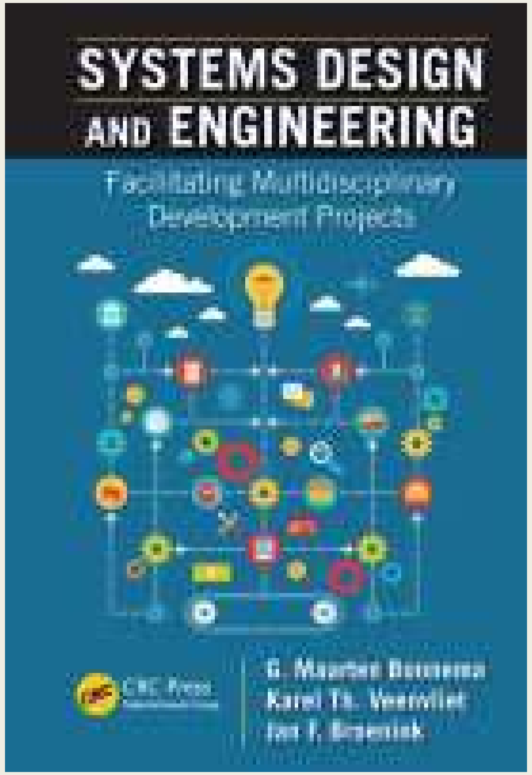
SYSTEMS ENGINEERING IN 45 MINUTES – A PRAGMATIC OVERVIEW

8. June 2018

Assoc. Professor Cecilia Haskins, ESEP

Topics



- Who am I?
- Design versus analysis
- What do systems engineers do?
- The core of systems engineering
- Systems engineering as a research method
- Closing
- With thanks to Assoc. Professor Maarten Bonnema of Delft for his inspiration, and drawings (and his book - Bonnema, Veenvliet and Broenink (2016))

Introduction – who am I ?

- Long industrial experience base – over 30 years
- ESEP = Expert Systems Engineering Professional
- Started teaching in 2000 to share experience with next generation
- Editor of the INCOSE SE Handbook, v.3 used for certification testing 2004-2015
- Earned a PhD in systems engineering in 2008
- Inventor of the SE method SPADE
- Associate Professor in Systems Engineering at NTNU since 2013

Design versus analysis

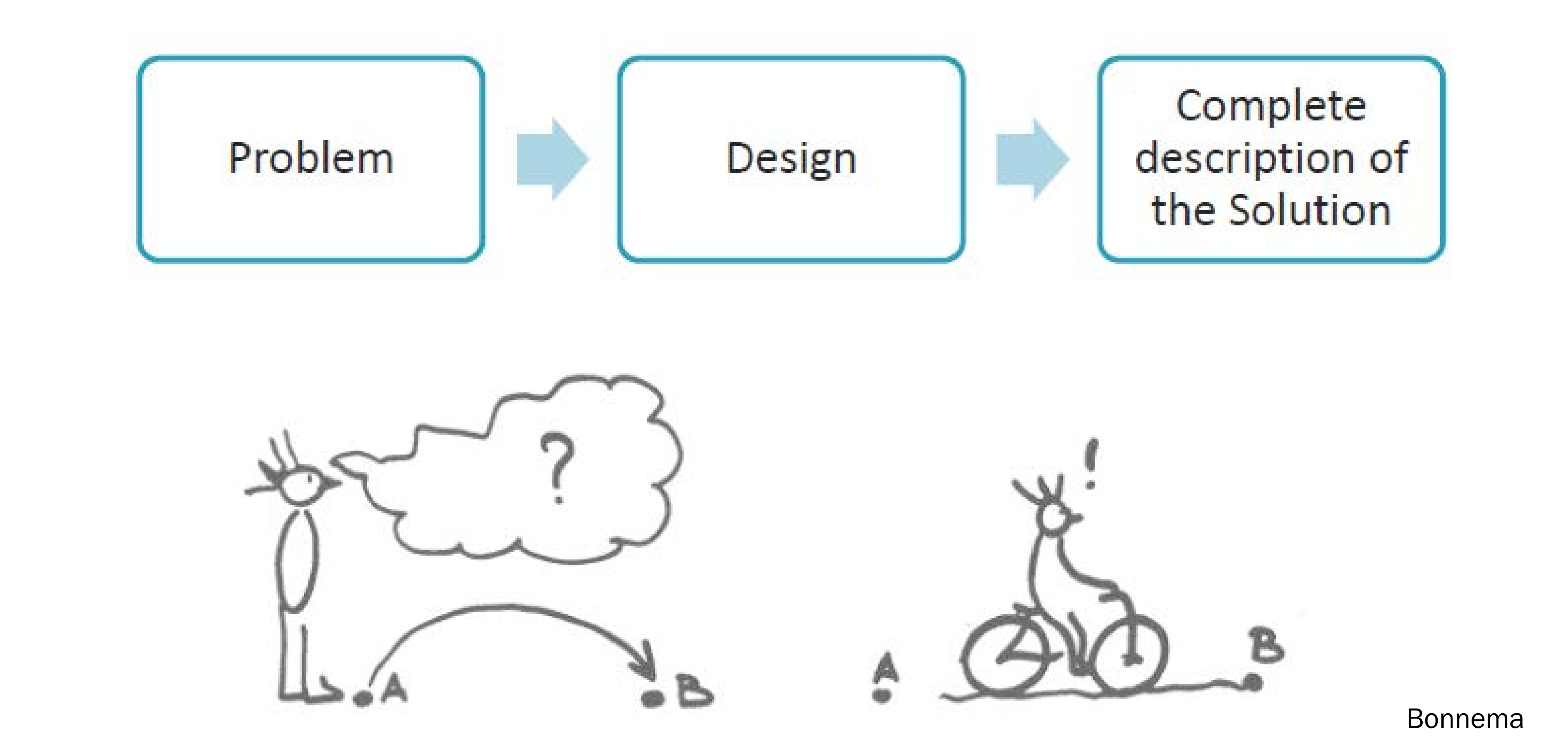
Analysis – based on an existing 'something'

- In-depth study to create knowledge
- Search for the one "correct" answer
- Descartes – reductionist – break up something to study the parts
- Piaget – constructionist – expand the view to study interactions,

Design – creating 'something' new

- Looking at a big picture to create knowledge
- Many possible 'answers' – no one best
- Many points of view on what it is, and how to achieve or define a good design

How design is often viewed



Questions that *SHOULD* pop up immediately

- What is the problem?
  - *And how do I learn more about it?*
  - *Is this the *real* problem, or a symptom?*
- What constraints do I have:
  - *To do the design*
  - *For the solution*
- Who has the need for a solution to this problem?
- What will be the impact of the solution on people, the environment, etc.
- …
- Interestingly, development projects often start from an idea for the solution…

Thinking of solutions as products versus systems

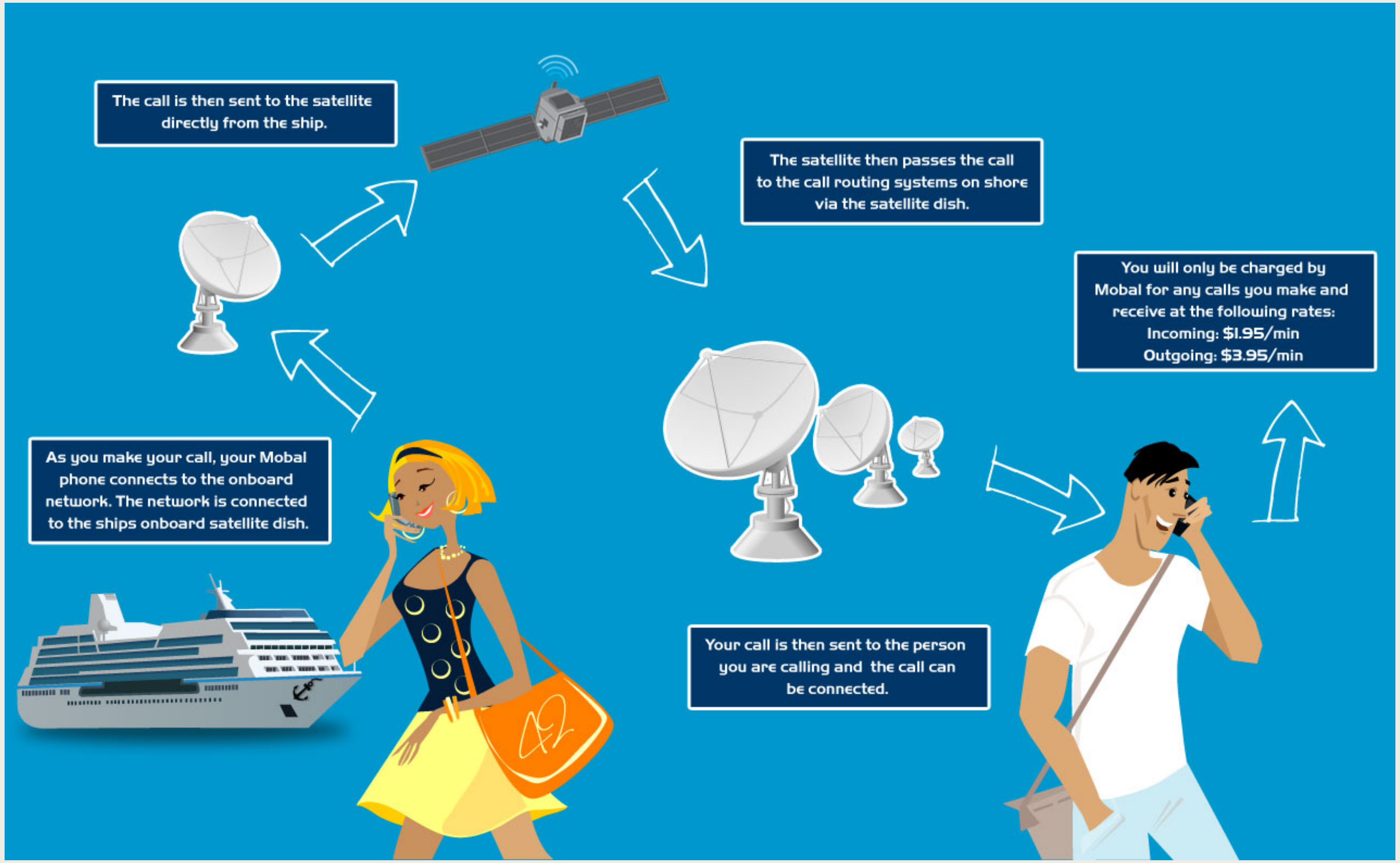
Product

- Mostly one-way influence
- Limited users / stakeholders
- Attempt to achieve optimization

