Systems Council Distinguished Lecture

Systems Engineering

Fundamentals of Systems Engineering

for Complex Systems

Bob Rassa

Systems Council Distinguished Lecturer

bobrassa1@gmail.com
202-320-9862

About your instructor

- Director, Engineering Programs, Raytheon Technologies RTX, Intelligence & Space Systems (Ret.)
- Life Fellow, IEEE
  - Treasurer, IEEE Systems Council
  - Founder & Past President, IEEE Systems Council
  - Vice-President, Publications, IEEE Transportation Electrification Council
  - Past President, Aerospace & Electronic Systems Society
  - Past President, IEEE Instrumentation & Measurement Society
  - Treasurer, IEEE Baltimore Section
- Adjunct Professor, Systems Engineering, Rowan University, NJ
- 45 years experience in aerospace & defense
  - Westinghouse Baltimore (now Northrop Grumman)
  - Mantech International
  - Hughes Aircraft (now Raytheon Technologies RTX)
  - Radar, automatic test systems, logistics, program management, systems engineering

About IEEE Systems Council

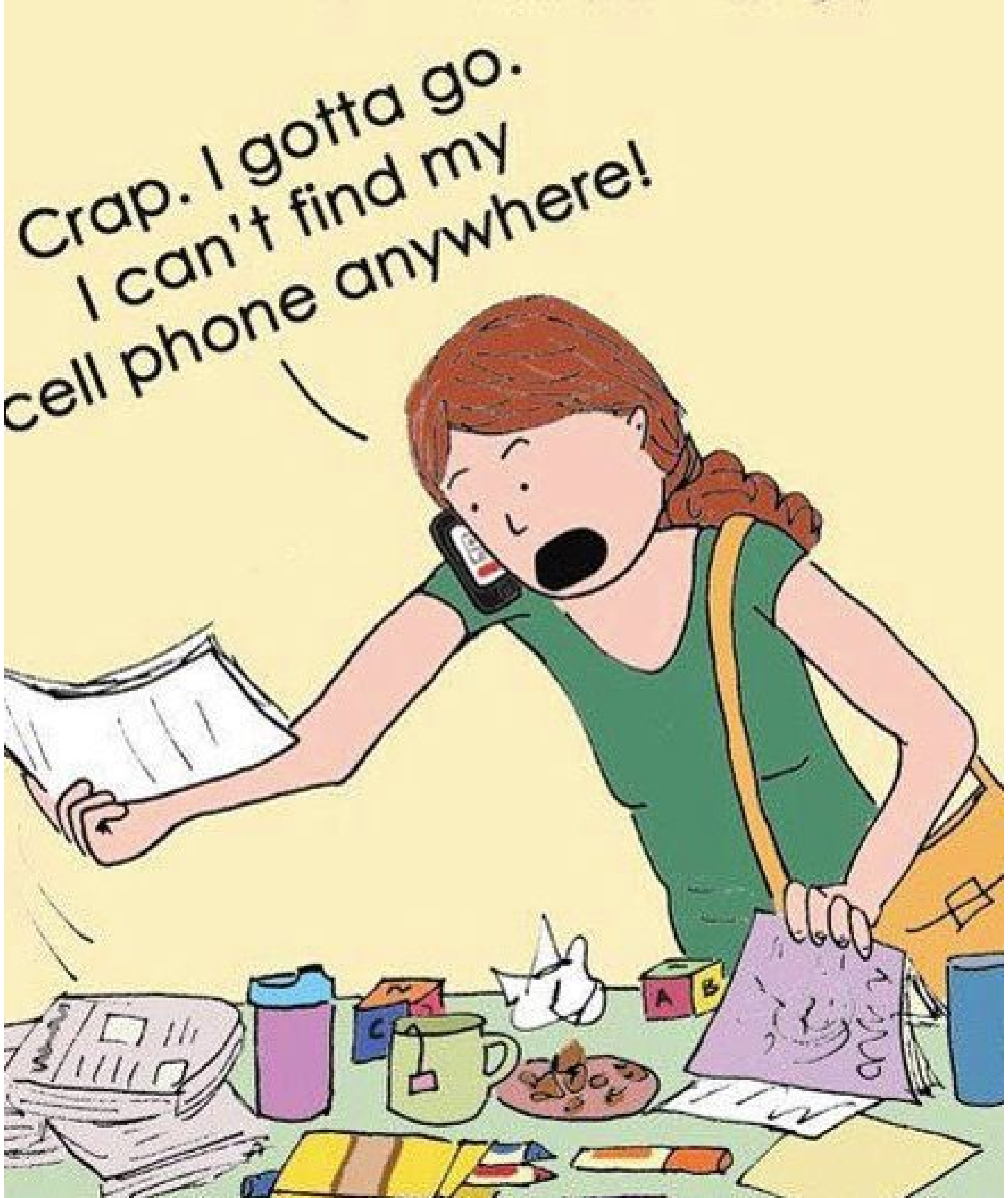
Systems Council Field of Interest:

The Systems Council integrates IEEE activities regarding aspects of multiple disciplines and specialty areas of systems engineering, and covers, but is not limited to, the following:

- Systems thinking
- Systems-engineering education, standards, processes, methodologies
- Systems modeling, simulation, integration, resilience
- Robust design, safety and human factors, security, usability, environmental aspects
- Product transition: design, production, test, deployment, disposal
- Program and project management
- Quality assurance
- Mission assurance
- Requirements development and management
- Risk management
- Systems architecture
- Systems of systems

IEEE Councils do not have members, but zero-cost “Participants”. You can sign up to be a Systems Council Participant when you renew your IEEE membership each year.

