Engineering 180
Systems Engineering

Professor Kenneth Kung

kennethkung@ucla.edu

(Cell) 562-387-3493

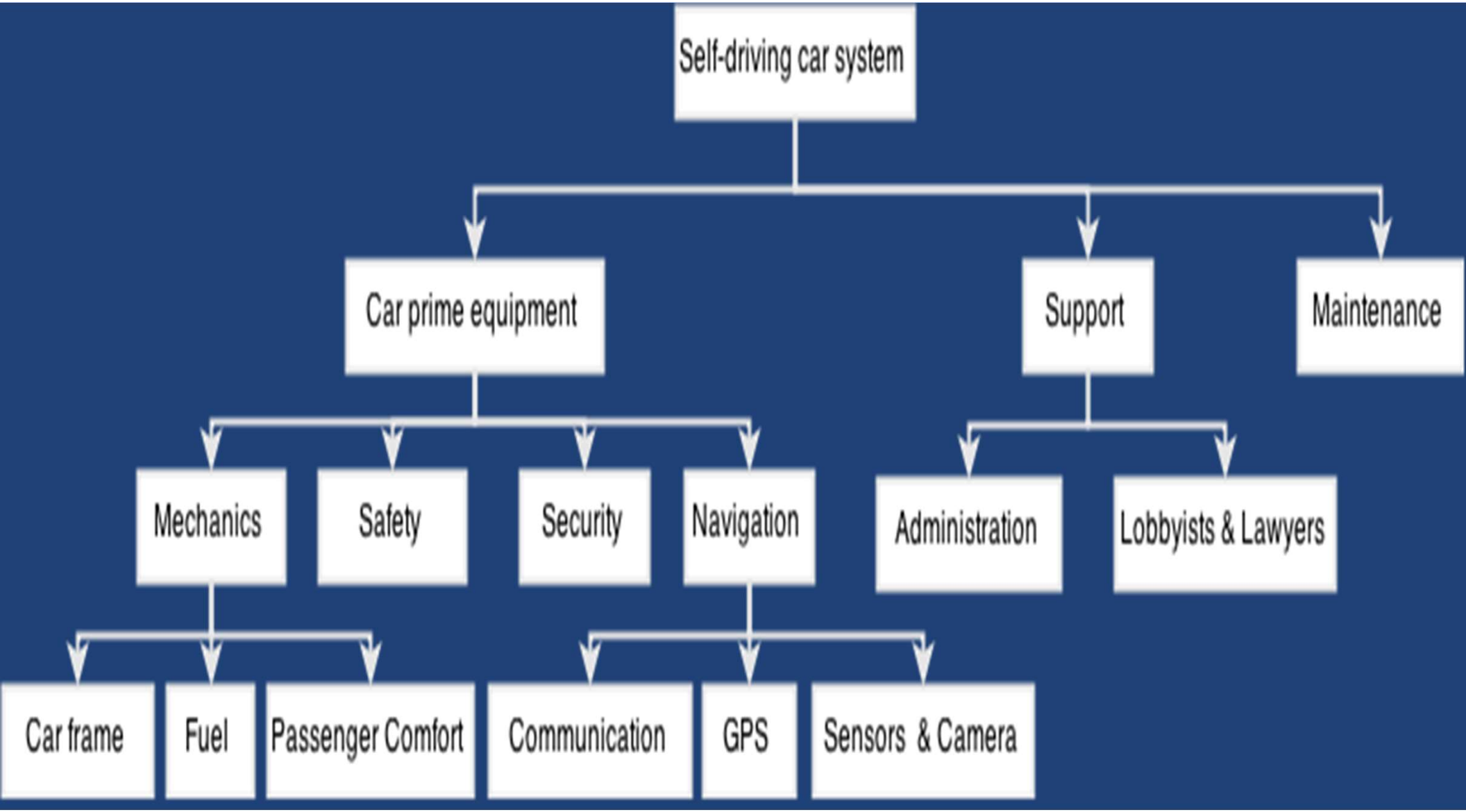
Introduction to Systems Engineering

Complex Systems Have Many Layers Of Subsystems

- How to identify these subsystems?
- How to keep track of the inter-relationships of these subsystems (interfaces)?
- How to make sure these system work together and satisfy the system objectives?
- During the system life cycle, subsystems will go through changes (product evolutions, parts replacements, …)
  - How to make sure these subsystems will continue to work together?
- How to deal with necessary system upgrades to meet future requirements?