Systems

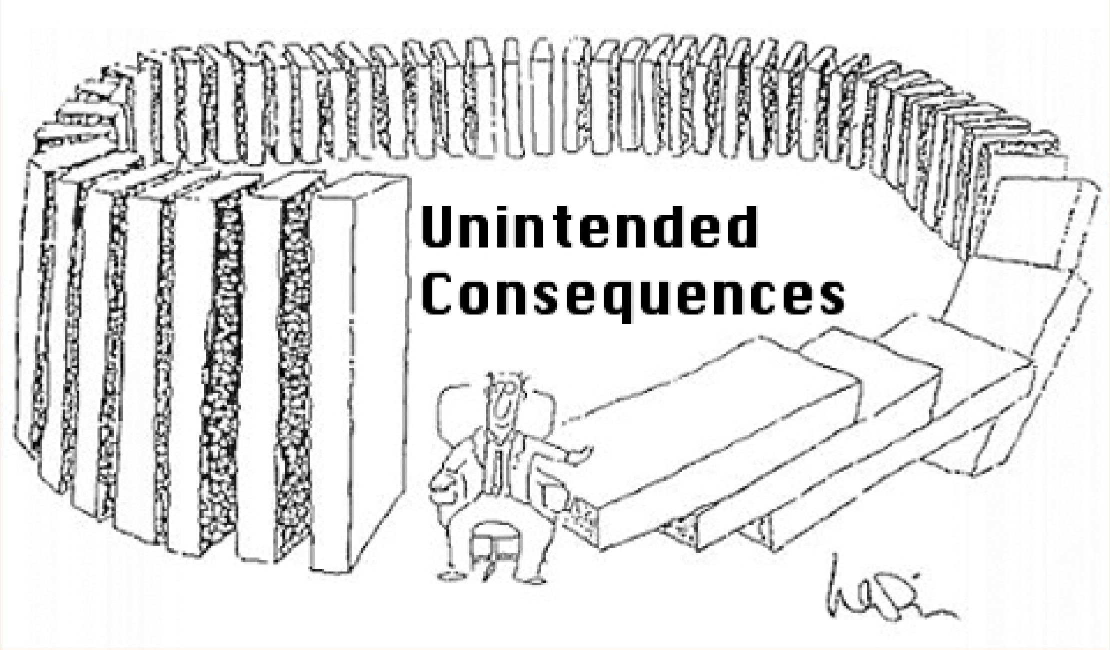
- Two-way influence
- Multiple users / stakeholders
- Trade-offs – not optimization
- Large scope
  - *with multiple subsystems*
  - *and multiple domains*

For example – product, a mobile telephone – versus system, i.e., a system of systems that supports an individual conversation



The consequence

- Most engineers are educated in creating products
- But… we don't create products, rather systems
- These systems present an increasing impact on the environment, and vice versa
- Resulting in various problems and unintended consequences



This brings us to the core of Systems Engineering!

- Take a systems perspective – big picture thinking
  Begin by separating WHAT and HOW WELL from *how*
- Focus on interfaces rather than individual components
- Decision-making with uncertain and incomplete information

Taking the system perspective

- Make the system central, instead of limiting focus to details
  - *Zoom out to see the whole*
  - *Zoom in on crucial issues only*
- Practice systems thinking
- Big picture; includes temporal concerns
- Anticipate unintended consequences
- Avoid the Pitfall of staying too general or superficial


Habits of a Systems Thinker
Seeks to understand the big picture
Observes how elements within systems change over time, generating patterns and trends
Size
Years
Recognizes that a system's structure generates its behavior
Identifies the circular nature of complex cause and effect relationships
Makes meaningful connections within and between systems
Changes perspectives to increase understanding
Surfaces and tests assumptions
Sugar
Vitamins
Super Star Cereal
Considers an issue fully and resists the urge to come to a quick conclusion
Considers how mental models affect current reality and the future
Uses understanding of system structure to identify possible leverage actions
Considers short-term, long-term and unintended consequences of actions
BANK
Donut Shop
Pays attention to accumulations and their rates of change
Recognizes the impact of time delays when exploring cause and effect relationships
Checks results and changes actions if needed: "successive approximation"
Second Edition ©2014, 2010 Systems Thinking in Schools, Waters Foundation, www.watersfoundation.org

System composition

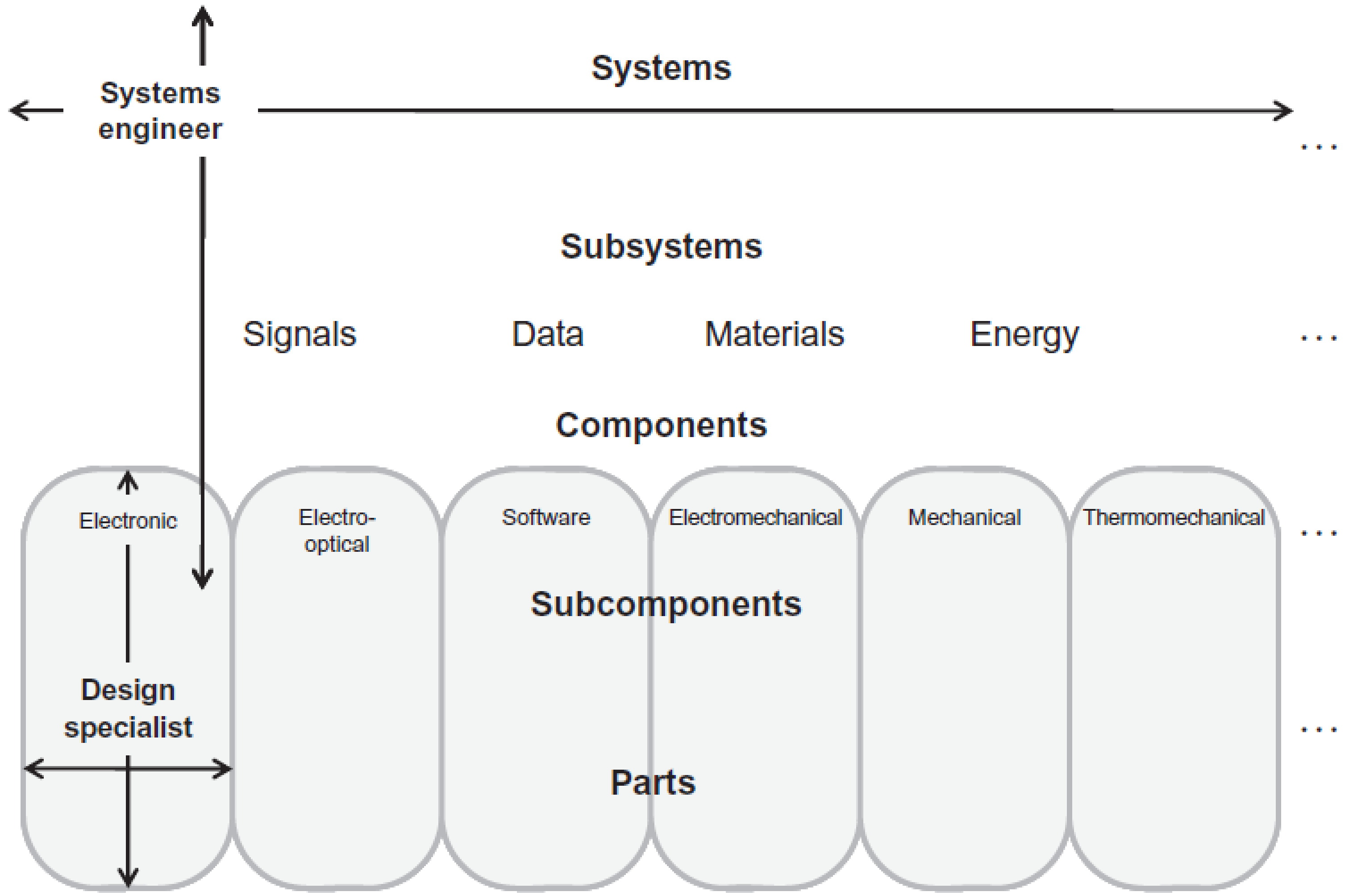


Figure 3.1. Knowledge domains of the systems engineer and the design specialist.