Please do not use cell phones during the lecture



Topics Covered

- Systems Engineering for Complex Systems – Brief Overview
    - Needs Analysis in complex systems
    - Identifying scope and roles
    - Developing high level system architecture
    - Establishing the operating environment
    - Understanding the problem
    - Identifying system operation
    - Understanding system capability

As well as:

- Principles of Balanced Design
  - Trade-offs between performance – development cost – unit cost – reliability/maintainability/availability - durability
- Risk Management
  - How to assess risks in your design
  - How to balance risk against performance
- Systems Engineering as an “Umbrella” discipline
- Need for understanding of all aspects of design: electrical, mechanical, software/hardware, etc etc etc
- Domain knowledge (i.e. entertainment system, military system, space system, etc)

But First - - -

- **What is Systems Engineering?**

Systems Engineering

Is NOT DOORS*

Or SLATE**

Or any one of the other dozen popular requirements management tools

*Doors = IBM's Dynamic Object-Oriented Requirements System for requirements management
**Slate = MathWorks' System Level Automation Tool for Engineers

Systems Engineering IS

---an interdisciplinary field of engineering that focuses on how complex engineering projects should be designed and managed over the life cycle of the project. Issues such as logistics, the coordination of different teams, and automatic control of machinery become more difficult when dealing with large, complex projects. Systems engineering deals with work-processes and tools to handle such projects, and it overlaps with both technical and human-centered disciplines such as control engineering, industrial engineering, organizational studies, and project management*.

Sort of an "engineering glue" that holds it all together

*Wikipedia definition of Systems Engineering

Or the alternate definition---

Systems 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, then proceeding with design synthesis and system validation while considering the complete problem:
Operations
Cost & Schedule
Performance
Training & Support
Test
Disposal
Manufacturing

Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs*.

*International Council on Systems Engineering (INCOSE) definition

Delving a bit deeper - - -

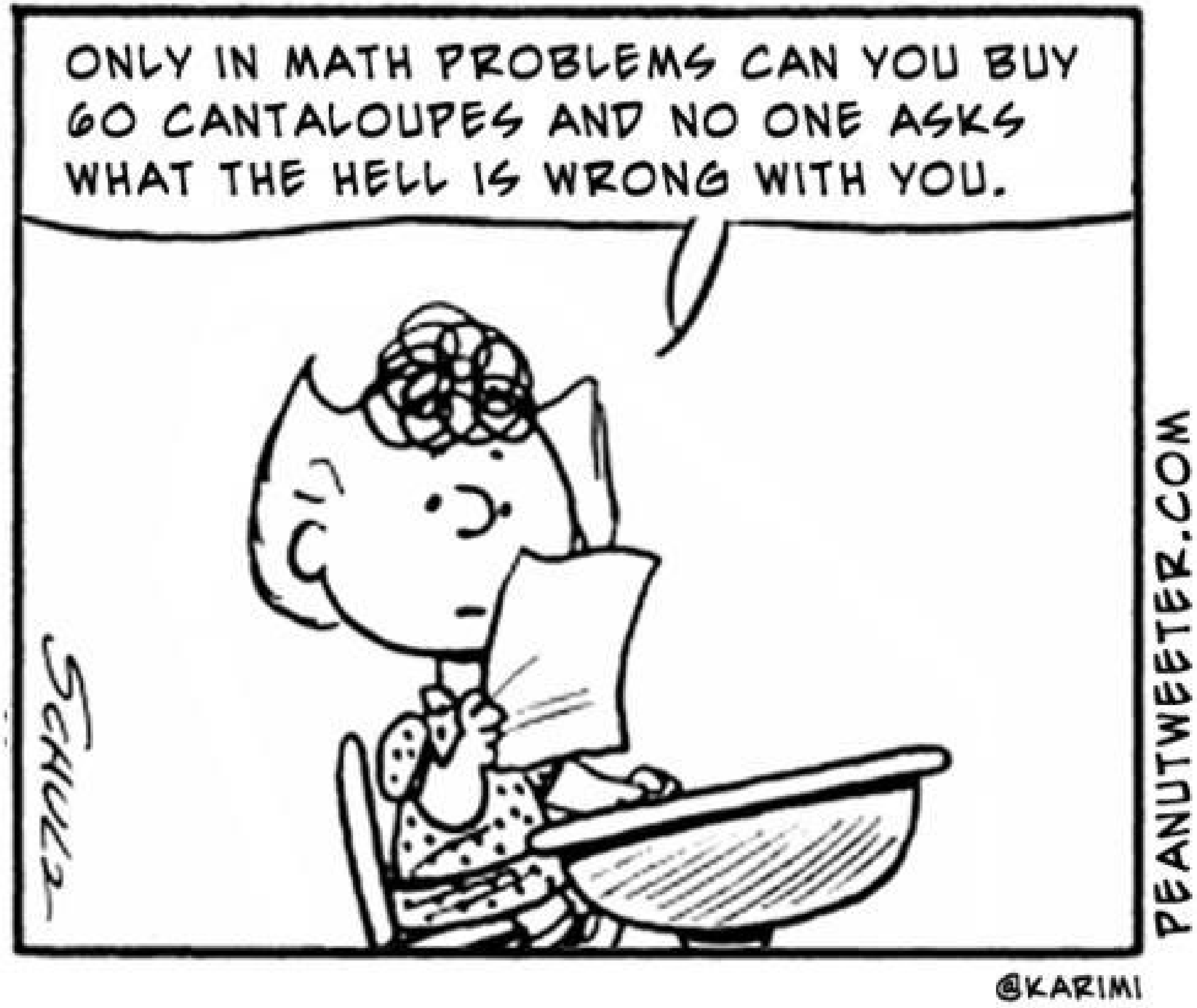
Systems engineering focuses on analyzing and eliciting customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem, the system lifecycle. Oliver* *et al.* claim that the systems engineering process can be decomposed into a
***Systems Engineering Technical Process*, and a**
***Systems Engineering Management Process*.**

Within Oliver's* model, the goal of the Management Process is to organize the technical effort in the lifecycle, while the Technical Process includes *assessing available information*, *defining effectiveness measures*, to *create a behavior model*, *create a structure model*, *perform trade-off analysis*, and *create sequential build & test plan*.

