Ingeniería de Software I

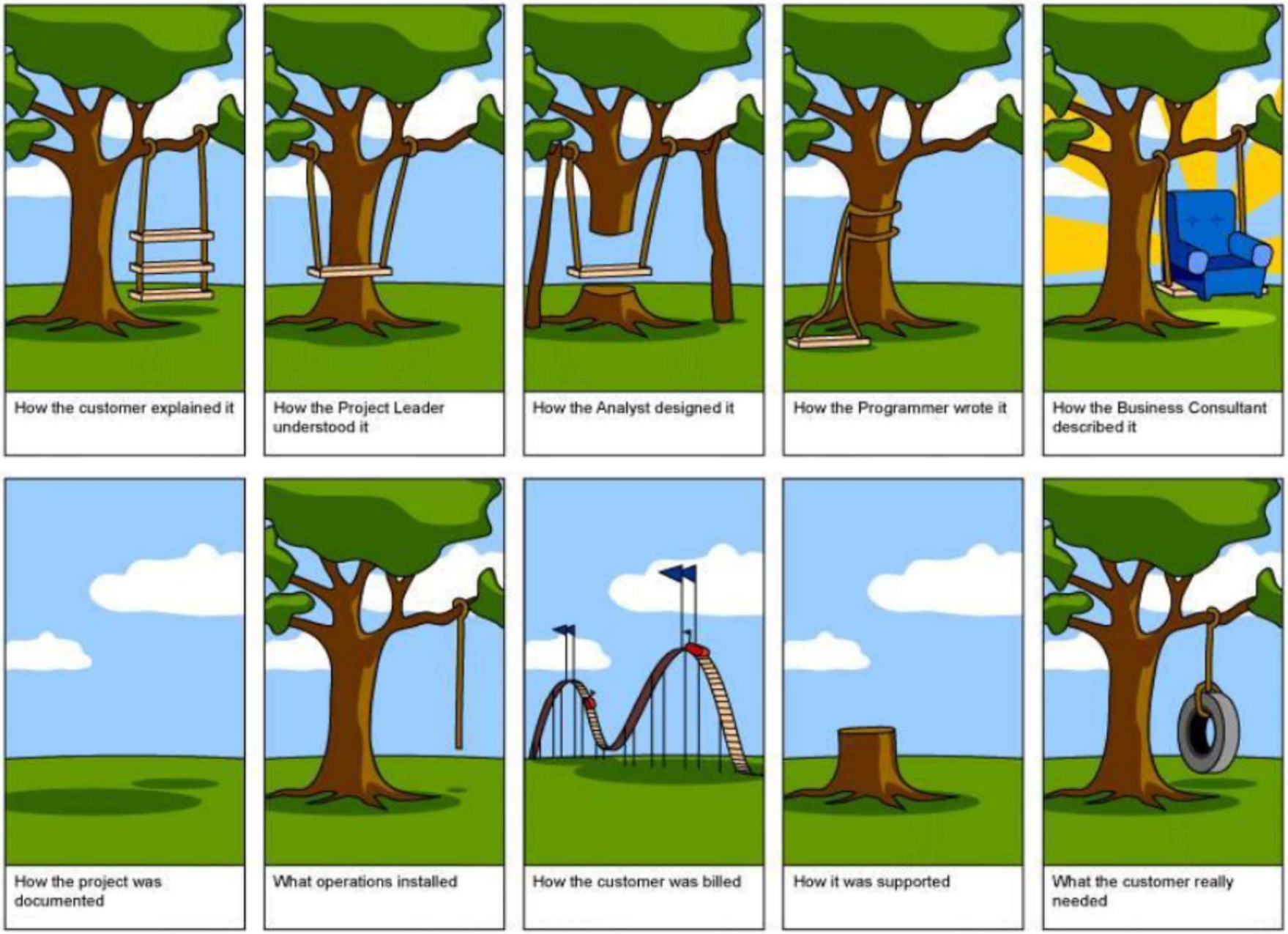
Ingeniería de Requisitos: Fundamentos

Edgar Sarmiento Calisaya

Escuela Profesional de Ciencia de la Computación
Universidad Nacional de San Agustín de Arequipa,
Arequipa, Perú

Contenido

1. Ingeniería de Software
2. Ingeniería de Software y Requisitos
3. Por que Ingeniería de Requisitos?
4. Requisito de Software
5. Qué es Ingeniería de Requisitos?
6. Carateristicas del Ingeniero de Requisitos
7. Conclusión