Interactions across the system boundary are external interfaces made visible in the **Context Diagram**

Interactions between parts are internal interfaces

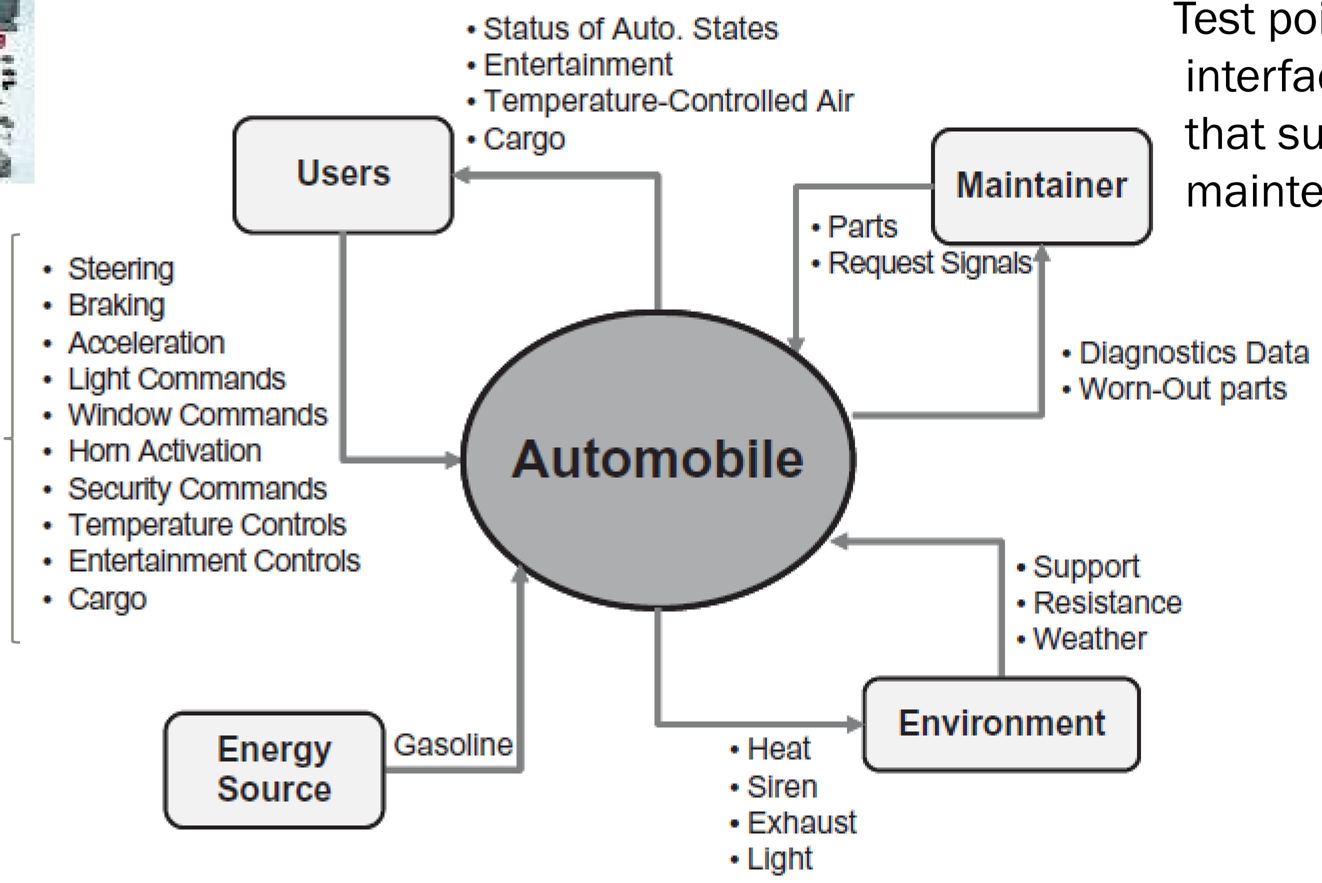


Figure 3.3. Context diagram for an automobile.

Test points are interfaces that support maintenance

Investigation – future phones

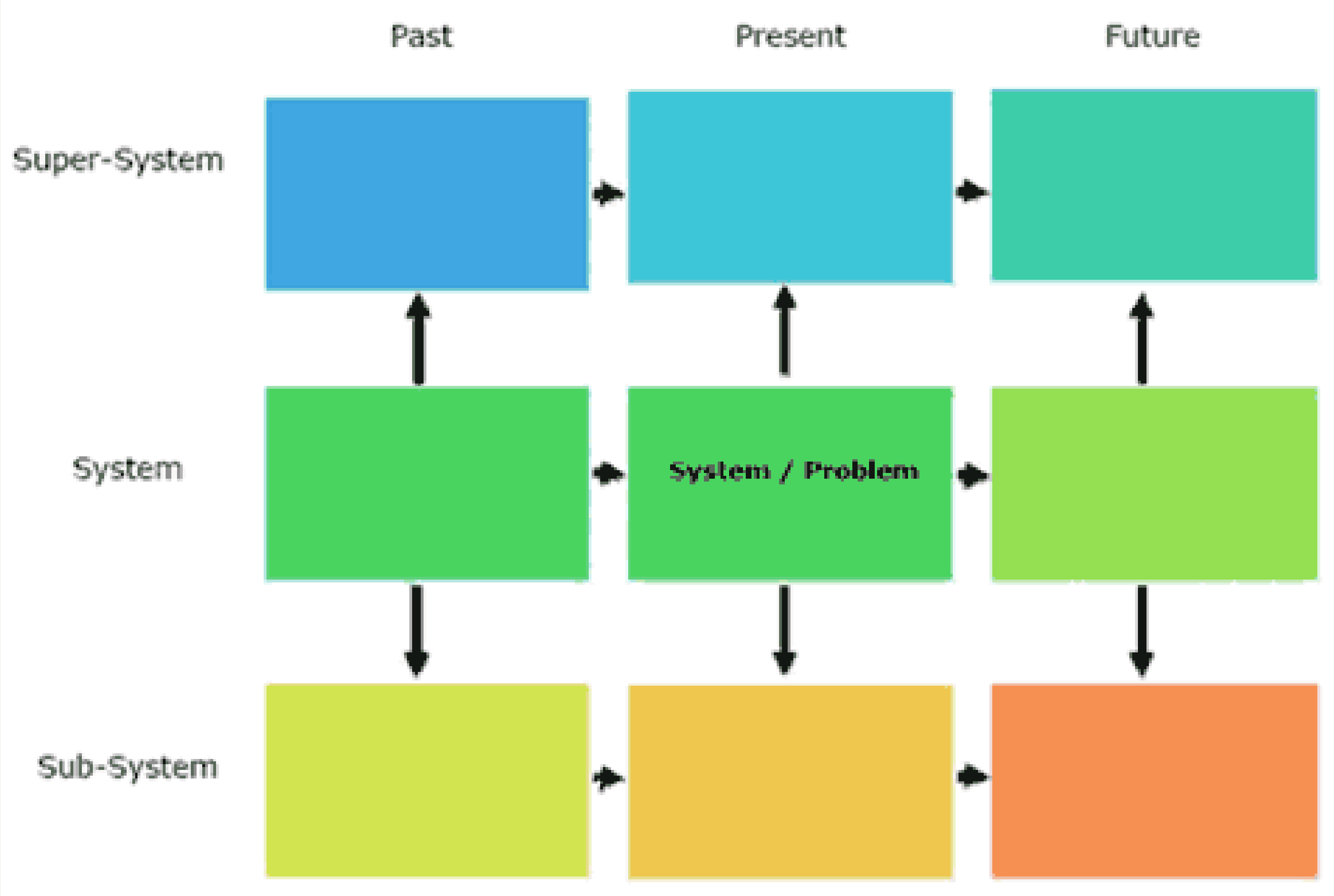
Problem domain

- What is the need for an extra type of phone in our line-up?
- What is the competition doing?
- What is the market pull?
- What are long-term developments?

Solution domain

- What new technologies have come up that might improve our phones?
- What are strengths and weaknesses of our current products?
- Describe the phone in three dimensions:
  - *Functional (what does it do)*
  - *Physical (how does it do it)*
  - *Performance (how well)*

Good tool – the nine-windows view (from TRIZ)