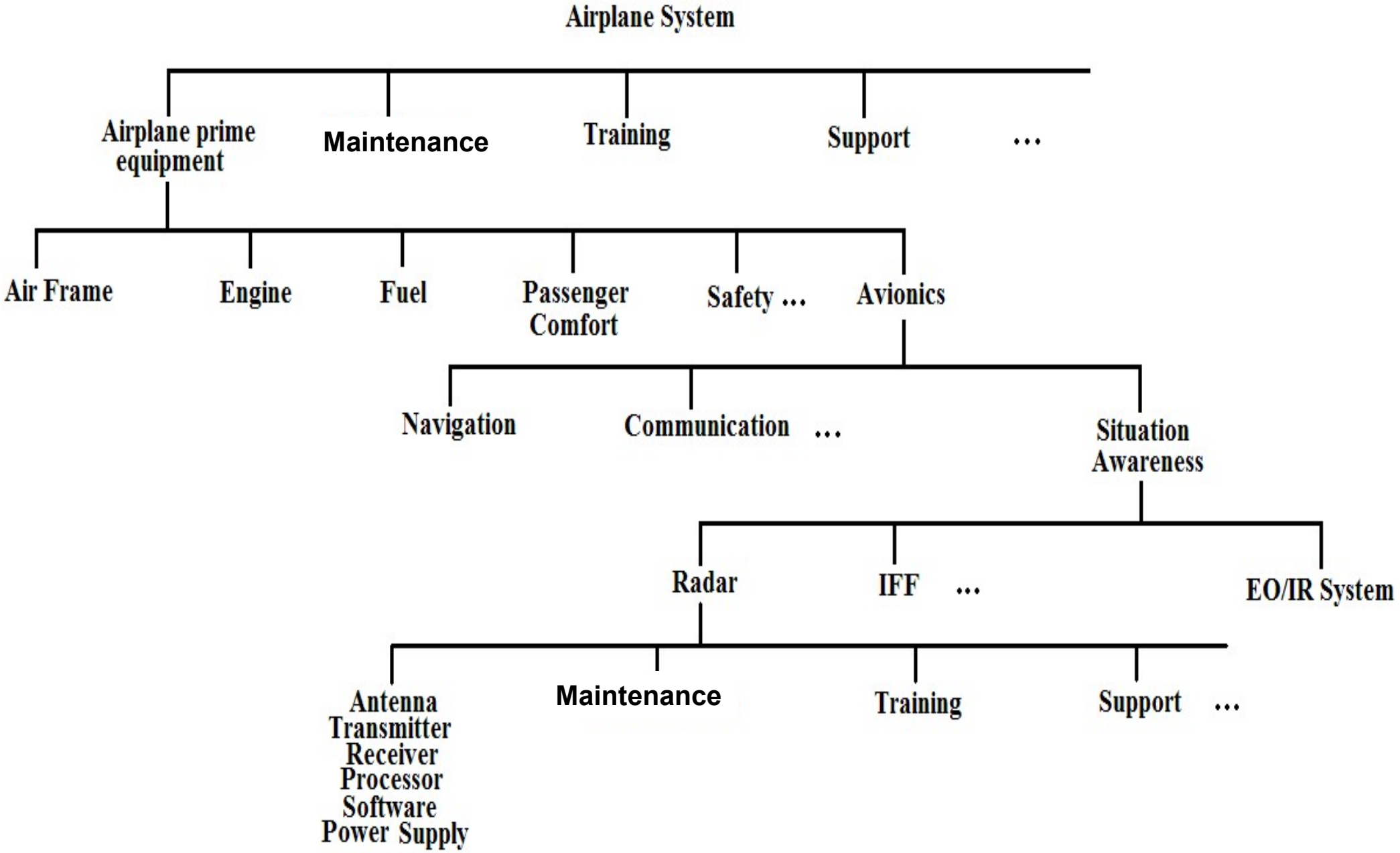
System Engineering processes, principles and tools are designed to solve these difficult problems

Example: Self-Driving Car System



Source: Project Otto, 2015

Example: An Airplane System



What are the resources for this aircraft example?