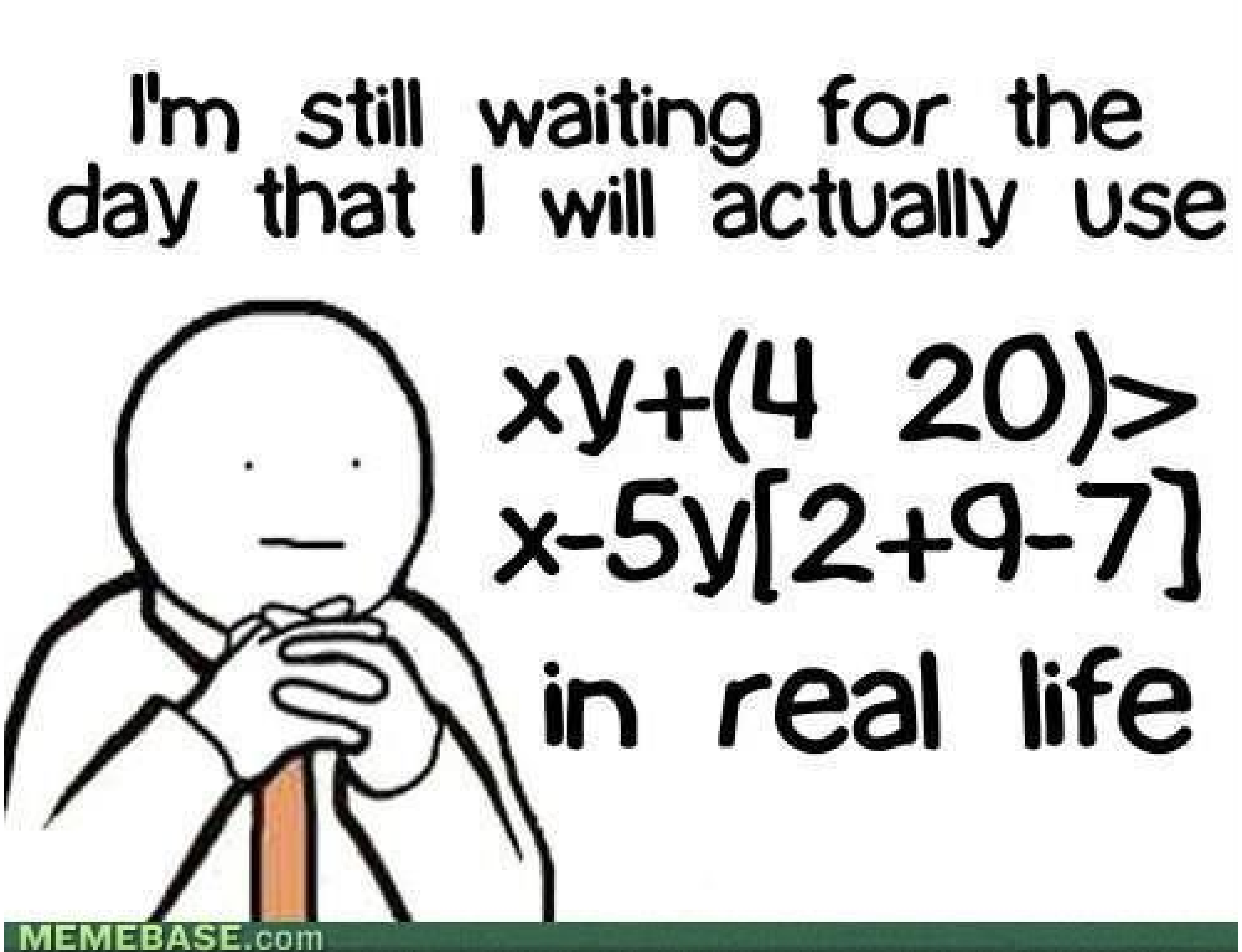
Depending on their application, although there are several models that are used in the industry, all of them aim to identify the relation between the various stages mentioned above and incorporate feedback. Examples of such models include the Waterfall model and the VEE model.

*Oliver, David W; Kelliher, T.P; Keegan, J.G; *Engineering Complex Systems with Models and Objects, McGraw-Hill, 1997*