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Ingeniería de Software

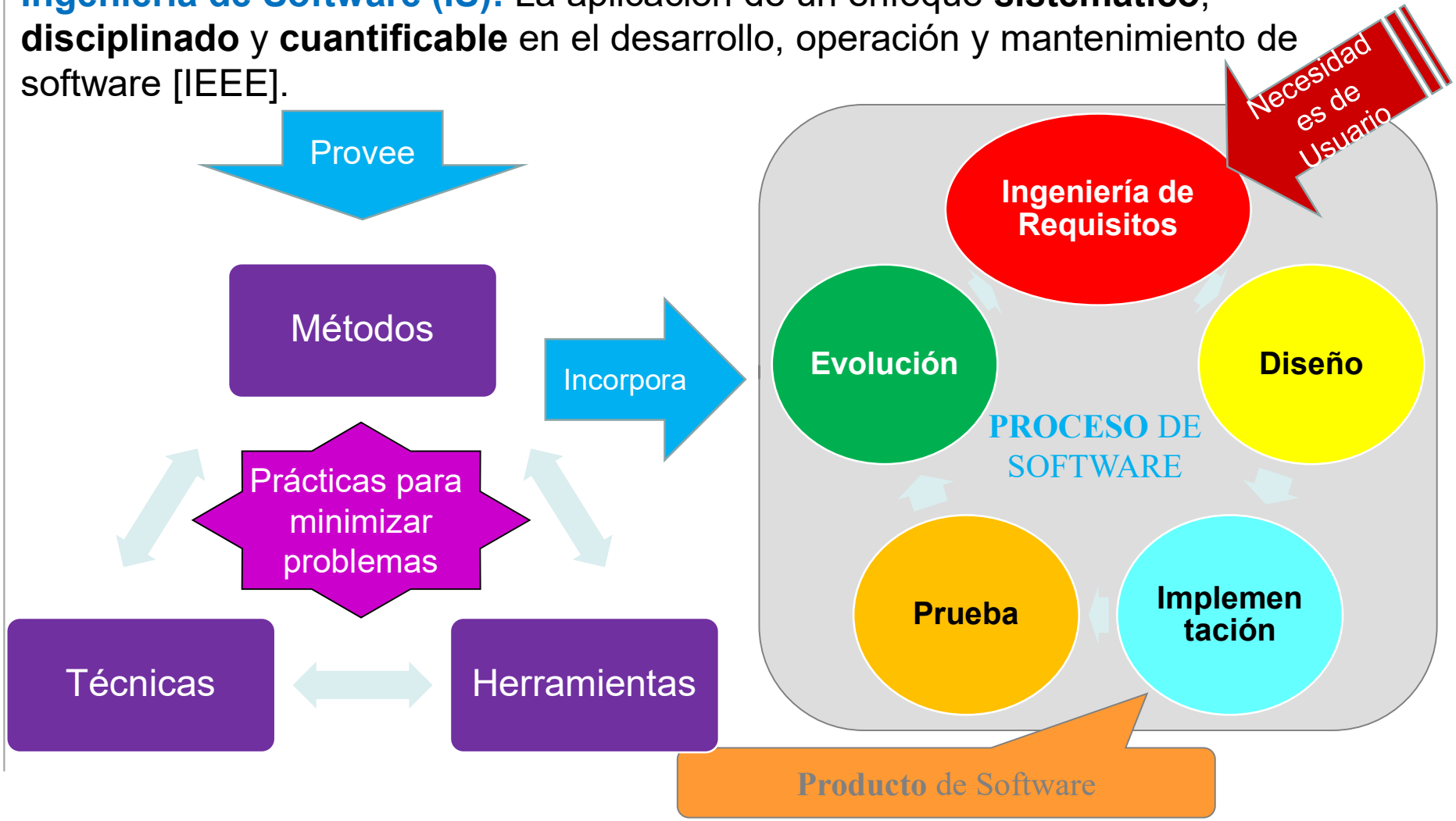
- **Desarrollar software es una actividad compleja** por naturaleza.
- Una de las razones de esta afirmación es que **no existe una única solución para cada escenario de desarrollo**.
- Además, nos ocupamos **todo el tiempo con personas**, lo que hace que el éxito del proyecto esté bastante relacionado con la competencia del equipo y la forma en que trabajan, y para dificultar aún más, **muchas veces no hacemos uso de un proceso bien definido** para apoyar las actividades del proyecto.
- Se entiende por **proceso**, en este contexto, como un **conjunto de actividades bien definidas con los respectivos responsables de ejecución, herramientas de apoyo y artefactos producidos**. Es decir, se define como el equipo deberá trabajar para alcanzar el objetivo: desarrollar software con calidad dentro de plazos, costos y requisitos definidos.

Ingeniería de Software

- La **buena noticia** es que muchas empresas se están moviendo en el sentido de **definir detalladamente sus procesos** para apoyar sus actividades de **desarrollo**.
  - Servicios de mejor calidad;
  - Internacionalización

Ingeniería de Software

Ingeniería de Software (IS): La aplicación de un enfoque **sistemático**, **disciplinado** y **cuantificable** en el desarrollo, operación y mantenimiento de software [IEEE].



Ingeniería de Software

- Ingeniería de software: La aplicación de un enfoque sistemático, disciplinado y cuantificable en el desarrollo, operación y mantenimiento de software [IEEE].
  - **Sistemática** porque parte del principio de que existe un proceso de desarrollo definiendo las actividades que deberán ser ejecutadas.
  - **Disciplinado** porque parte del principio de que los procesos definidos serán seguidos.
  - **Cuantificable** porque se debe definir un conjunto de medidas a ser extraídas del proceso durante el desarrollo de forma que las tomas de decisión relacionadas al desarrollo del software (por ejemplo, mejora de proceso) se basen en datos reales, y no en “creencias" .
- Algunos de sus principales **objetivos** son:
  - **Calidad** de software;
  - **Productividad** en el desarrollo, operación y mantenimiento de software;
  - Permitir que los profesionales tengan **control** sobre el desarrollo de software dentro de **costos**, **plazos** y **niveles de calidad** deseados.