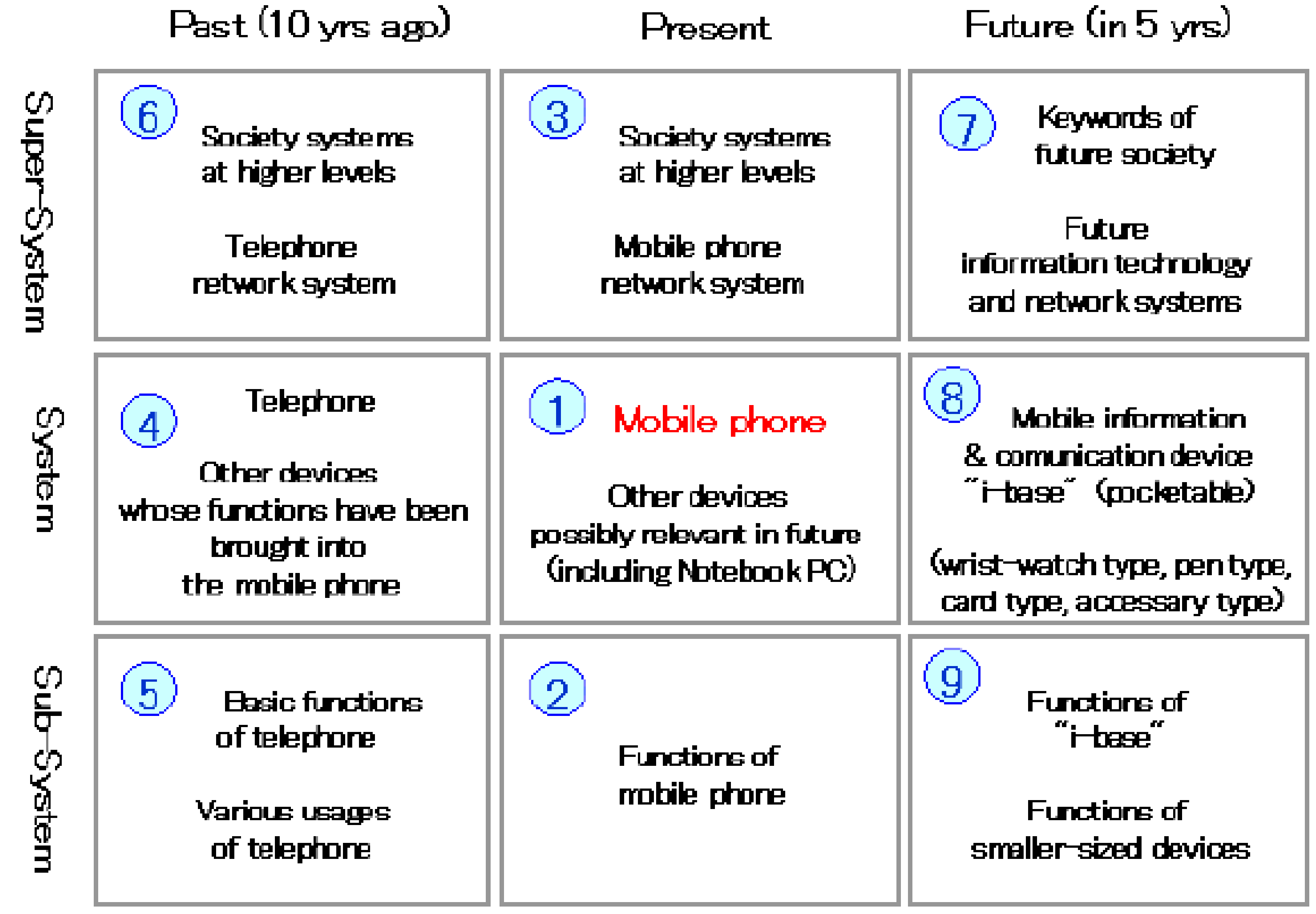
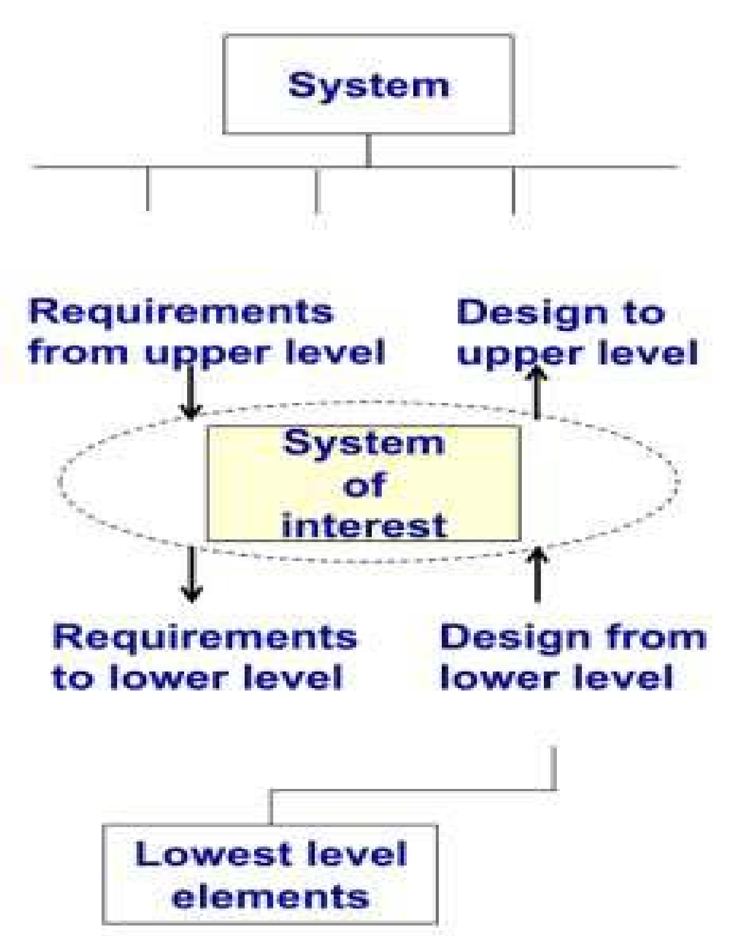


Figure 2. An example of description in the 9-window method: Forecasting the future of mobile phone

Systems perspective is layered



- One person's system is another person's subsystem
- Implication: Systems Engineering can be applied on different levels:
  - *System*
  - *Subsystem*
  - *Assembly*

- – And:
  - *Super-system*
  - *Society*

Separate WHAT and HOW WELL from *how*…

- "What" and "How much" relate to the problem domain/the need
- "How" relates to the solution domain
- NOT a linear flow, there is iteration and feedback involved throughout the process

What — Functions: expected behavior of the system

How well — Performance: quantified measures

How — Physical architecture: parts of the system and interfaces

There are two types of systems engineers --- those who pay attention to interfaces…

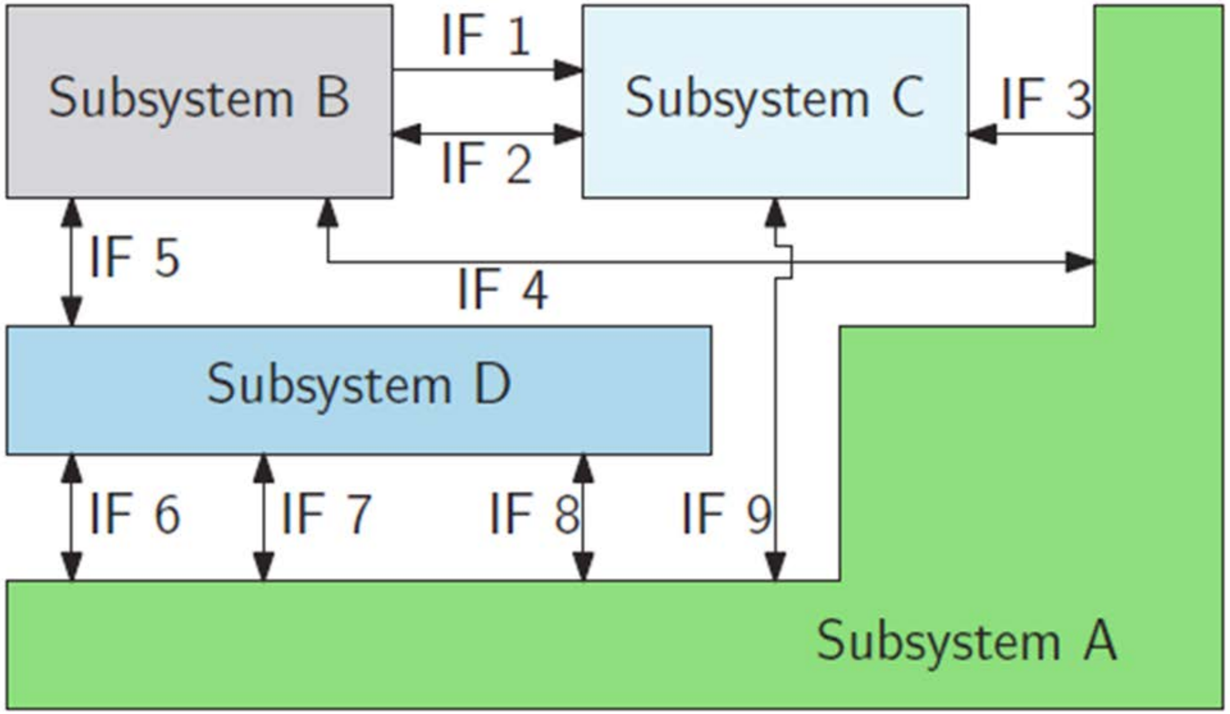
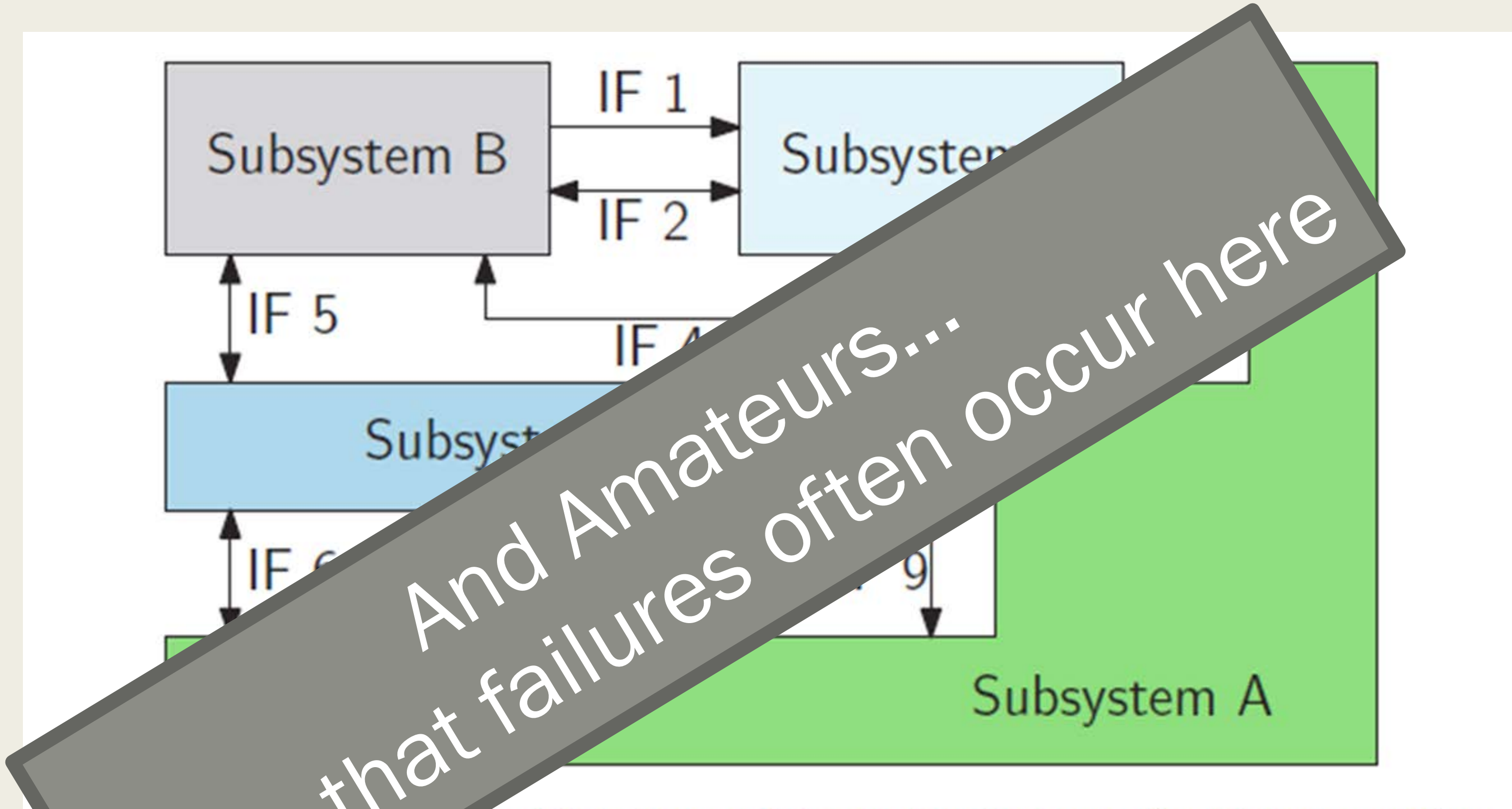


Figure 2.5 Subsystems and interfaces: all interaction between the subsystems takes place via identified and specified interfaces (IFx). Note that within a subsystem, a further splitting up can take place. Then again interfaces have to be defined.