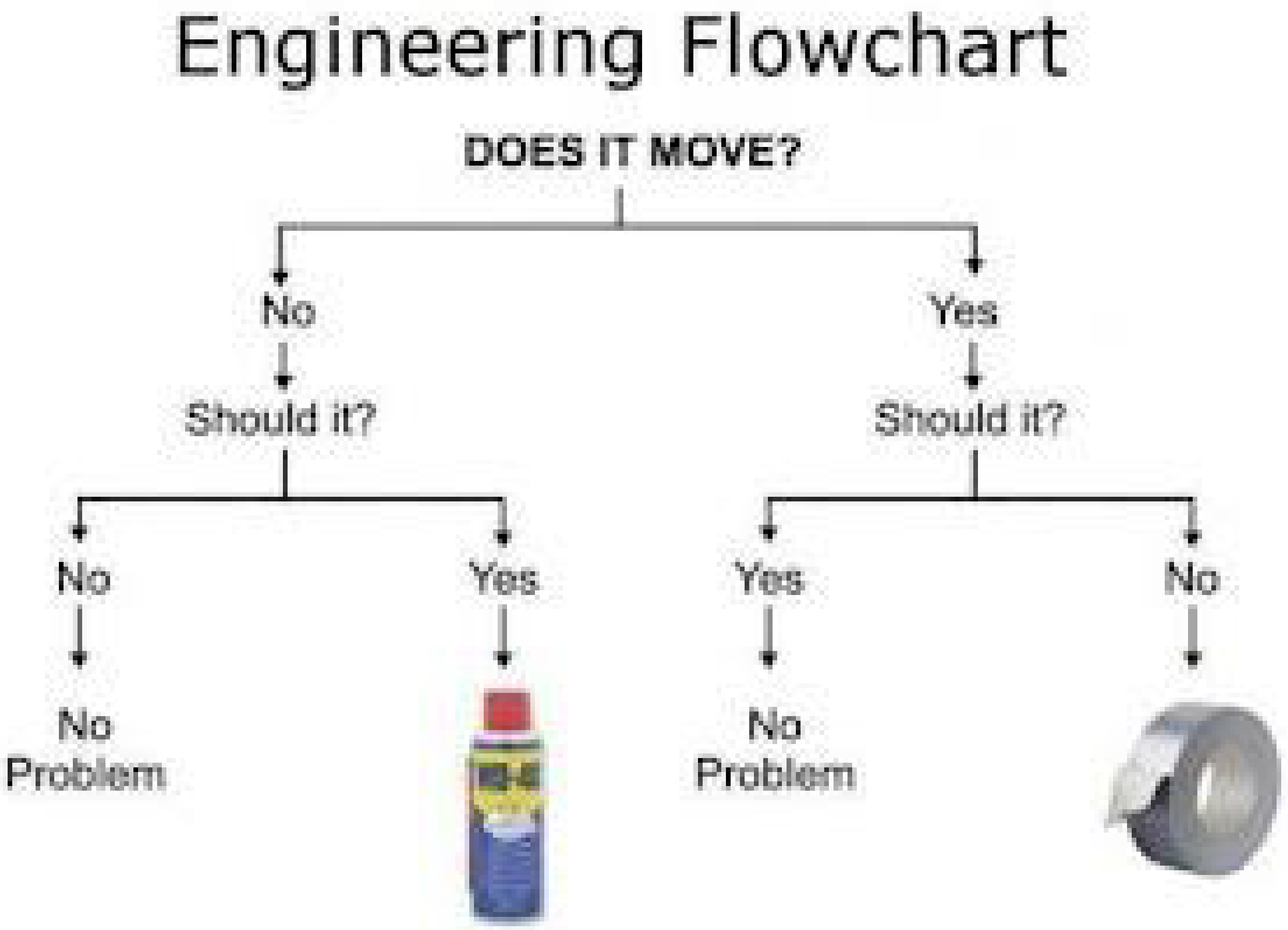
Systems Engineering is not Math or Formula based!



