# SYSTEM ENGINEERING

System engineering is an interdisciplinary approach and means to enable the realization of successful systems.

It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal.

SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user's needs.

# MODEL BASED SYSTEM ENGINEERING

The formalized application of modeling to support system requirements, design, analysis, verification, and validation activities throughout development and later life cycle stages.

MBSE enhances the conventional document-based approach to obtain:

- Improved communications among stakeholders
- Increased ability to manage system complexity
- Improved product quality
- Reduced cycle time
- Reduced risk
- Enhanced knowledge capture and reuse

# WHAT IS A SYSTEM?

A purposeful collection of inter-related components working together to achieve some common objective

A system may include software, mechanical, electrical and electronic hardware and be operated by people.

System components are dependent on other system components.

The properties and behavior of system components are inextricably inter-mingled.

# SYSTEM CATEGORIES

## Technical computer-based systems:

System that includes hardware and software but where the operators and operational processes are not normally considered to be part of the system.

## Socio-technical system:

System that includes technical systems but also operational processes and people who use and interact with the technical system. Socio-technical systems are governed by organizational policies and rules.

## Socio-technical software-intensive system:

Socio-technical system in which software represents the largest segment in terms of development cost and time, development risk or functionality.

# SOFTWARE SYSTEMS ENGINEERING

Discipline for software production founded on well-known engineering principles (design and validation)

Essential to consider software as an industrial product

When missing we observe:

- Software products not providing the expected quality
- Reduced competitiveness:
    - Late delivery
    - Budget overrun

# A YOUNG DISCIPLINE…

Electrical and electronic engineers, interested in building computers, regarded programming ad something to be done by others – either scientists who wanted the numerical results or mathematicians interested in numerical methods

Engineers viewed programming as a trivial task, akin to using a calculator

Many refer to programming as a “skill” and deny that engineering principles must be applied when building software

# THE UNCONSUMMATED MARRIAGE

Unconsummated marriage between…

- Computer science (programming theory) and
- Engineering principles (design and validation)

Software engineering should wed a subset of computer science with the concepts and discipline taught to other engineers:

- Engineers must accept that they don't know enough computer science
- Computer scientists must recognize that being an engineer is different from being a scientist, and that software engineers require an education very different from their own.

Successful marriage example: chemical engineering

- A marriage of chemistry with classical engineering areas(such as thermodynamics, mechanics, and fluid dynamics)
- Nowadays chemical engineering is not regarded as a branch of chemistry

SwEng, term coined about 50 years ago:

- NATO conference at Garmisch, Germany (1968)
- To testify the need of considering software production.

# RESULTS OF THE NATO CONFERENCE

Programming is neither science nor mathematics

Programmers are not adding to our body of knowledge, they build products

Using science and mathematics to build products for others is what engineers do

Software is a major source of problems for those who own and use it. The problems are exaclty those to be expected when products ar built by people who are educated for other professions and believe that building things is not their "real job".

# TYPICAL ASPECTS OF SW PRODUCT

**Accidental** difficulties (can be solved by technology advancements)

- Attitude
- Maintenance
- Specification and design
- Teaming

# SW LIFECYCLE = 3 STAGES, 6 PHASES

SW production = development + maintenance

## Development (stage 1) = 6 phases

1. Requirements definition
2. Requirements specification (or analysis)
3. Planning
4. Design (architectural and detailed)
5. Coding
6. Integration

## Maintenance (stage 2):

- Cover 60% of lifecycle costs

## Phasing-out/Retirement (stage 3)