Lo que se entiende por ingeniería de software.
  
- ✧ Ética de la ingeniería de software
  - Una breve introducción a los aspectos éticos asociados a la ingeniería de software.

## Ingeniería de software



- ✧ Las economías de TODOS los países dependen de software.
- ✧ Cada vez más sistemas se controlan mediante software.
- ✧ La ingeniería de software usa teorías, métodos y herramientas para el desarrollo de software profesional.
- ✧ Los gastos en software representan una fracción significativa del PIB de todos los países desarrollados.

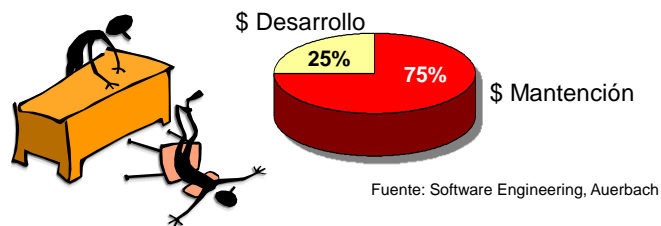
Introducción

3

## Costos del software



- ✧ Los costos del software son a menudo mayores que el costo del hardware.
- ✧ Cuesta más mantener el software que desarrollarlo.
- ✧ La ingeniería de software se preocupa de desarrollar software costo-efectivo.



Introducción

4

## Fallas de los proyectos de software



### ✧ *Complejidad creciente de los sistemas*

- A medida que las técnicas de la ingeniería de software nos ayudan a construir sistemas más grandes y complejos, la demanda cambia. Los sistemas se tienen que construir y distribuir más rápidamente; se requieren sistemas aún más complejos; los sistemas deben tener nuevas capacidades que anteriormente se pensaba eran imposibles.

### ✧ *Fallas al usar, o no usar, métodos de ingeniería de software*

Introducción

5



## Éxitos y fracasos en el desarrollo de software

Introducción

6

## Accidente del vuelo 501 del ARIANE-5



- ✧ Ariane-5 explotó el 4 de Junio de 1996, cuando a 40 segundos después de la iniciación de la secuencia de vuelo se desvió de su ruta, se partió y explotó. En el proyecto global se invirtieron 10 años de construcción y 7 mil millones de euros, lo que supuso un duro varapalo para la Agencia Espacial Europea (ESA)
- ✧ **A los 36,7 segundos el software de guiado inercial produce overflow al convertir un número representado en 64 bits (float) a 16 bits (int).**
- ✧ Caída del sistema de backup, a los 0.05 segundos caída del sistema principal.
- ✧ El sistema de control principal recibe datos de diagnóstico que interpreta como datos de vuelo (ha cambiado la posición del cohete). Fuerte reacción para corregir la trayectoria. Desintegración por la fuerza aerodinámica. Autodestrucción.