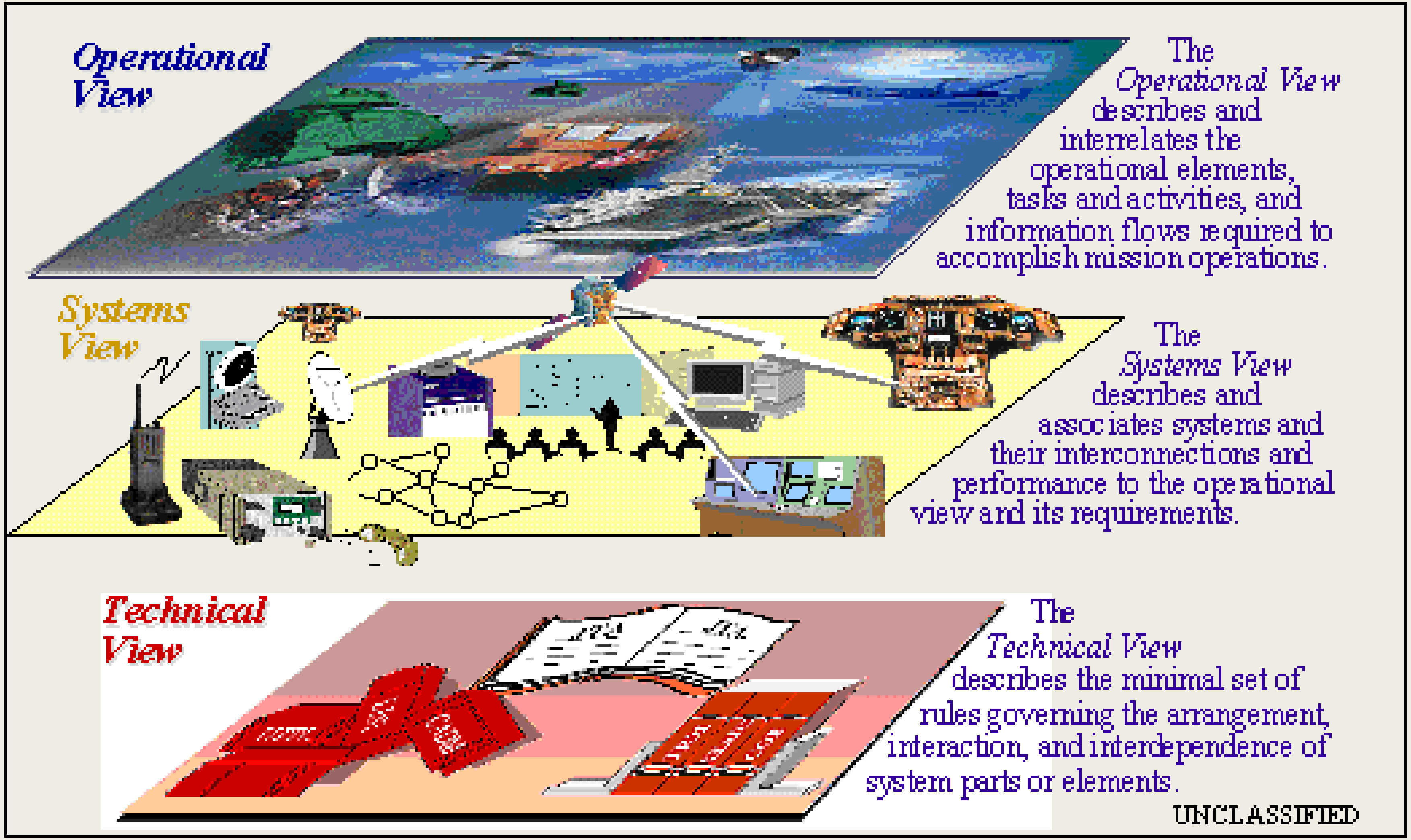
There are two types of systems engineers --- those who pay attention to interfaces…



…s and interfaces: all interaction between the subsystems takes p… …d and specified interfaces (IFx). Note that within a subsystem, a further… …up can take place. Then again interfaces have to be defined.

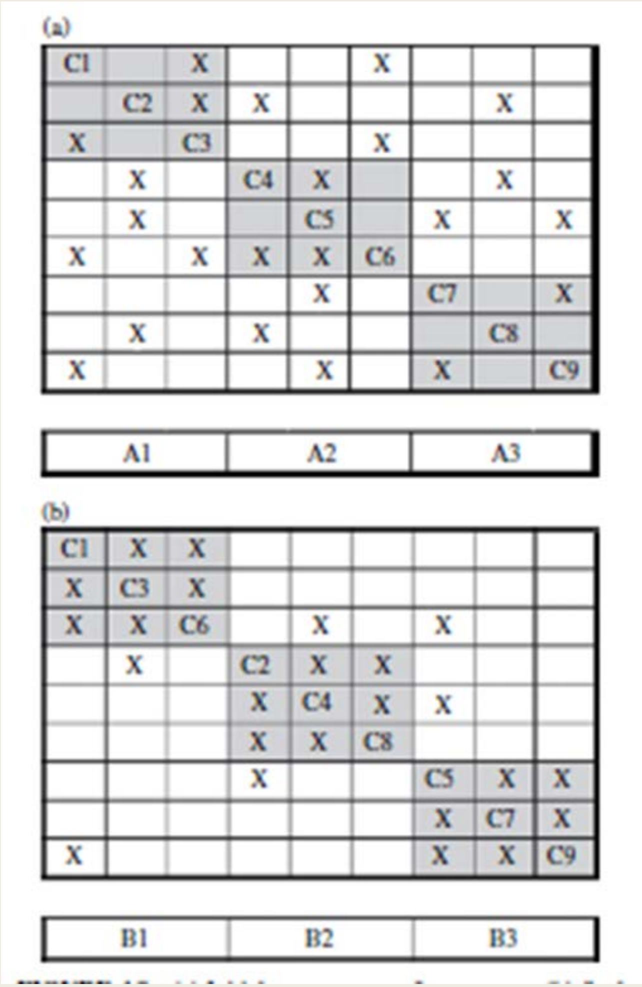
Interfaces are more than technical…

Architecture framework

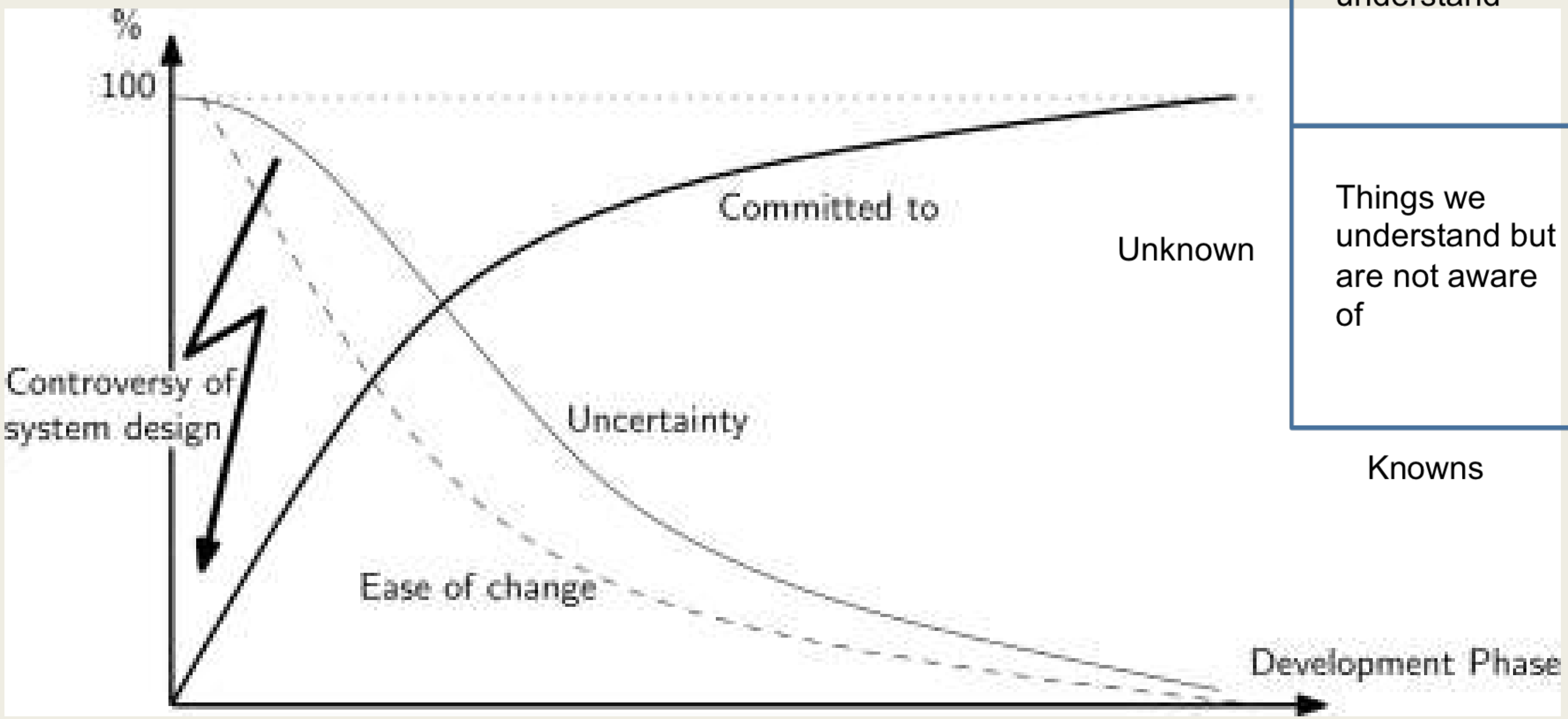


Address interfaces directly – the n-squared diagram is useful for this

- Make people responsible for system coherence
- Draw the system structure
- Look at it from both a functional and physical point of view
- Agree on the specifications for the interfaces, especially with multiple parties involved
- Illustrations shows usefulness of n-squared to identify efficient interfaces to define partitions