You won’t be facing this

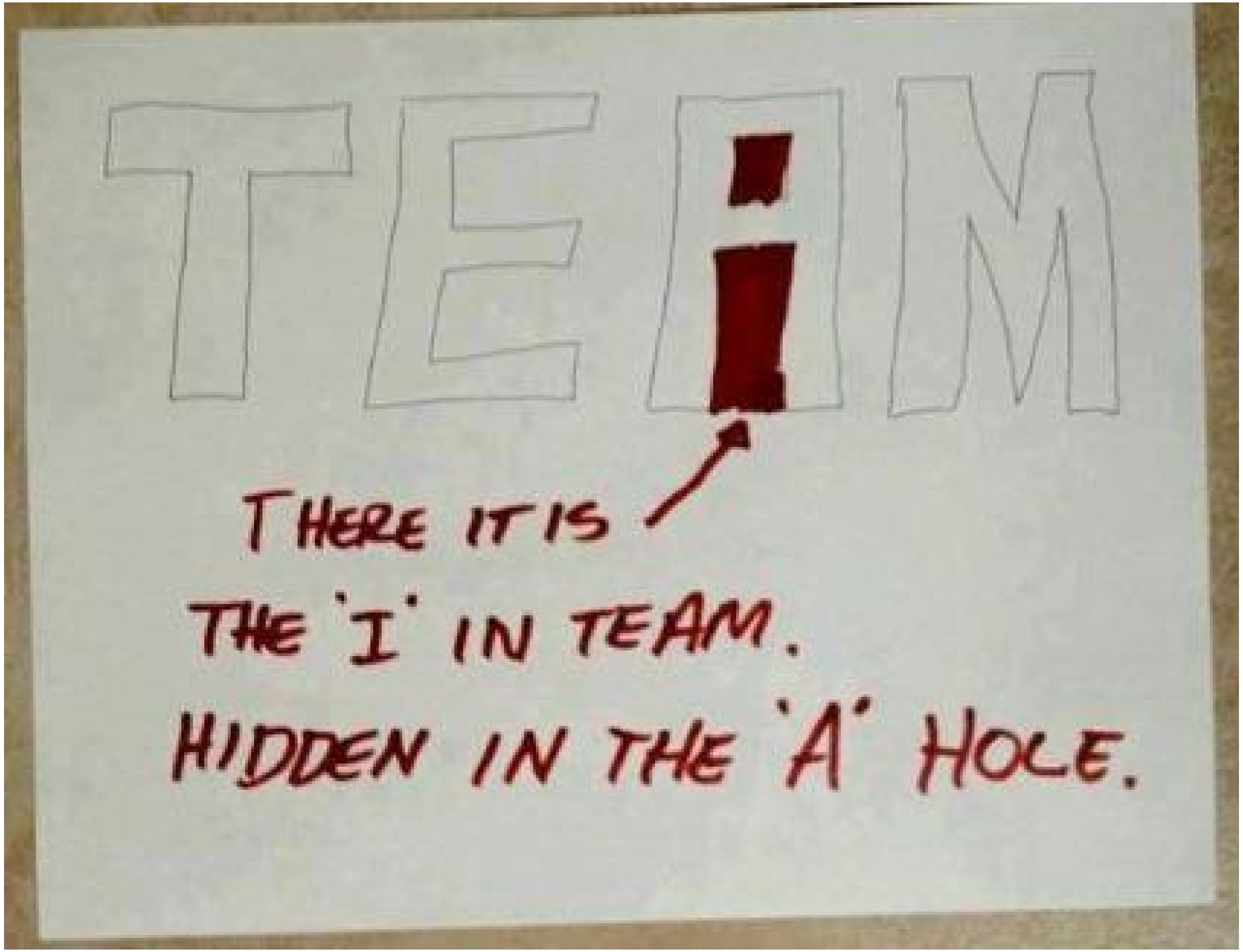


Systems Engineering represents a logical process for design

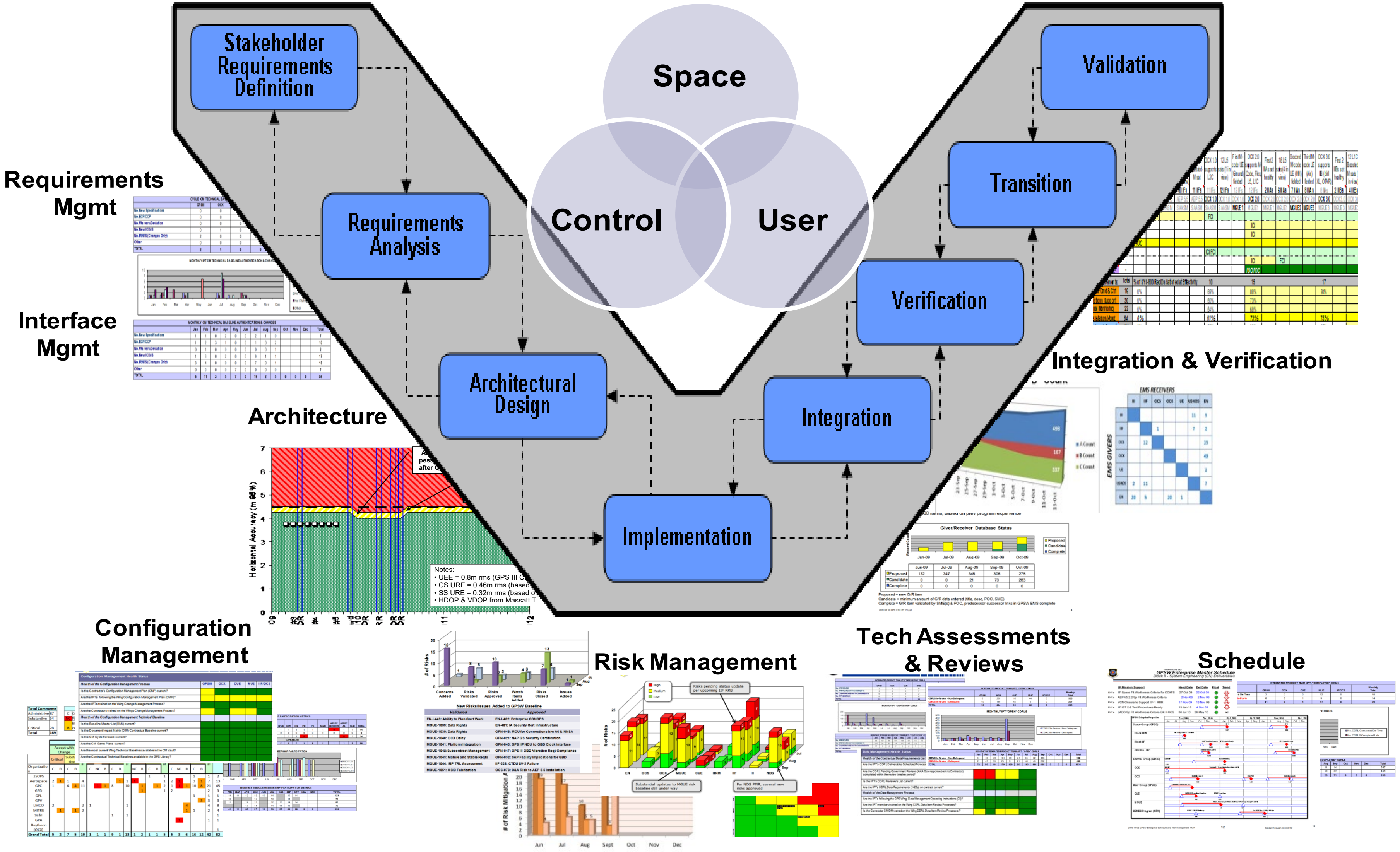


But you will soon realize that Systems Engineering really is- - -

. . . As long as you have the right team



Updated SE "V" 2017



Stakeholder Requirements Definition

Requirements Analysis

Architectural Design

Implementation

Integration

Verification

Transition

Validation

Space

Control

User

Requirements Mgmt

Interface Mgmt

Architecture

Configuration Management

Risk Management

Tech Assessments & Reviews

Schedule

Integration & Verification

Balanced Design

- A balanced design is one that balances performance, cost and schedule. It always includes trade-offs between performance factors, such as functionality, reliability, supportability/maintainability, as well as inherent system characteristics such as safety, security, resilience, obsolescence/upgradeability, and similar. And cost means *development cost* as well as *selling cost*.
- Balanced design is important because it provides the optimum solution to the problem when all rationalized requirements have been considered. It is important that the design result in a practical and feasible product, where all performance requirements themselves are both practical and feasible, and with acceptable risk
- The entire system life cycle must also be considered.