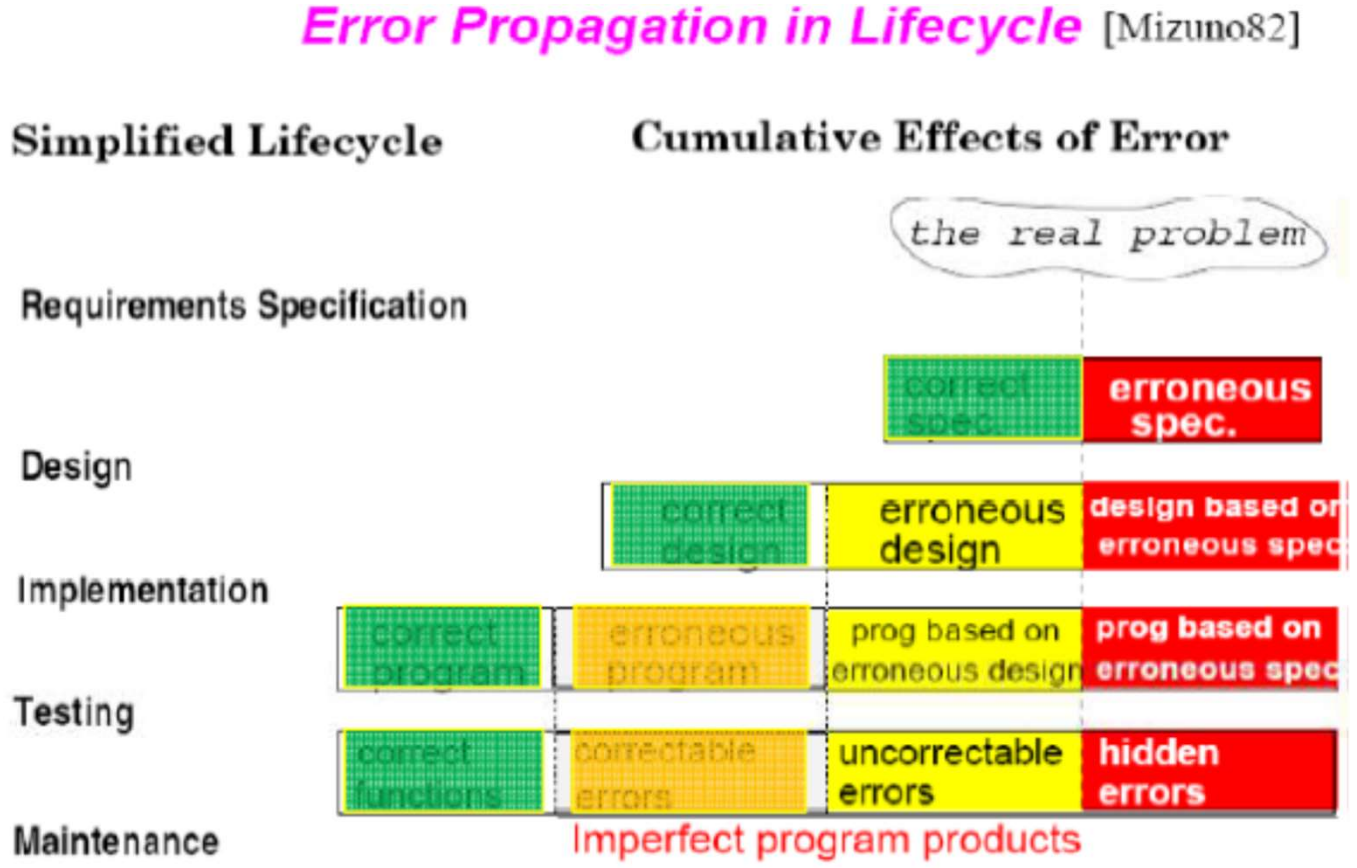
Ingeniería de Software

- Sin embargo, el **escenario de desarrollo de software actual y el escenario ideal** de la ingeniería de software todavía están **distantes**.
- Varios factores contribuyen a ello, podemos citar dos:
  - El no uso de los fundamentos de la ingeniería de software para apoyar las actividades del desarrollo;
  - El mal uso de los fundamentos de la ingeniería de software para apoyar las actividades del desarrollo.

Ingeniería de Software

- Esto tiene varias consecuencias.
  - El creciente **costo con el mantenimiento de los sistemas**.
  - Mantenimiento es cualquier re-trabajo (a nivel de requisitos, diseño, codificación, prueba) causado por una definición del dominio del problema mal elaborada en las fases iniciales del desarrollo (REQUISITOS).
- Defectos en requisitos se propagan y aumentan en las fases siguientes!

Ingeniería de Software y Requisitos



How big is the erroneous spec.?
How costly is it?

Chung, L. Why is RE?

Ingeniería de Software y Requisitos

How big is the "erroneous specification"?

- **Bell Labs and IBM studies**

  80% of all defects are inserted in the requirements phase.
  Improving the requirements definition process reduces
  the amount of testing and rework required.

  And the above figures do not include the end user losses
  who have to live with poor software on a daily basis *[Testing Techniques Newsletter]*

- **U.S. Air Force projects**

  36% of all defects were due to faulty requirements translation.
  Only 9% of these errors were resolved (in the requirements phase) *[Sheldon92]*

- **Voyager and Galileo spacecraft**

  Of the 197 significant software faults found during integration & system testing
  only 3 of those errors were programming errors;
  the vast majority of the faults were requirements problems. *[Lutz93]*

- **Application Specific Integrated Circuits [ASICs)**

  >1/2 are faulty on first fabrication. A majority of these faults are related to reqs. errors.

- **[UK Health and Safety] Executive**

  Specification 44.1%
  Design and Implementation 14.7%
  Installation and Commissioning 5.9%
  Operation and Maintenance 14.7%
  Changes after commissioning 20.6%

  [Her Majesty's Stationary Office 1995 ISBN 0 7176 0847 6]

Chung, L. Why is RE?

Ingeniería de Software y Requisitos

- Project Management Institute – 2014:
  - 47% de los proyectos de software que fallaron tiene como origen Requisitos de Software imprecisos como su **causa principal**.

Firesmith Donald, 2003

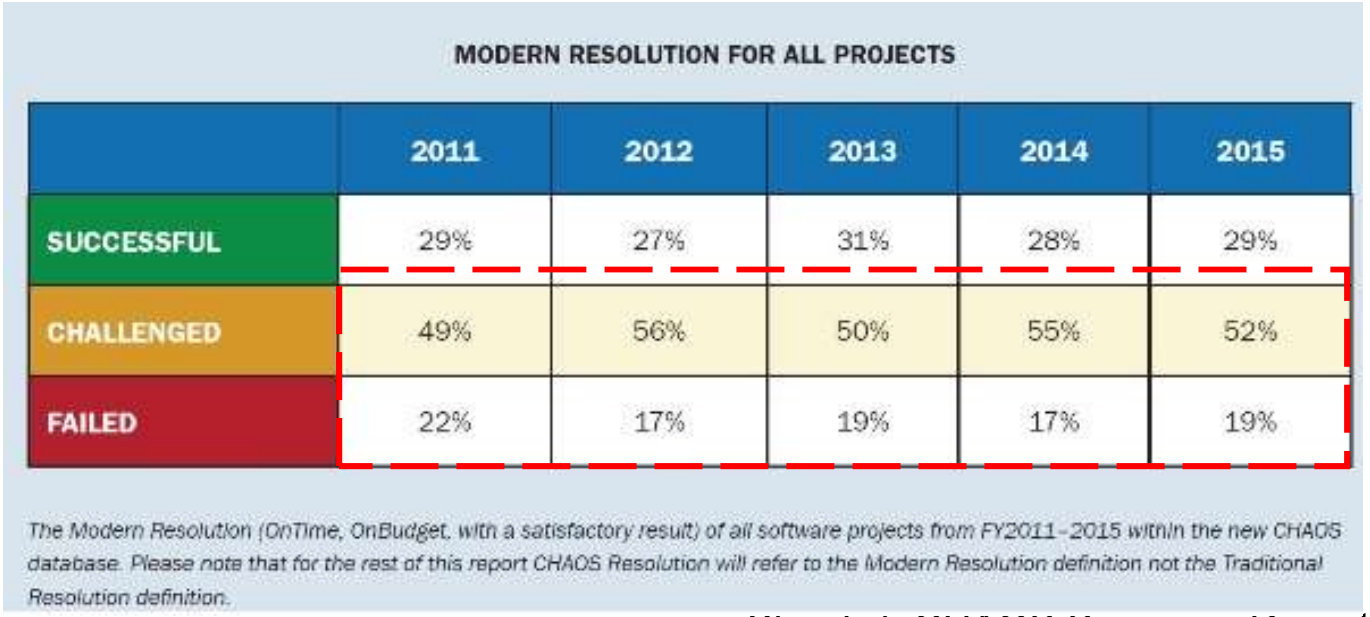
Bad Requirements Cause Failures

Requirements problems are the single number one cause of project failure:

- **Significantly over budget**
- **Significantly past schedule**
- **Significantly reduced scope**
- **Poor quality applications**
- **Not significantly used once delivered**
- **Cancelled**

Ingeniería de Software y Requisitos

- Standish CHAOS Summary Report – 2015:
  - 71% de los proyectos no alcanzan sus objetivos, y tienen como origen **Requisitos** de software imprecisos entre sus causas.