Uncertain and incomplete information during decision-making



	Knowns	Unknowns
Known	Things we are aware of and understand	Things we are aware of but don’t understand
Unknown	Things we understand but are not aware of	Things we are neither aware of nor understand

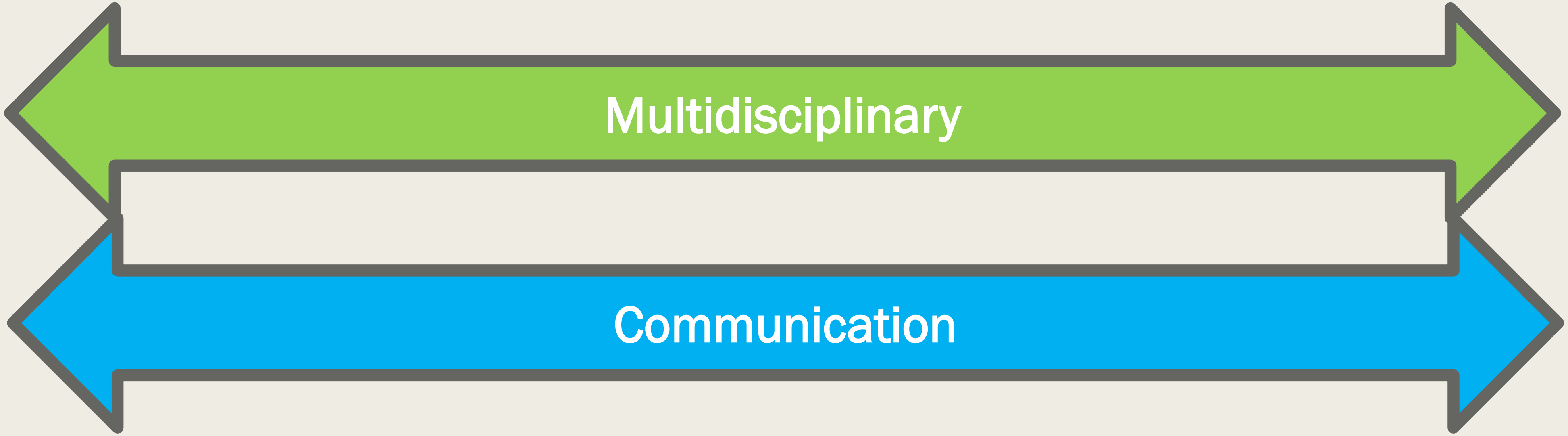
Systems engineering addresses uncertainty

- Uncertainty exists over the entire life cycle of a system, but ideally diminishes with time
- Rarely are decisions made with all the information needed to make an optimal decision
  - *Related to technical complexity*
    - Immature technology
  - *Related to consequences of prior actions*
    - Incomplete requirements
  - *Related to organizational issues*
    - Funding constraints

Status so far…
Any questions?

- Take a systems perspective – big picture thinking
  Begin by separating WHAT and How Much from *how*
- Focus on interfaces rather than individual components
- Decision-making with uncertain and incomplete information

The essence of Systems Engineering summarized in two words

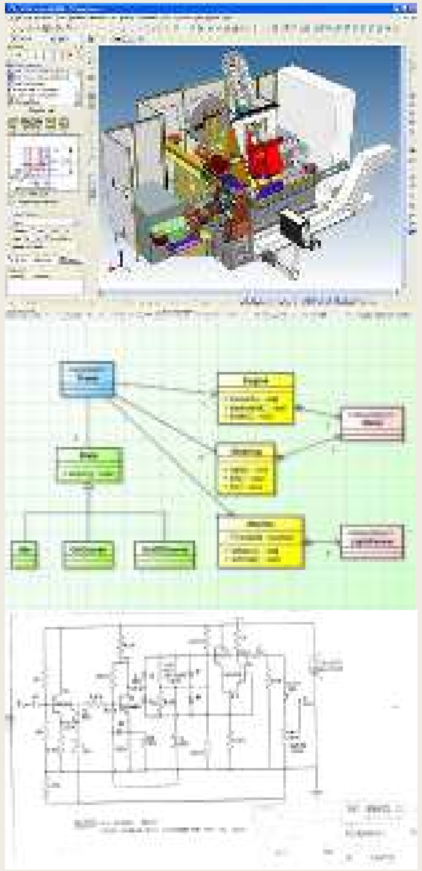


These are the two primary learning objectives of TPK4185 – Lego™ Lab
https://mediasite.ntnu.no/Mediasite/Catalog/catalogs/tpk4185

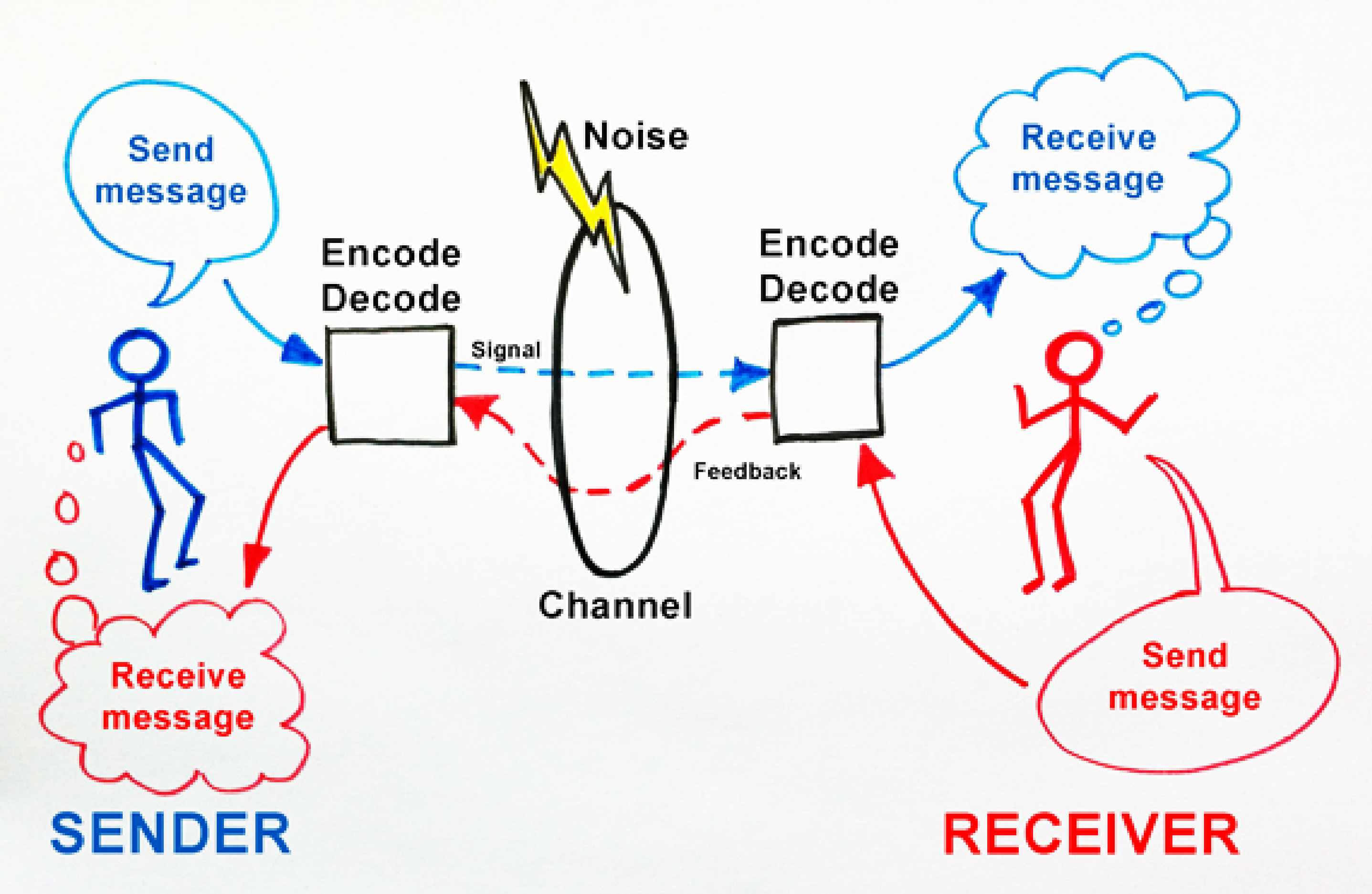
Multidisciplinarity

- Mechanical engineers use CAD systems
- Software engineers use UML and class diagrams
- Electrical engineers use circuit schematics
- Ergonomists use percentiles and mockups
- Etc.

- • But how to make sure we get ONE system?