Introducción

7

## Un error de programación destruye la sonda Mariner I

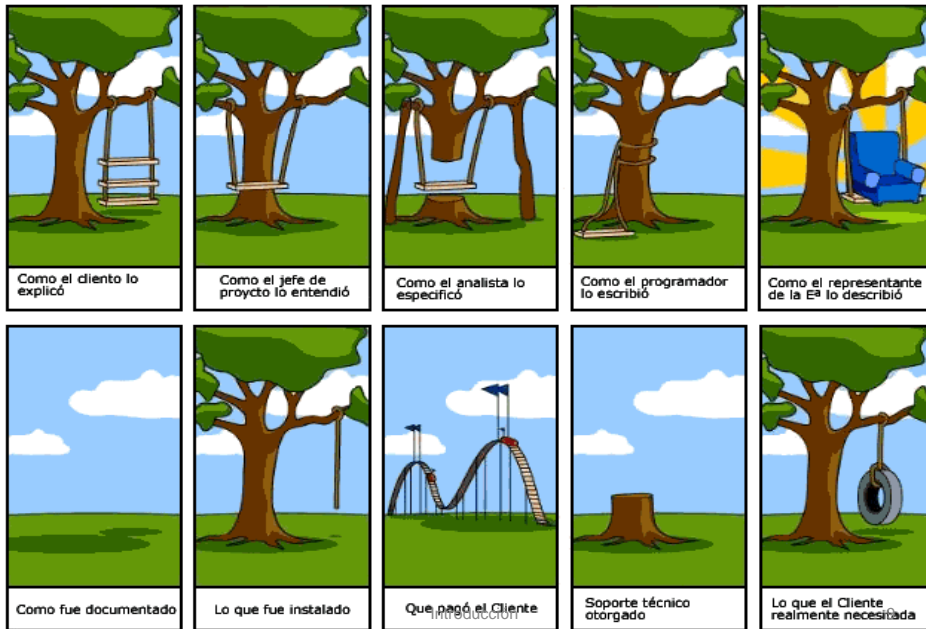


- ✧ Un "bug" en el software de vuelo de la sonda Mariner I provocó que, segundos después del lanzamiento de la nave, en julio de 1972, esta se desviara de su curso preestablecido. Los responsables de la misión se vieron obligados a destruir el cohete cuando se encontraba sobrevolando el Atlántico.
- ✧ La mala transcripción de una fórmula escrita a lápiz en lenguaje FORTRAN a la computadora de la nave, hizo que esta se desviara de su trayectoria y se tuviera que destruir desde el centro de control.
- ✧ La instrucción era **DO 17 I = 1, 10** en el programa original pero se escribió **DO 17 I = 1.10**. El cambio de la coma por un punto hizo que en lugar de una instrucción de programa se convirtiera en la asignación de una variable **DO17I=1.10** ya que FORTRAN no tiene en cuenta los espacios en blanco.
- ✧ El error hizo que la nave diera un giro de 180 grados y se dirigiera a la Tierra segundos después del despegue obligando a los responsables de la misión a destruirlo sobre el Océano Atlántico.

Introducción

8

## Paradojas



## Chaos Report, 2011-2015

✧ Contempla alrededor de 50.000 proyectos

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 database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

## Chaos Report 2011-2015



- ✧ The Standish Group has redefined project success as **onTime, onBudget with a satisfactory result.**
- ✧ This year's report (2015) includes six factors in the overall measure of success:
  - on Time
  - on Budget
  - on Target (% requirements)
  - satisfaction (very high to very low)
  - value (very high to very low)
  - on strategic corporate goal (precise to distant)

Introducción

11

## Chaos Report, 2011-2015



**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%
<b>TOTAL</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>

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

Introducción

12

## Chaos Report, 2011-2015



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.

Introducción

13

## Chaos Report, 2011-2015



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%

Introducción

14

## Chaos Report, 2011-2015 Success factors - definition



- ✧ **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. Emotional maturity skills include the 5 deadly sins of PM (overambition, arrogance, ignorance, abstinence, and fraudulence), managing expectations, consensus building, and collaboration.
- ✧ **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.

Introducción

15

## Chaos Report, 2011-2015 Success factors - definition



- ✧ **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.
- ✧ **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.

Introducción

16



## Chaos Report, 2011-2015 Success factors - definition



- ✧ **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.

Introducción

17

## Chaos Report, 2011-2015



		COMPLEXITY				
		C1	C2	C3	C4	C5
SIZE	S1	100	250	400	550	700
	S2	175	325	475	625	775
	S3	250	400	550	700	850
	S4	325	475	625	775	925
	S5	400	550	700	850	1000

- ✧ The more complex and bigger the higher the risk of failure.
- ✧ An agile approach help overcome complexity because it can by failing earlier and restarting faster.