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.

Standish CHAOS Summary Report – 2015

Ingeniería de Software y Requisitos

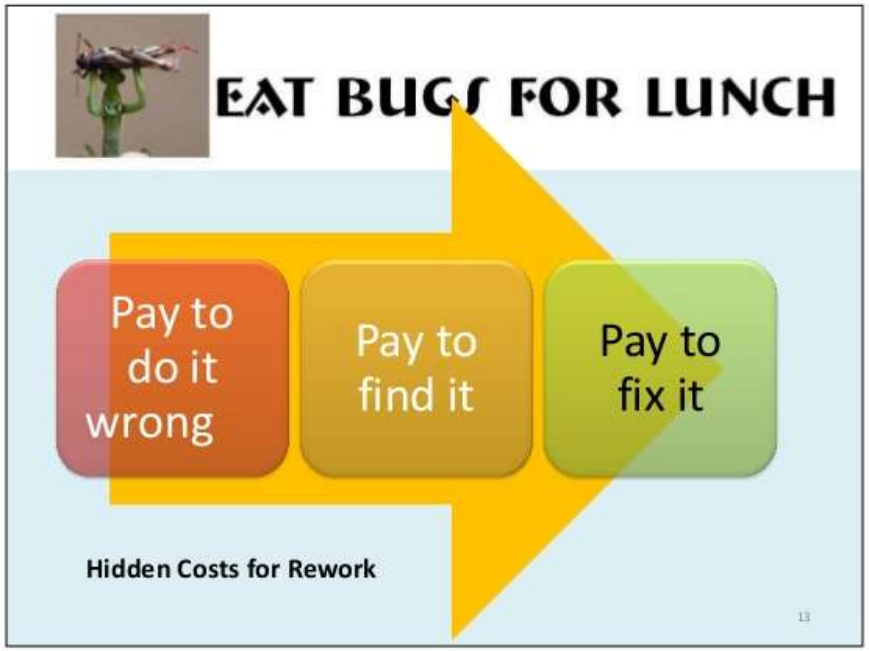
Impact of Requirements Defects

Organization/Project	Overruns Attributed to Requirements Problems
NASA over two decades (Werner Gruhl)	70% of overrun amount
U.S. Census Bureau project 2009	80% cost overrun locked in solely due to poor requirements
Marine One Helicopter Program	83% cost overrun attributed by Lockheed to requirements problems
Schwaber, 2006; Weinberg, 1997; Nelson et al, 1999	"Requirements errors are the single greatest source of defects and quality problems"
Hofmann and Lehner, 2001	"Deficient requirements are the single biggest cause of software project failure."
Standish Group, The Chaos Report on 8300 IT projects	60.9% of an average 89% cost overrun

Halligan Robert, 2014 - Project Performance International (PPI)

Ingeniería de Software y Requisitos

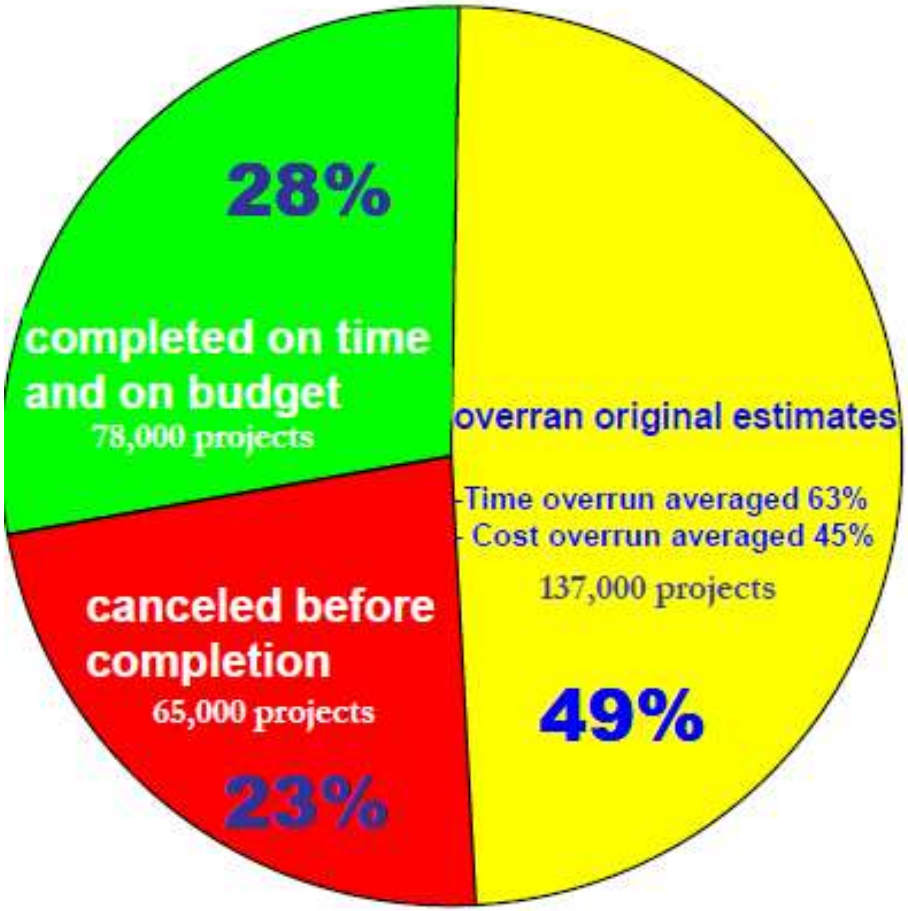
- ***HOY*** dedicamos nuestros esfuerzos a corregir lo que hicimos mal ***AYER*** – **REWORK** [Arnd Von Staa, PUC-Rio]



Rebecca Staton-Reinstein, Irv Browntein, 2012

What Factors Contribute to Project Success?