Communication



The Shannon-Weaver model of communication

Systems Engineering bringing order to chaos

- This may all sound a bit chaotic
- And some chaos is inevitable
- A moderate level of chaos is even inspiring

- Vee model
  - *Requirements*
  - *Decomposition-composition*
  - *Verification and validation*
  - *Trade-off analysis*
  - *Various SE support tools*
  - *Model-based systems engineering*

MBSE – transition from documents to models

- David Long (CEO Vitech corporation):

  "Models are inherent to the way engineers think, reason, and communicate – and that includes systems engineers."

  "Fundamentally, MBSE is about making system-descriptive and analytical models explicit, coherent, and consistent. It is a natural evolution from low-fidelity representations in documents to higher fidelity, richer representations."

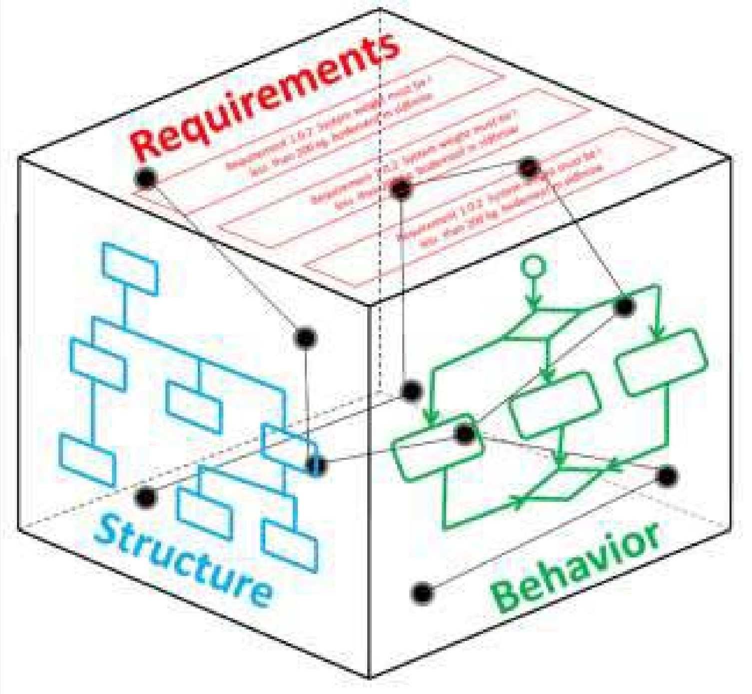
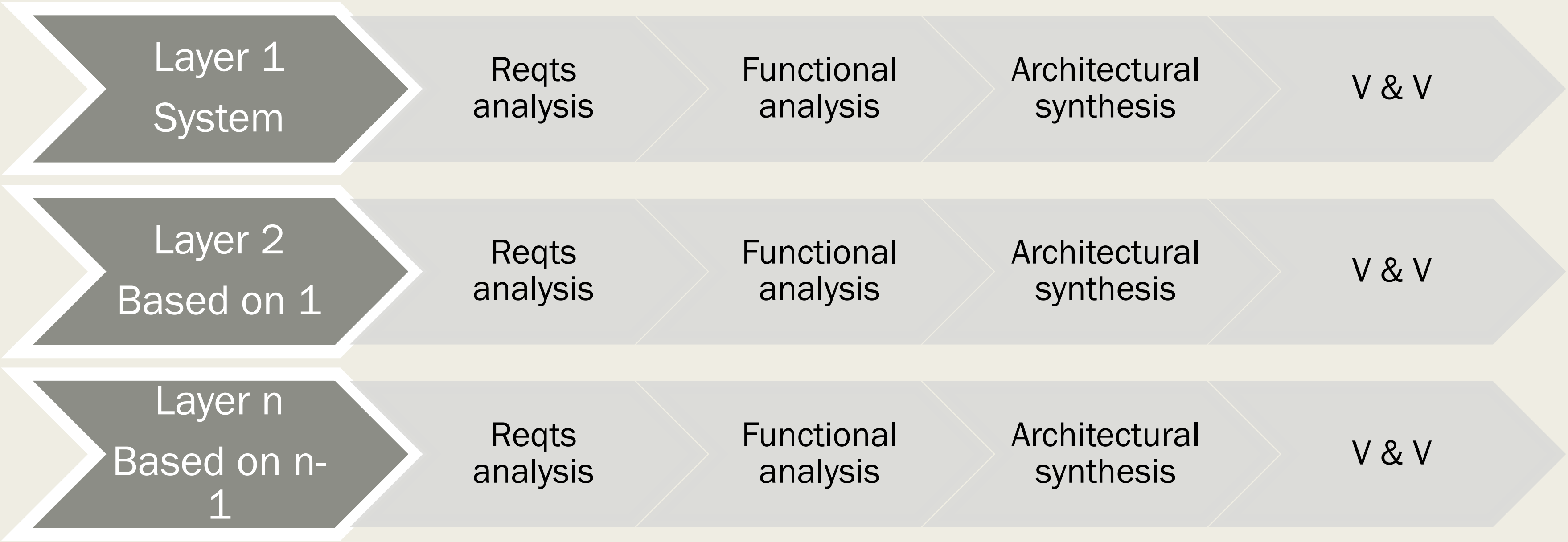


Illustration: Thomas Haduch, Director of Systems Engineering, RDECOM, http://armytechnology.armylive.dodlive.mil/index.php/2015/07/01/15-3/

Basic process – working in layers



- Focus of Systems Engineering
  - From Original Need
  - To Final Product
    - The Whole System
    - The Full System Life Cycle

iterate

Focus of Component Engineering on Detailed Design And Implementation

Need

- What needs are we trying to fill?
- What is wrong with the current situation?
- Is the need clearly articulated?

Operations Concept

- Who are the intended users?
- How will they use our products?
- How is this different from the present?

Functional Requirements

- What specific capability will we provide?
- To what level of detail?
- Are element interfaces well defined?

System Architecture

- What is the overall plan of attack?
- What elements make up the overall approach?
- Are these complete, logical, and consistent?

Allocated Requirements

- Which elements address which requirements?
- Is the allocation appropriate?
- Are there any unnecessary requirements?

Detailed Design

- Are the details correct?
- Do they meet the requirements?
- Are the interfaces satisfied?

Implementation

- Will the solution be satisfactory in terms of cost and schedule?
- Can we reuse existing pieces?

Test & Verification

- What is our evidence of success?
- Will the customer be happy?
- Will the users' needs be met?

Two Types of Iterations – types of work

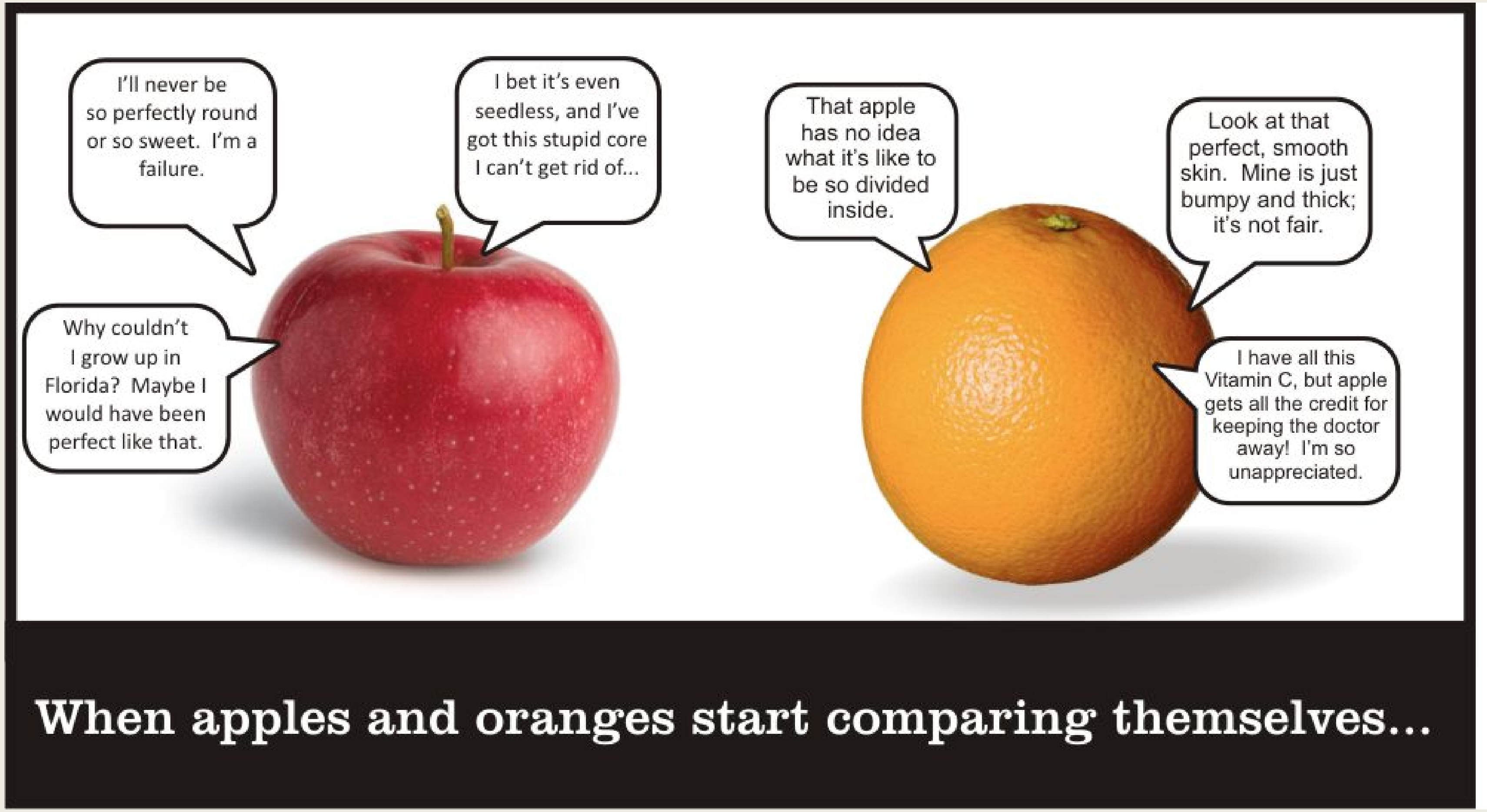
Planned iteration – value creating

- Caused by the need to "get it right the first time."
- We know where these iterations occur, but not necessarily how much work.
- Planned iterations should be facilitated by good processes, tools, and coordination.