Achieving a Balanced Design in complex systems is one of the greatest challenges we face

System Life Cycle

- Systems life cycle is the overall depiction of the target system, from concept to development to deployment and use and eventual disposal
- It includes three main stages or phases:
  - Concept Development
  - Engineering Development
  - Post-development
- It is important because unless all stages of the life cycle are considered and executed carefully, the system will be flawed

Complex Systems

Systems Engineering is becoming more and more important as systems become more and more complex

- **A major attribute of complex systems is that the component systems (or subsystems) in of themselves are fully functional, meaning they can perform a stand-alone function independent of the complex system of which they become a part.**
- **A major issue in the design of most complex systems or systems-of-systems is that many of the component subsystems are legacy systems whose design cannot be altered or changed, and this often puts design constraints, often serious, on the design of the overall system. This point is repeated further on in this tutorial.**

Concept Development

- **In Concept Development we do the following;**
- **Needs Analysis**
  - **System Studies**
  - **Technology Assessment**
  - **Operational Analysis (CONOPS)**
- **Concept Exploration**
  - **Requirements Analysis**
  - **Concept Synthesis**
  - **Feasibility Studies**
- **Concept Definition**
  - **Analysis of Alternatives**
  - **Functional Architecture**
  - **Physical Architecture**

We will focus on the extra emphasis needed to do Needs Analysis and Risk Management for complex systems

Engineering Development

- The Engineering Development Phase includes:
- Advanced Design or Advanced Development
  - Requirements Analysis
  - Functional Analysis and Design
  - Development Testing
  - Risk Reduction (which includes prototyping)
- Engineering Design
  - Requirements Analysis (with focus on systems design and external interfaces and inherent characteristics such as safety, security, resilience, obsolescence/upgradability and similar)
  - Functional Analysis and Design (with focus on component interfaces and modularity)
  - Component Design
  - Design Validation

Engineering Development -2

- **Software Systems Engineering**
  - Design and coding of software per the hardware-software partitioning and the overall physical and functional architecture
- **Integration & Evaluation (also Systems Integration, Systems Test, System Test & Evaluation)**
  - Assembling all components and performing adequate testing to assure that all performance parameters are met
    - Factory testing is done to assure all functions are performing as designed
    - Operational test is done in the user environment to assure that the product is suitable for the use for which it was intended
    - NOTE that some aspects of system design cannot be tested per se, such as some elements of safety, security, and resilience – these must be tested partly by analysis

Post-development

- The Post-development Phase includes:
- Production
  - Assuring that the factory and manufacturing processes are in place to build the product and all of its components at lowest practical cost and highest practical quality
- Operations & Support
  - Delivery of the system to the customer
  - Installation of the system and final test (often called acceptance test) in the user facility
  - Preparation of training and maintenance materials, and training of the users of the system (often an option)
  - Providing initial spares (often an option)

Systems Engineering Life Cycle

The Systems Engineering Life Cycle is composed of the following: (not to be confused with the System life cycle described above)

- Concept exploration & development
- Advanced Development & Engineering Design
- Integration & Test/Evaluation

This is where we ascertain that we are developing a balanced design

This step is even more important for complex systems

More on Balanced Design

A balanced design is one that balances performance, cost and schedule, with minimum risk, but here is where we also make sure all the “ilities” are considered:

- Reliability - design to minimize failures during use
- Availability - design to assure the product is available for use when needed (note- a highly reliable product will have high availability)
- Maintainability - design to facilitate maintenance of the product when it does fail (note- ease of maintenance also contributes to high availability)
- Producibility - assuring that the product can readily be built at reasonable cost and high quality

As well as certain inherently-required critical elements such as safety, security, resilience, obsolescence/upgradability, and similar

Complex Systems

- **Complex systems are represented by a hierarchal structure that consists of major systems or subsystems, components, subcomponents, and parts.**

- **A major attribute of complex systems is that the component systems (or subsystems) in of themselves <u>are or can be fully functional,</u> meaning they can perform a stand-alone function independent of the complex system of which they become a part. A major issue in the design of most complex systems or systems-of-systems is that many of the component subsystems are legacy systems whose design cannot be altered or changed, and this often outs design constraints, often serious, on the design of the overall system. This point is clarified further on in this tutorial.**

The more complex the target system, the higher the inherent risk

Risk Management

The 4 important attributes of Risk Management are

- Identify all potential risks deemed inherent in the target system development environment
- Estimate the probability of occurrence of each risk
- Assess the impact or criticality of each identified risk
- Establish a risk mitigation strategy for each identified risk.

Risk management must continually assess

the outcomes of the risk mitigation strategy,

including the viability of alternate or fallback plans

Risks tend to be greater and more subtle when designing complex systems

Architecture

Architecture is the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time. Architecture defines the overall structure of a product, both physically as well as operational. There are several perspectives, including:

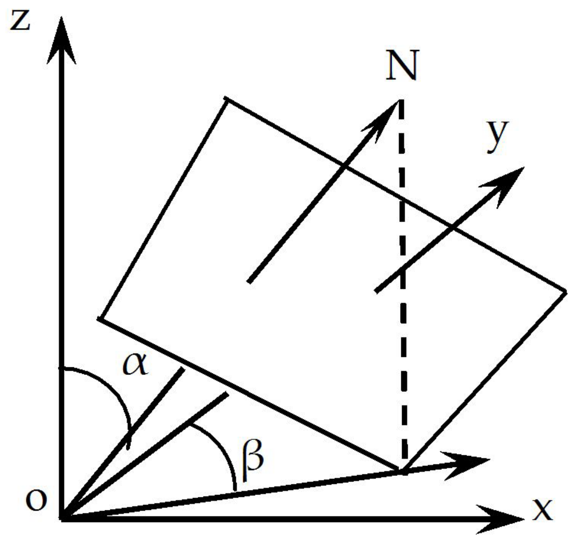
- Operational View – the system representation from the user/operator perspective
- Logical View – the system representation from the customer or manager perspective
- Physical View – the system representation from the designer/engineer's perspective

Developing the system architecture is far more difficult in designing a typical complex system

Model-based Systems Engineering - MBE

Models are mathematical representations of the product.