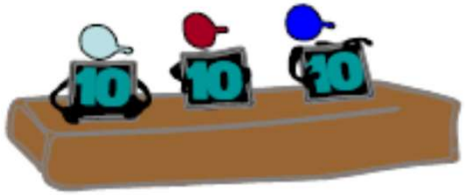
The Standish Group Report, '01 – The "Chaos" Report (www.standishgroup.com)
yearly since 1994, survey of close to 300,000 projects



The CHAOS Ten
Project Success Factors
1. Executive Management Support
2. **User Involvement**
3. Experienced Project Manager
4. **Clear Business Objectives**
5. **Minimized Scope**
6. Standard Software Infrastructure
7. **Firm Basic Requirements**
8. Formal Methodology
9. Reliable Estimates
10. Other

Chung, L. Why is RE?

What Factors Contribute to Project Failure?



The CHAOS Ten

Project Challenged Factors

1. Lack of User Input
2. Incomplete Requirements & Specifications
3. Changing Requirements & Specifications
4. Lack of Executive Support
5. Technology Incompetence
6. Lack of Resources
7. Unrealistic Expectations
8. Unclear Objectives
9. Unrealistic Time Frames
10. New Technology

The CHAOS Ten

Project Impaired Factors

1. Incomplete Requirements
2. Lack of User Involvement
3. Lack of Resources
4. Unrealistic Expectations
5. Lack of Executive Support
6. Changing Requirements & Spec
7. Lack of Planning
8. Didn't Need It Any Longer
9. Lack of IT Management
10. Technology Illiteracy

Chung, L. Why is RE?

Ingeniería de Software y Requisitos

"The definition of insanity is doing the same thing over and over again and expecting a different result."
[Albert Einstein]

Chung, L. Why is RE?

How costly are requirements errors?

- *[Lindstrom93]*

  Get the requirements wrong, you'll destroy the project.

- *[Boehm87]*

  COST (correcting design/implementation errors)
  = 100 X COST (correcting requirements errors)

- *[Humphrey, Managing the Software Process, Ch1, p11-12]*

  a useful rule of thumb: It takes about 1 to 4 working hours to find and fix a bug through inspections and about 15 to 20 working hours to find and fix a bug in function or system test.

- *[Curtis88]*

  Three most frequent problems plaguing large software systems:

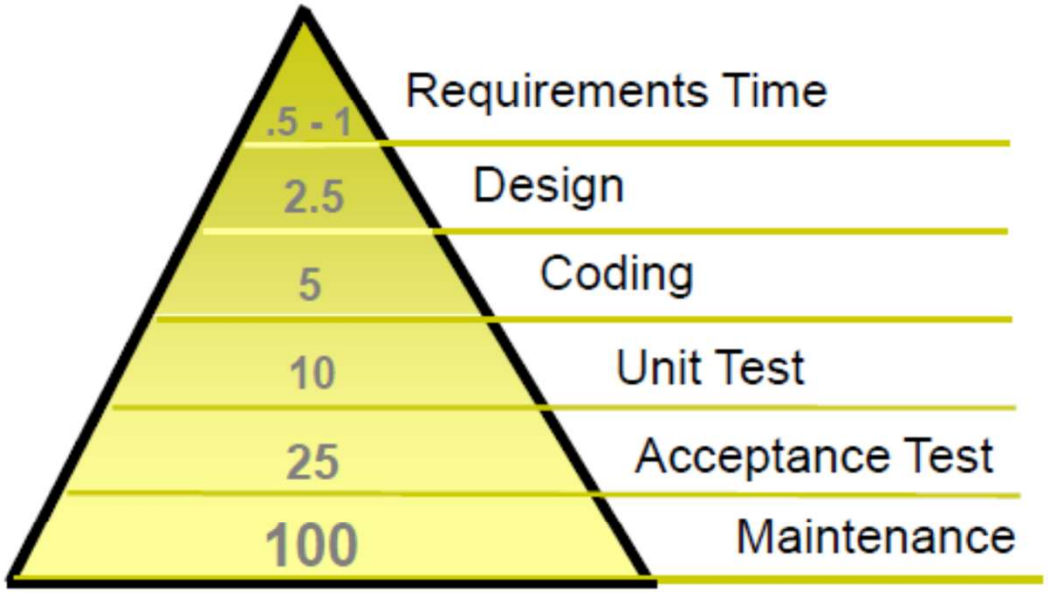
  communication and coordination
  thin spread of domain application knowledge
  changing and conflicting requirements

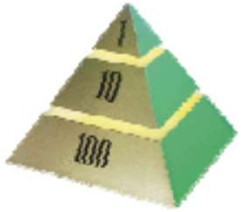
  Defining the problem is The Problem

Chung, L. Why is RE?

The High Cost of Requirement Errors

The 1-10-100 Rule





"All together, the results show as much as a 200:1 cost ratio between finding errors in the requirements and maintenance stages of the software lifecycle."

Relative cost to repair errors:
When introduced vs. when repaired.
[Davis 1993]

Average cost ratio 14:1
[Grady1989] [Boehm 1988]

Why?

Chung, L. Why is RE?

Por qué Ingeniería de Requisitos?

- Sabemos que **un trabajo más delicado en el área de requisitos es fundamental para el éxito de proyectos de software.**
  - **Conocer y**
  - **Documentar los requisitos;**