- Personnel – aircrew, ground crew
- Material – fuel, lubricants, tires, spares
- Facilities – terminals, maintenance & repair
- Data – maintain and operate (repair drawing, user manuals and instructions)
- Hardware – aircraft
- Software – engine management, navigation, communication, flight control, etc.

Use Zachman Framework to Organize

We will be using the Zachman Framework to explain some of the concepts

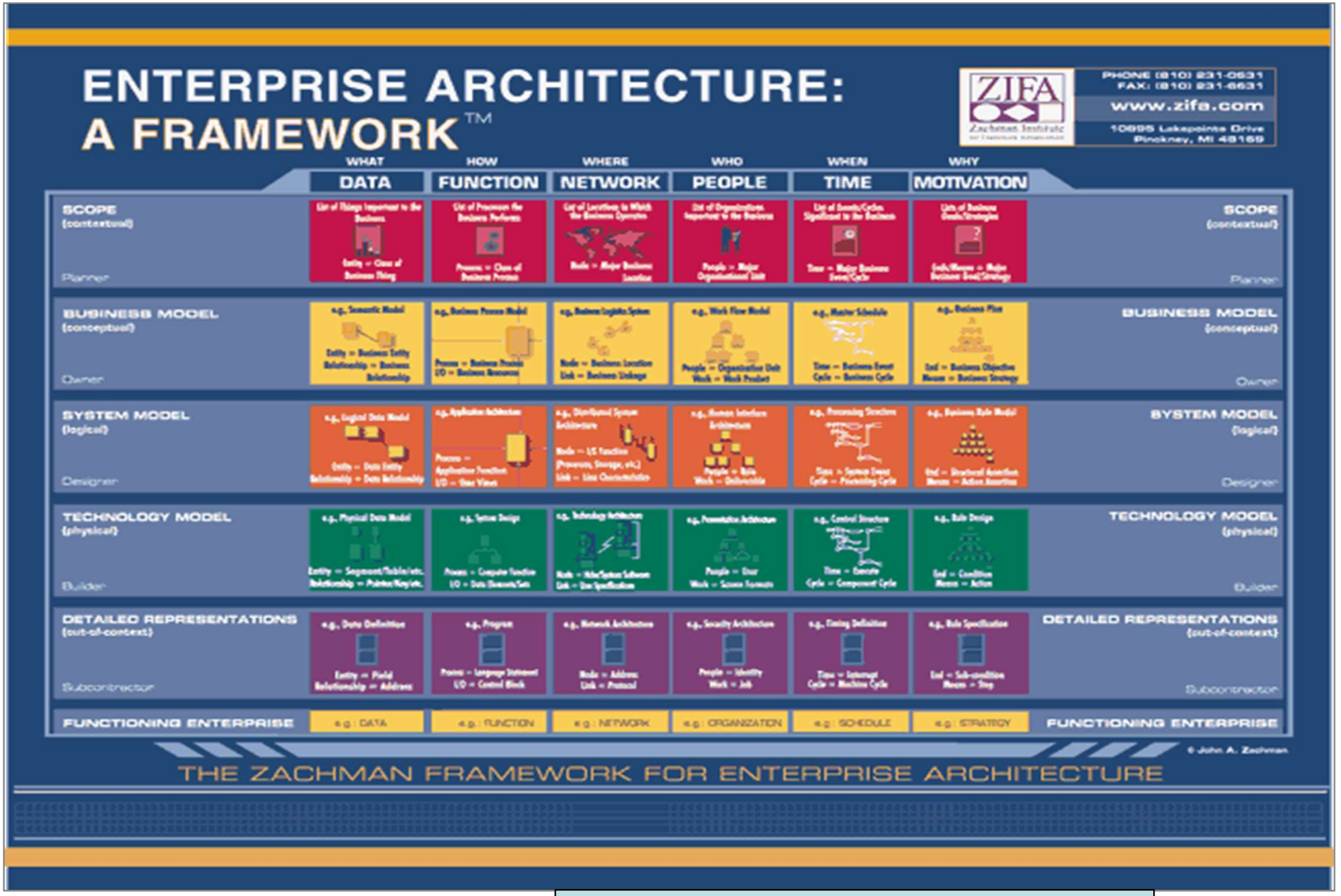
- A brief history
  - Industrial Age – architecting more complex products
  - Information Age – architecting more complex enterprises
  - Framework for Enterprise Architecture
- What is this framework
  - Six communication interrogatives – columns
  - Transformation of an abstract idea into instantiation in six steps – rows