Use of models in design can simplify the design process by reducing or eliminating the need to prototype certain aspects of the design.



Numerous models are available from vendors

It is beyond the scope of this tutorial to address the various models

Where multiple models are used, it is important that there is consistency and that Interfaces exist between model inputs/outputs where the models need interaction

Again, MBSE is far more complex when designing complex systems

Digital Thread

- Model-based systems engineering has been around for decades
- Advances in computer power and digital modeling capability now allow for more accurate digital models of systems
  - More rapid designs
  - Reduced cost
  - Greater accuracy
  - Improved manufacturing and test
  - Reduced risk
- Ability to interconnect various digital models has been termed the “digital thread” and is becoming a major systems engineering tool
- Supercomputers allow for multi-physics models using advanced computational fluid dynamics (CFD)
  - CREATE Program, “Computational Research and Engineering Acquisition Tools and Environments”

What is a good systems engineer?

A good systems engineer is

- Skilled in multiple disciplines and multiple domains
- Technically competent
- Highly experienced in his/her field
- A systems thinker
- Can see the forest as well as the trees
- A natural leader
- A good listener
- A good negotiator
- Patient
- Analytical
- Highly motivated
- Humble

Needs Analysis

Needs Analysis and validation is one of the most important first steps in systems engineering.

The systems engineer must understand the customer's needs before he can start any conceptual design solutions.

Steps include an in-depth system study:

- **Identifying all input conditions, requirements, plans, etc.**
- **Identifying the "why" of all stated requirements in terms of operational needs, constraints, environment etc.**
- **Clarifying each stated requirement—develop understandings**
- **Correct uncertainties in each stated requirement**

What we are doing here is defining the problem as completely and accurately as possible

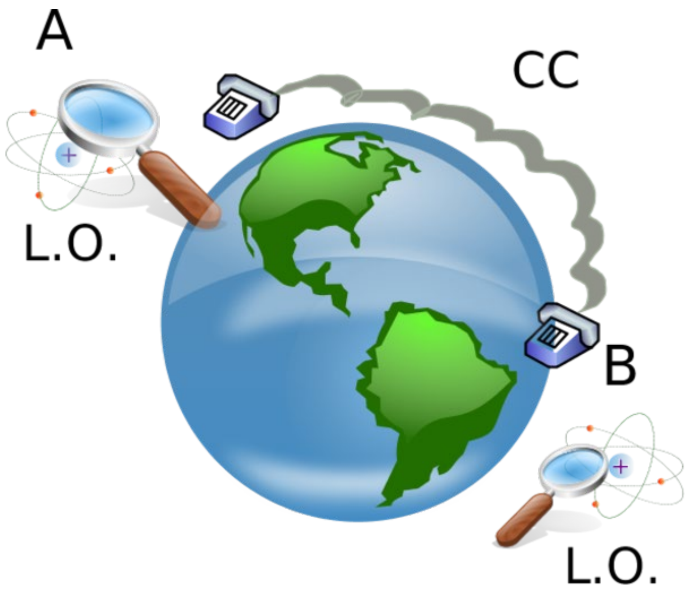
And the customer may not realize that he needs certain inherent things such safety, security, resilience and similar, which can add complications

Needs Analysis defines system capabilities and effectiveness

Operational Requirements

What is also needed is a set of system performance parameters and system operational requirements, all derived from the customers' statements of need and the inherent characteristics we have mentioned

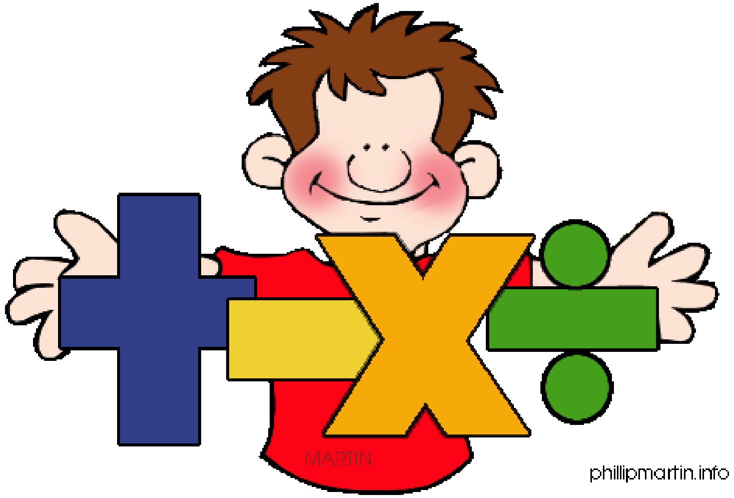
This will be used through the concept exploration and development process as part of our measure of effectiveness, to ascertain that we are on the right path to satisfy the stated need.



Systems Requirements

Translating the customer needs including the inherent elements into system requirements is one of the most critical front-end steps of any design if you wish to achieve a successful design.

This is the job of the systems engineer and is the reason why it is critical to do systems engineering as the 1st step in the overall design process.
This important concept is repeated further along in this tutorial.