Requisito de Software

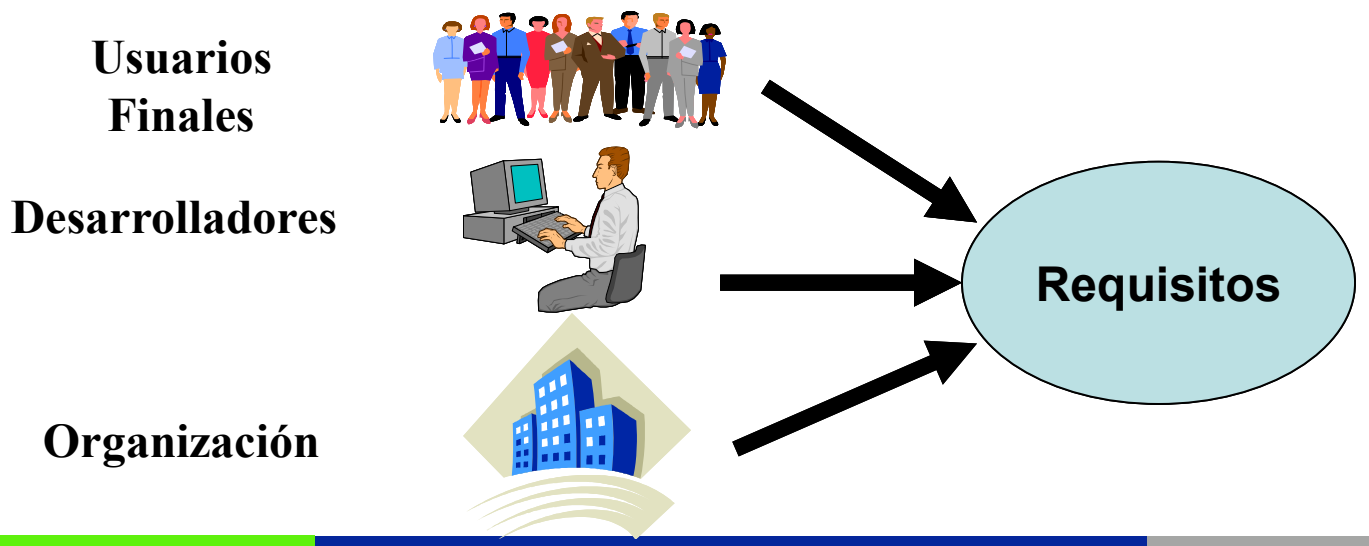
- **Requisito**:
  1. Una **condición** o **capacidad** necesaria para que un usuário pueda resolver un problema o alcanzar un objetivo;
  2. Una **condición** o **capacidad** que debe ser alcanzada o estar presente en un sistema o componente de sistema para satisfacer un contrato, norma, especificación u otro documento formalmente impuesto.
  3. Una representación documentada de una **condición** o **capacidad** como en (1) y (2).

IEEE standard 610.12-1990

Requisito de Software

- **Stakeholder**
  - Un stakeholder de un sistema es una persona o una organización que tiene una **influencia** (directa o indirecta) sobre los **requisitos** del sistema

IREB



Requisito de Software

- **Ejemplos de Requisitos**:
  - "El usuario debe realizar búsquedas en todo el acervo de materiales bibliográficos."
  - "El sistema debe proporcionar pantallas apropiadas para que el usuario pueda leer documentos disponibles en el repositorio de documentos ".
  - "El sistema debe permitir el registro de los proveedores de la tienda “
  - "El sistema debe utilizar los datos obtenidos a partir de los sensores e interpretarlos para realizar la navegación "

Qué es Ingeniería de Requisitos?

- La **Ingeniería de Requisitos** establece el procedimiento de definición de Requisitos como un proceso en el que lo que se debe hacer es elicitado, especificado, analizado y gestionado.
- Este proceso debe tratar con diferentes puntos de vista, y utilizar una combinación de métodos, herramientas y el personal.
- El producto de este proceso es un **modelo**, del que un **documento de requisitos** se produce.
- Este proceso se realiza en un contexto previamente definido a la que llamamos **Universo de Información**.

(Júlio Leite, 1994)

Qué es Ingeniería de Requisitos?

- **Universo de Información**.
  - Es el conjunto general en el que el software será desarrollado.
  - Incluye todas las fuentes de información y todas las personas relacionadas con el software, a las que se denominan agentes de ese universo.
  - El **Udel** es la realidad circunstanciada por el conjunto de objetivos definidos por quién solicitó el software.

    (Júlio Leite, 1994)

Qué es Ingeniería de Requisitos?

- Ingeniería de Requisitos : **Es un enfoque sistemático y disciplinado para la especificación y gestión de requisitos**, con los siguientes objetivos:
  - Conocer los requisitos relevantes, establecer un consenso entre los *stakeholders* sobre estos requisitos, documentarlos según los estándares, y gestionar los requisitos de forma sistematica.
  - Entender y documentar las expectativas y necesidades de los *stakeholders*, especificar y gestionar los requisitos para minimizar el riesgo de entregar un sistema que no satisfaga las expectativas y necesidade de los *stakeholders*.

IREB – International Requirements Engineering Board

Qué es Ingeniería de Requisitos?

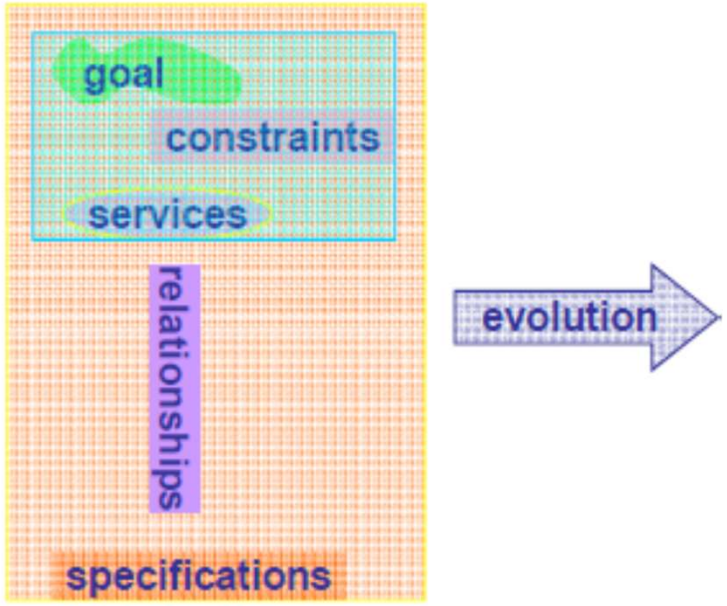
"... Requirements Engineering is the branch of Systems engineering concerned with

real-world goals for,
services provided by, and
constraints on
software systems

Requirements engineering is also concerned with

the relationships of these factors

to precise specifications of system behavior
and
to their evolution over time and across system families..."
[Zave94]