Introducción

18



## Desarrollo de software profesional

Introducción

19

## Ingeniería de software



### ✧ Ingeniería de software según Bohem 1976:

“Es la aplicación práctica del conocimiento científico en el diseño y construcción de software y documentación asociada, requerida para desarrollar, operar y mantenerlos”

### ✧ Ambiente de Ingeniería de Software (SEE)

Es la colección coordinada de herramientas de software organizadas para apoyar algunos enfoques usados para el desarrollo de software o conforme a algún modelo de proceso de software

Introducción

20

## Ingeniería de software



- ✧ La ingeniería de software es una disciplina de ingeniería que se ocupa de todos los aspectos de la producción de software, desde las primeras etapas de la especificación del sistema hasta el mantenimiento después que se pone en operación... También se agrega el retiro del software.
- ✧ Disciplina de ingeniería
  - Aplicando teorías científicas y métodos apropiados para resolver problemas usando la tecnología, teniendo en mente las restricciones organizacionales y financieras.
- ✧ Todos los aspectos de la producción del software
  - No solo el proceso técnico del desarrollo, sino también la gestión de proyectos y el desarrollo de herramientas, métodos, etc. para apoyar la producción de software.

Introducción

21

## Importancia de la ingeniería de software



- ✧ Cada vez más, los individuos y la sociedad se apoyan en los avanzados sistemas de software. Por ende, se requiere producir económica y rápidamente sistemas confiables.
- ✧ A menudo resulta más barato a largo plazo usar métodos y técnicas de ingeniería de software para los sistemas de software, que solo diseñar programas como si fuera un proyecto de programación personal. Para muchos tipos de sistemas, la mayoría de los costos consisten en cambiar el software después de ponerlo en operación.

Introducción

22

## Actividades del proceso de software



- ✧ Especificación del software, donde los clientes e ingenieros definen el software que se producirá y las restricciones en su operación.
- ✧ Desarrollo del software, donde se diseña y programa el software.
- ✧ Validación del software, donde se verifica el software para asegurarse que sea lo que el cliente requiere.
- ✧ Evolución del software, donde se modifica el software para reflejar los requerimientos cambiantes del cliente y del mercado.

Introducción

23

## Asuntos generales que afectan al software



- ✧ Heterogeneidad
  - Cada vez más, se requieren sistemas que funcionen de manera distribuida a través de redes que incluyen diferentes tipos de dispositivos informáticos y móviles.
- ✧ Cambio empresarial y social
  - Las empresas y la sociedad están cambiando muy rápidamente a medida que las economías emergentes se desarrollan y están disponibles nuevas tecnologías. Ellos tienen que ser capaces de cambiar su software existente y desarrollar rápidamente nuevo software.

Introducción

24

## Asuntos generales que afectan al software



### ✧ Seguridad y confianza

- Dado que el software está interconectado con todos los aspectos de nuestra vida, es esencial que podamos confiar en ese software.

### ✧ Escala

- El software se tiene que desarrollar en una amplia variedad de escalas, desde sistemas muy pequeños embebidos en dispositivos portátiles o en prendas tecnológicas (wearable devices) hasta sistemas a escala Internet, basados en la nube que sirven a la comunidad global.

Introducción

25

## Diversidad de la ingeniería de software



- ✧ Existen muchos diferentes tipos de sistemas de software y no existe un conjunto universal de técnicas de la ingeniería de software que sea adecuado para todos ellos.
- ✧ Los métodos y herramientas de la ingeniería de software a usar dependen del tipo de aplicación que se está desarrollando, los requerimientos del cliente y la experiencia y conocimiento del equipo de desarrollo.