System Requirements…

System Requirements are developed in the concept exploration phase of development. Activities that comprise concept exploration are:

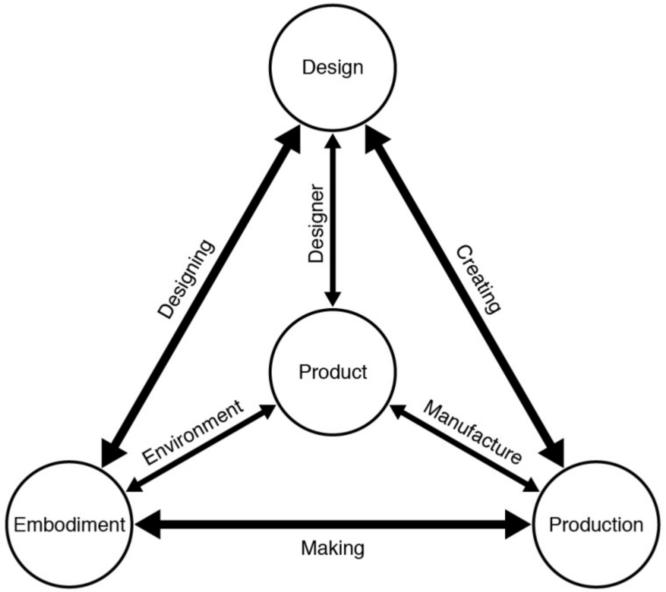
- Operational Requirements Analysis – make sure the operational requirements are complete and consistent
- Implementing Concept Exploration – refining the functional characteristics derived from the needs analysis
- Performance Requirements Formulation – deriving all required functions and operational parameters
- Performance Requirements Validation – ensuring that the derived performance requirements are operationally valid with respect to the CONOPS; assuring that the requirements are practical and feasible

Development of system requirements is really a challenge with complex systems, since the capability of any legacy systems must be factored in

System Requirements…

It is important to remember that system requirements includes a lot more than how high it flies or how fast it goes. The following are also inherent system performance requirements that a customer does not always recognize:

- Safety
- Reliability and resilience
- Security
- Maintainability/supportability/upgradability
- Operator interface/ergonomics
- Longevity (system life)
- Transportability/mobility
- Size/weight
- Environment
- And similar…



Real vs Implied Requirements

- It is essential to understand what the “real” system needs and requirements are – the ones that are essential – vs. the implied requirements, the like-to-haves, etc.
- Customers will often express the need for features or capabilities that aren’t really necessary to meet the real, bottom line requirement

  This occurs quite often in military systems
- It is essential to separate the “real” need or requirement from the “nice-to-haves” in order to successfully design a competent, cost-effective solution for the customer
- This can occur during a “requirements negotiation” with the customer in which the real requirements are separated from the non-essential ones – we call it Requirements Validation
- In complex systems, determining all of the “real” requirements can be a significant challenge

CONOPS

CONOPS – Concept of Operations – tells us how the new system is going to be used

The customer must provide sufficient information for the designers to understand exactly how the system is going to be used

- Who will use it
- How will they use it
- How often will they use it
- Under what conditions will they use it
- How will it be stored
- How will it get to the user or use site

A good CONOPS will provide this important information

Requirements Validation

This is the step or process in which we also start to apply the Practical and Feasible rule

Is each requirement practical?

(and we must include the inherent performance characteristics of every system, as discussed earlier)

Is each requirement feasible?

Often we get conflicting requirements; those that are in direct or implied conflict with each other and it is the job of the systems engineer to separate the real need from the "non-essential" needs

We are validating that every stated AND implied requirement is essential in terms of the objective system performance.

Feasibility Studies

- **The systems engineering team will begin with feasibility studies to see which of the potential solutions that have been suggested offer the most feasible in terms of truly satisfying the customer needs**
- **Trade studies start to come into play as we do tradeoff analyses of the proposed potential solutions to see which have the greatest potential to solve the customer needs, in terms of performance (overall) , cost (development and sell), and schedule (development and production), and with acceptable risk.**
- **Tradeoff analyses can get complicated when we start to trade off the customer real requirements and the inherent systems characteristics yhat we have been discussing, but this is an essential function of the systems engineers**

Needs Validation is Continuous

We must continuously and systematically examine the validity of the results of all the previous steps.

We must determine that the basic concept is sound and that a solid case can be made that the proposed conceptual solution is the best option to meet the customer needs

- At affordable cost
- Within reasonable schedule
- And at acceptable risk

It may be wise to develop an operational effectiveness model by which to measure the effectiveness of the system

This will entail using and expanding a full set of system performance parameters and system operational requirements derived from the customer need and CONOPS

Finally - - - How Much SE is “right”?

- Managers now ask, how much systems engineering do I need?
- The answer is – it depends!
  - For new, custom complex defense systems, a study by the National Defense Industrial Association’s Systems Engineering Division and the Carnegie Mellon University determined that 15% of the total design effort should be expended on systems engineering to assure a successful program.
  - Expenditures of less than amount this resulted in programs that ran over budget, over schedule, and missed important performance parameters.

Summary

- Systems engineering is one of the most complex engineering jobs going
- Competent systems engineers are in demand, everywhere
- In my area of defense, we find it takes at least 15, often 20 years of engineering, notably systems engineering, to gain enough experience to be a good lead systems engineer
- Systems engineers tend to be innate-
  - Born of natural "systems thinkers"
  - Those who can easily see the big picture and can readily visualize the consequences of everything they do, every action they take
- At Raytheon, we find that our best lead systems engineers become our best Program Managers
- Complex Systems, or Systems-of-Systems, has pretty much become quite commonplace and drives even greater need for solid, experienced systems engineers.

Early SE avoids trouble and helps Assure Success

It is essential to employ systems engineering early in the program to help ensure that the design is practical and feasible, that all requirements have been validated and found realistic, that all realistic needs have been met and that all needs are valid needs and not just desires, that a good CONOPS has been developed and validated and understood, and that the target system can be developed and produced within the customer cost and schedule constraints.

We call this "starting the program right". We need to not only build the right thing, but also build the thing right!

Discussion