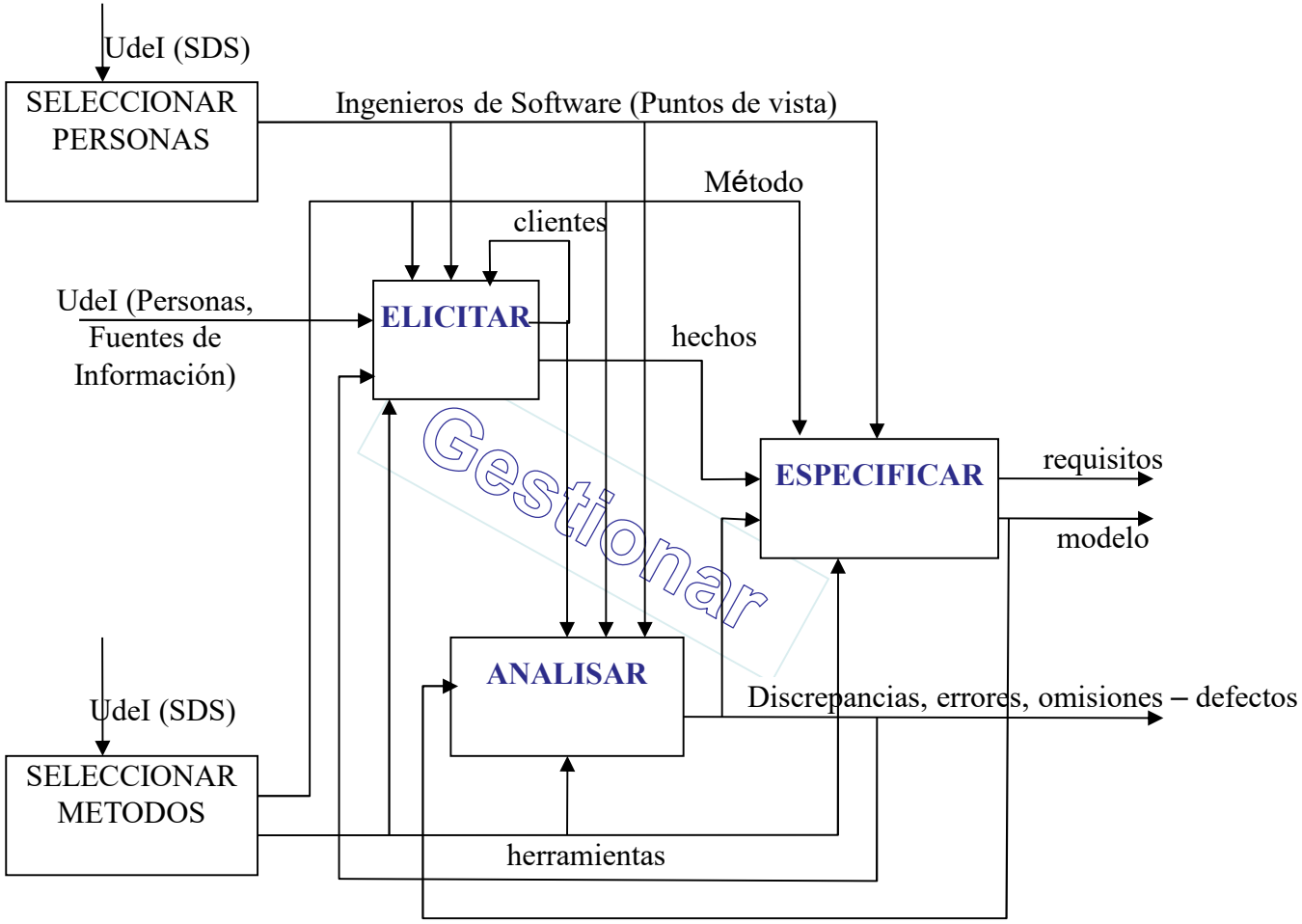
Chung, L. Why is RE?

Qué es Ingeniería de Requisitos?

- Actividades de Ingenriería de Requisitos
  - Los requisitos y las formas de obtenerlos y documentarlos varían drásticamente de un proyecto a otro
  - Sin embargo, existe una serie de actividades genéricas comunes a todos los procesos:
    - Elicitación (Extracción) de requisitos;
    - Especificación de requisitos;
    - Análisis de requisitos;
    - Gestión de requisitos.

Qué es Ingeniería de Requisitos?

SADT –
Actividades



Julio Leite, 1994

Qué es Ingeniería de Requisitos?

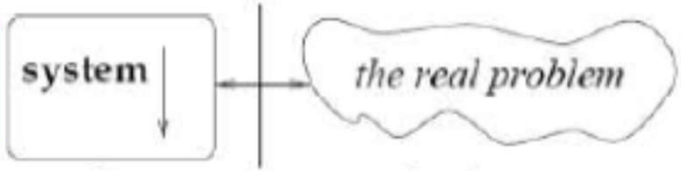
Role of requirements

- **agreement regarding the requirements between system developers, customers, and end-users.**
  - => *legal contract (flexible, inflexible)*
  - => *multi-party*
    - => *communication and coordination*
    - => *conflicting views*
    - => *changing views*

should be written in the user's language!

- **the basis for software design**
  - => *defect-free as much as possible*
  - => *technically feasible*
- **support for verification and validation**

complete & sound I/O
# of I/O items,
and relationships between them
and constraints on them



- **support for system evolution**
  - => *system evolution = change (old system, new system)*
    change (old requirements, new requiremens)

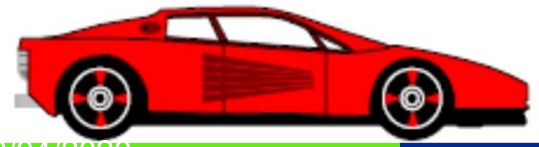
Chung, L. Why is RE?

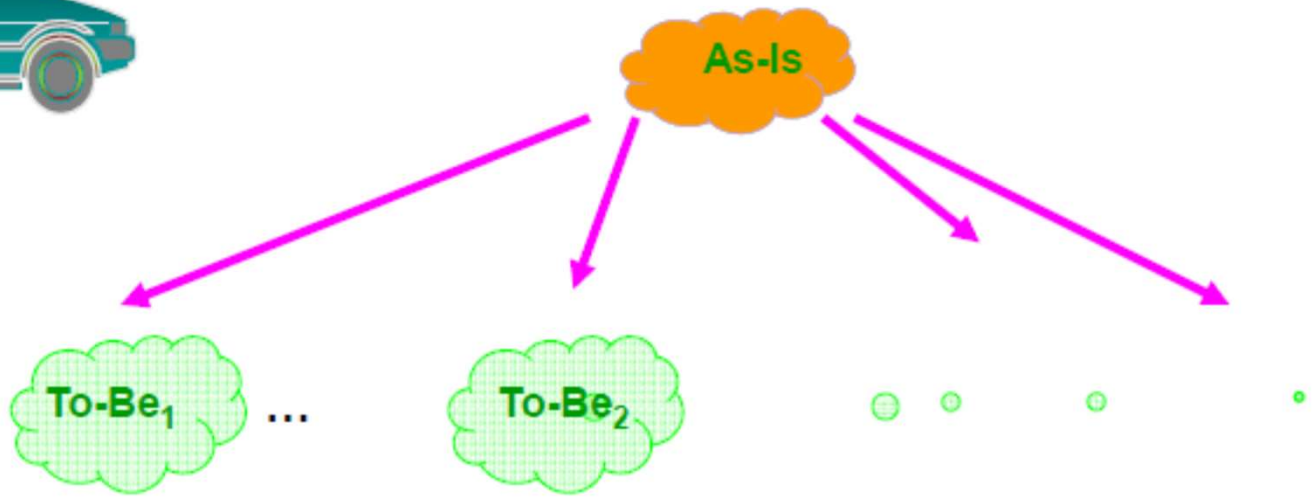
Qué es Ingeniería de Requisitos?