Introducción

26

## Tipos de aplicaciones



- ✧ Aplicaciones independientes (standalone)
- ✧ Aplicaciones interactivas basadas en transacciones
  - Se ejecutan en equipos remotos accedidos por los usuarios desde sus equipos.
  - P.e.: Aplicaciones web tales como las de e-commerce.
- ✧ Sistemas de control embebido
  - Controlan y administran dispositivos de hardware.
  - Hay más sistemas de este tipo que de cualquier otro.
- ✧ Sistemas de procesamiento en lotes

Introducción

27

## Tipos de aplicaciones



- ✧ Sistemas de entretenimiento
- ✧ Sistemas para modelado y simulación
  - Desarrollado por científicos e ingenieros para modelar procesos físicos o situaciones en que intervienen muchos objetos separados que interactúan.
- ✧ Sistemas de adquisición de datos
  - Recopilan datos desde su entorno usando un conjunto de sensores y envían dichos datos a otros sistemas para su procesamiento.
- ✧ Sistemas de sistemas

Introducción

28

## Fundamentos de la ingeniería de software



- ✧ Algunos principios fundamentales que se aplican a todos los tipos de sistemas de software, independientemente de las técnicas de desarrollo usadas:
  - Se deberían desarrollar usando un proceso de desarrollo administrado y comprendido. Por supuesto, se usan diferentes procesos para diferentes tipos de software.
  - La confiabilidad y el desempeño son importantes para todos los tipos de sistema.
  - Es importante comprender y administrar las especificación y los requerimientos (lo que el software debería hacer).
  - Donde sea adecuado, se debería reusar software que haya sido desarrollado en lugar de construir uno nuevo.

Introducción

29

## Ingeniería de software e Internet



- ✧ La web es ahora la plataforma para ejecutar aplicaciones y las organizaciones están desarrollando cada vez más sistemas basados en la web en lugar de sistemas locales.
- ✧ Los servicios web permiten que la funcionalidad de una aplicación se pueda acceder a través de la web.
- ✧ La computación en la nube es una aproximación para proveer servicios de cómputo en el que las aplicaciones se ejecutan remotamente en la 'nube'.
  - Los usuarios no compran software sino que pagan de acuerdo al uso que hagan de él.

Introducción

30

## Ingeniería de software basada en web



- ✧ Los sistemas basados en web son sistemas distribuidos complejos, pero los principios fundamentales de la ingeniería de software, antes mencionados, son tan aplicables a ellos como lo son a cualquier otro tipo de sistema.
- ✧ No obstante lo anterior, existen enfoques más apropiados para este tipo de sistemas:
  - Reúso de software, utilizando componentes reusables.
  - Desarrollo incremental y ágil.
  - Sistemas orientados a servicio.
    - Se puede implementar usando ingeniería de software orientada a servicios, donde los componentes de software se construyen como servicios web independientes.
  - Interfaces de usuario enriquecidas (HTML5, AJAX).

Introducción

31



## Ética en la ingeniería de software

Introducción

32



## Ética en la ingeniería de software



- ✧ La ingeniería de software implica responsabilidades mayores que la simple aplicación de habilidades técnicas.
- ✧ Los ingenieros de software deben comportarse de manera honesta y éticamente responsable para ser respetado como profesionales.
- ✧ El comportamiento ético es más que solo respetar la ley, sino implica seguir un conjunto de principios que son moralmente correctos.

Introducción

33

## Responsabilidad profesional



- ✧ Confidencialidad
  - Los ingenieros deberían respetar la confidencialidad de sus empleadores o clientes sin importar si firmó o no un acuerdo formal de confidencialidad.
- ✧ Competencia
  - Los ingenieros no deben desvirtuar su nivel de competencia. No deberían aceptar de manera intencional trabajo que está fuera de su competencia.

Introducción

34

## Responsabilidad profesional



### ✧ Derechos de propiedad intelectual

- Los ingenieros deben conocer las leyes locales que rigen el uso de propiedad intelectual, como las patentes y el copyright, etc. Debe tener el cuidado de garantizar que se protege la propiedad intelectual de empleadores y clientes.

### ✧ Mal uso de computadores

- Los ingenieros de software no deben emplear sus habilidades técnicas para usar incorrectamente los computadores de otros individuos. El mal uso de computadores varía desde lo relativamente trivial (digamos, usar los juegos del equipo de un empleado) hasta lo extremadamente serio (diseminación de virus u otro malware).

Introducción

35



Introducción

36