Unplanned iteration – non-value creating

- Caused by errors and/or unforeseen problems.
- We generally cannot predict which unplanned iterations will occur.
- Unplanned iterations should be minimized.

Trade-offs – answers are often not clear-cut; often dissimilar properties are compared



Systems engineering – the 3 Rs

V&V

- VALIDATION – building the right system
  - *Managing customer expectations by reviewing the results of requirements analysis with the user to validate that you understand their needs*
  - *DO THIS EARLY IN THE PROCESS*
- VERIFICATION – building the system right
  - *Quality assurance testing against requirements to verify that the correct behavior is exhibited in the solution*
  - *DO THIS OFTEN IN THE PROCESS*

Lessons learned

- Managing the system design requires insights into the **system interactions** and their ramifications.
- **Interface requirements reconciliation** requires agreement among multiple subsystem requirement authors who may not have the same interests.
- Address system integration issues as early as possible, and **reduce late rework**.
  - *Good understanding of the system interactions*
  - *Good coordination among organizations*
  - *Organizational boundaries can cause inefficiency in information flow especially when system knowledge is dispersed in a tacit versus explicit form*

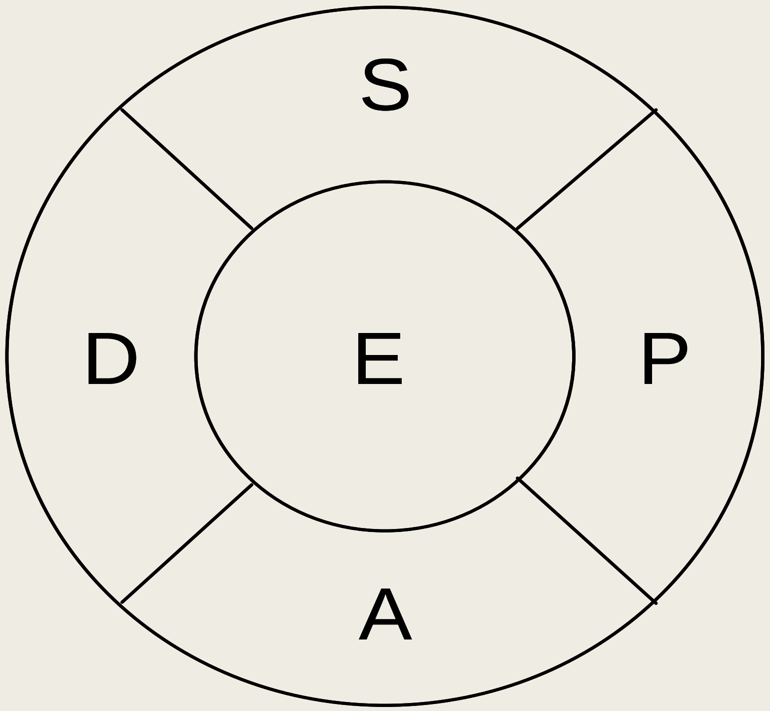
SE AS A RESEARCH METHOD

A generic SE Process – closely matches problem solving and scientific method

- Identify the system stakeholders and define the boundary of the system
- Analyse the stakeholders' needs in a life cycle context and their rationale – formulate a problem statement
- Define the functional, operational, financial and physical requirements to satisfy the needs under anticipated system changes, challenges and disturbances over the life cycle – including concerns for Health, Safety, Environment
- Search for technologies and other options to solve the problem
- Recommend and specify a system solution through iterative trade-off evaluations, where requirements are satisfied and accepted by the stakeholders
- Monitor the performance of the system for possible improvements

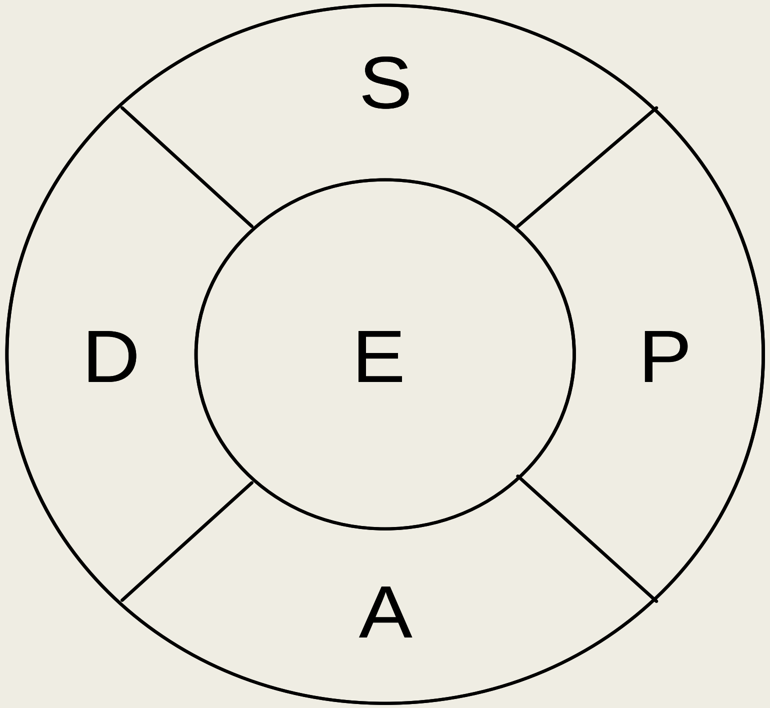
Essential SE framework

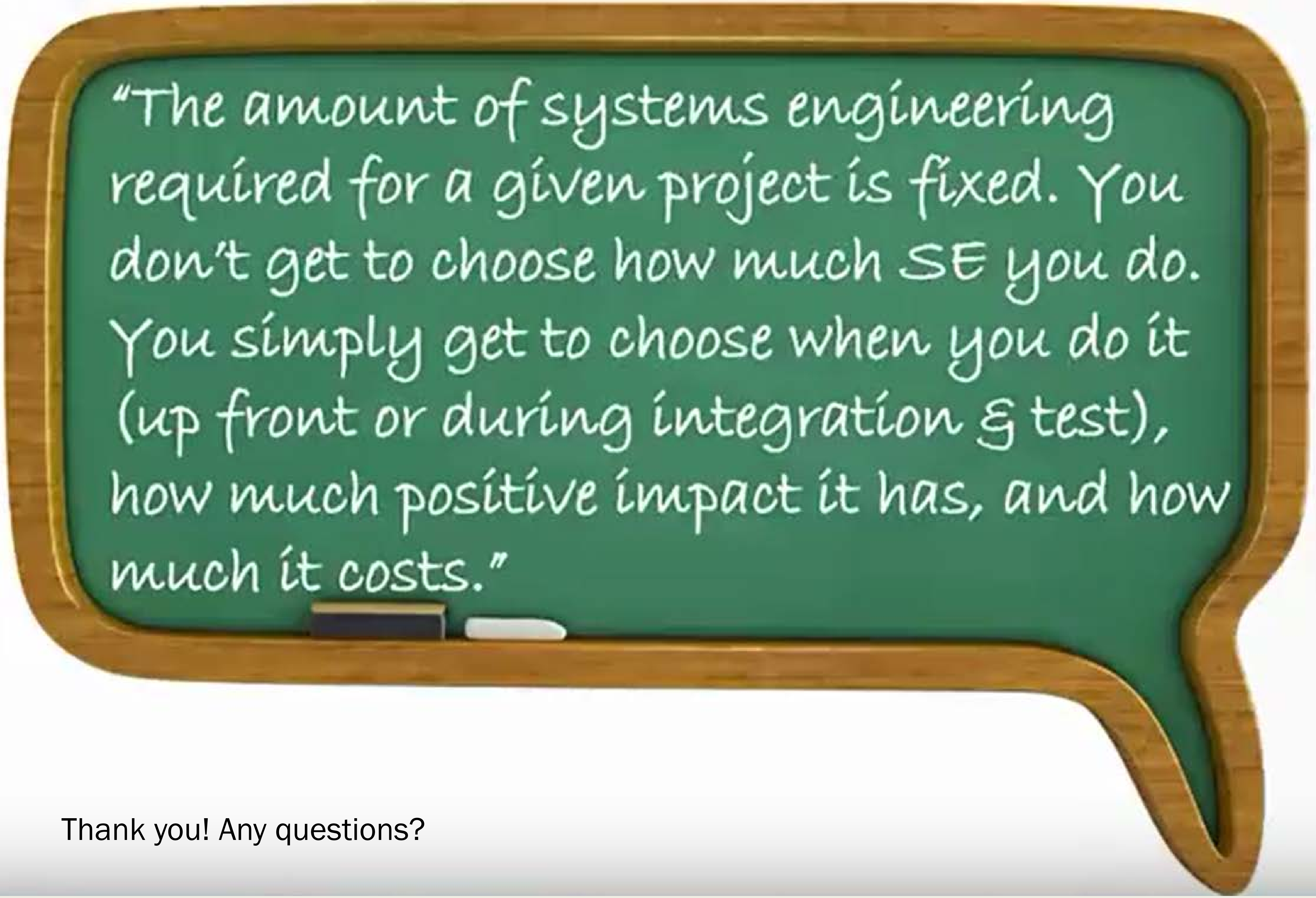
- SPADE
  - *Stakeholders*
  - *Problem formulation*
  - *Analysis, Alternatives*
  - *Decision-making*
  - *Evaluation*



Essential SE framework

- SPADE
  - *S = Who, why*
  - *P = What*
  - *A = How well?*
  - *D = How*
  - *Evaluation – are we there yet?*





Thank you! Any questions?