- Requirements Engineering is about determining
  - problems with the current status (As-Is)
  - objectives to achieve
  - changes to bring about

  for a better future (To-Be)

We want to make a change in the environment
We will build some system to do it
This system must interact with the environment



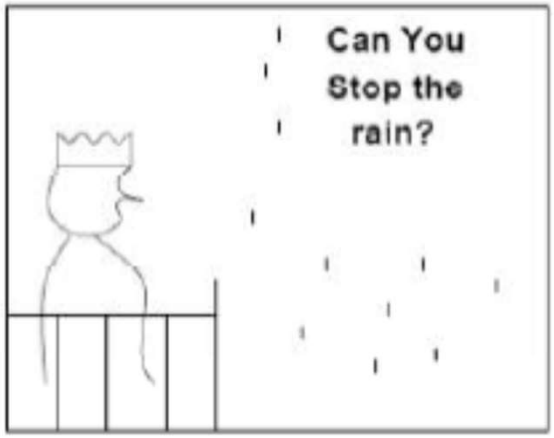


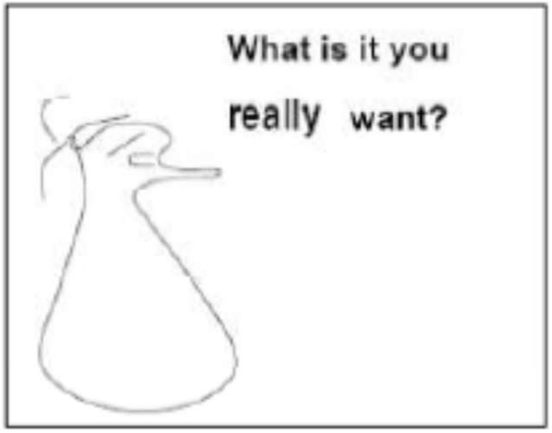
Chung, L. Why is RE?

Carateristicas del Ingeniero de Requisitos (Pohl, K. and Rupp, C. 2015)

- ***Analytic thinking**:* The requirements engineer must be able to become familiar with domains that are unknown to her and must understand and analyze complicated problems and relationships. Since stakeholders often discuss problematic requirements by means of concrete examples and (suboptimal) solutions, the requirements engineer must be able to abstract from the concrete statements of the stakeholder.
- ***Empathy**:* The requirements engineer has the challenging task of identifying the actual needs of a stakeholder. A core requirement to be able to achieve this is to have good intuition and empathy for people. In addition, she must identify problems that might arise in a group of stakeholders and act accordingly.
- ***Communication skills**:* To elicit the requirements from stakeholders and to interpret them correctly and communicate them in a suitable manner, a requirements engineer must have good communication skills. She must be able to listen, ask the right questions at the right time, notice when a statement does not contain the desired information, and make further inquiries when necessary.
- ***Conflict resolution skills**:* Different opinions of different stakeholders can be the cause of conflicts during requirements engineering. The requirements engineer must identify conflicts, mediate between the parties involved, and apply techniques suitable to resolving the conflict.
- ***Moderation skills**:* The requirements engineer must be able to mediate between different opinions and lead discussions. This holds true for individual conversations as well as group conversations and workshops.
- ***Self-confidence**:* Since the requirements engineer is frequently at the center of attention, she occasionally is exposed to criticism as well. As a result, she needs a high level of self-confidence and the ability to defend herself should strong objections to her opinions arise. She should never take criticism personally.
- ***Persuasiveness**:* Among other things, the requirements engineer is, in a matter of speaking, a kind of attorney for the requirements of the stakeholders. She must be able to represent the requirements in team meetings and presentations. In addition, she must consolidate differing opinions, facilitate a decision in case of a disagreement, and create consensus among the stakeholders.

What is RE *Really* about?





What does the customer *really* want?

Lawrence Chung

Conclusión

- La **Ingeniería de Requisitos** tiene como tarea fundamental definir los requisitos de un software [Julio Leite, 1994].

Conclusión

- Los principales beneficios que se obtienen de la Ingeniería de Requisitos son [Ecured]:
  - Permite **gestionar las necesidades del proyecto en forma estructurada**: Cada actividad de la Ingeniería de Requisitos consiste de una serie de pasos organizados y bien definidos.
  - Mejora la **capacidad de predecir cronogramas** de proyectos, así como sus resultados: La Ingeniería de Requisitos proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
  - **Disminuye los costos y retrasos** del proyecto: Muchos estudios han demostrado que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la Especificación de Requisitos.
  - **Mejora la calidad** del software: La calidad en el software tiene que ver con cumplir un conjunto de requisitos (Funcionalidad, Facilidad de Uso, Confiabilidad Desempeño, etc.).
  - **Mejora la comunicación entre equipos**: La especificación de requisitos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.
  - **Evita rechazos de usuarios finales**: La Ingeniería de Requisitos obliga al cliente a considerar sus requisitos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto.

Referencias

- Basado en:

  - Leite, J.C.S.P. 2007. Livro Vivo: Engenharia de Requisitos, PUC-Rio. http://livrodeengenhariaderequisitos.blogspot.com/
  - Chung, L. 2015. Requirements Engineering: Introduction.U. Dallas. http://www.utdallas.edu/~chung/SYSM6309/Introduction.pdf
  - Devmedia. 2008. Artigo Engenharia de Software - Introdução à Engenharia de Requisitos. https://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-engenharia-de-requisitos/8034
  - Rosana T. Vaccare Braga. 2017. Requisitos de Software. USP. https://edisciplinas.usp.br/pluginfile.php/3142953/mod_resource/content/2/Aula09-Requisitos.pdf
  - Elisa Yumi Nakagawa. ENGENHARIA DE REQUISITOS. USP. https://edisciplinas.usp.br/pluginfile.php/58062/mod_resource/content/1/Aula08_Engenharia_Requisitos.pdf
  - Ecured. Ingeniería de Requisitos. https://www.ecured.cu/Ingenier%C3%ADa_de_requisitos
  - Pohl, K. and Rupp, C. 2015. Requirements Engineering Fundamentals. IREB