Zachman Framework for Enterprise Architecture

This one is for more technical users

UCLA

SYSTEM ENGINEERING



Used with permission

Example of Applying Zachman

Let us do this together

Apply to (pick a system)

Hospital
Given COVID-19 situation

Pick the admission of the hospital operation

Start with Column 6

- Start with Needs and Visions, produce Goals

	What	How	Who	Where	When	Why
Scope						
Business Model						
System Model						
Technology Model						
Detailed Representation						
Functional Enterprise						

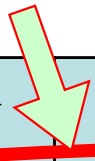
Develop the Functions to support the Goals

	What	How	Who	Where	When	Why
Scope						
Business Model						
System Model						
Technology Model						
Detailed Representation						
Functional Enterprise						

Determine the Data, Place, People and Time to Support Functions

	What	How	Who	Where	When	Why
Scope						
Business Model						
System Model						
Technology Model						
Detailed Representation						
Functional Enterprise						

Analyze cross cells to determine whether we missed any information



	What	How	Who	Where	When	Why
Scope						
Business Model						
System Model						
Technology Model						
Detailed Representation						
Functional Enterprise						

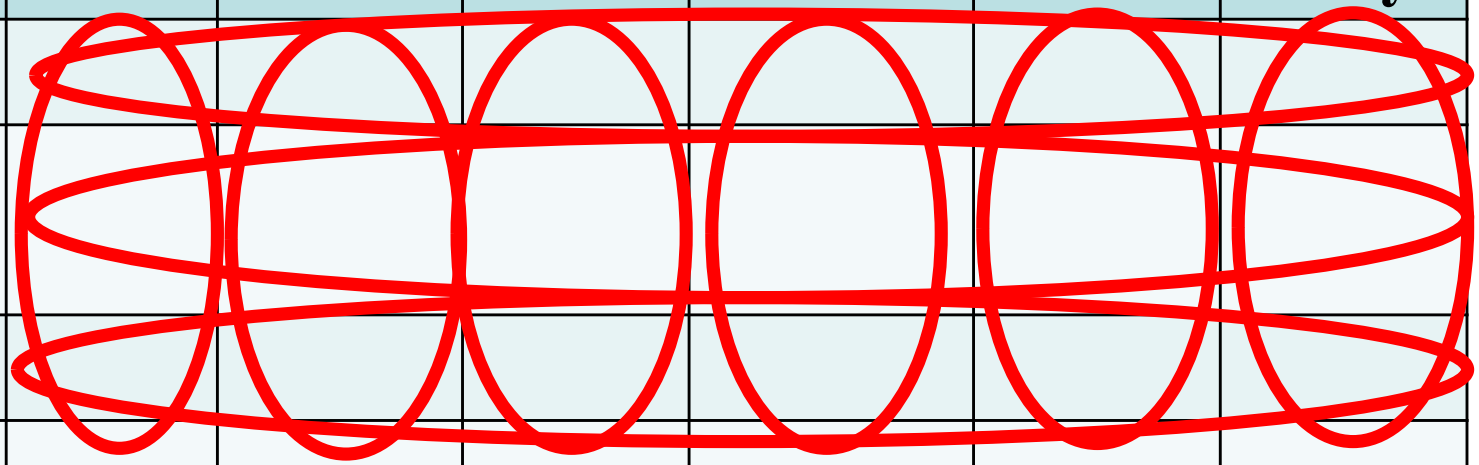
Develop Row 2 and 3 for Business Models

- Identified many of these items while developing Row 1

	What	How	Who	Where	When	Why
Scope						
Business Model						
System Model						
Technology Model						
Detailed Representation						
Functional Enterprise						

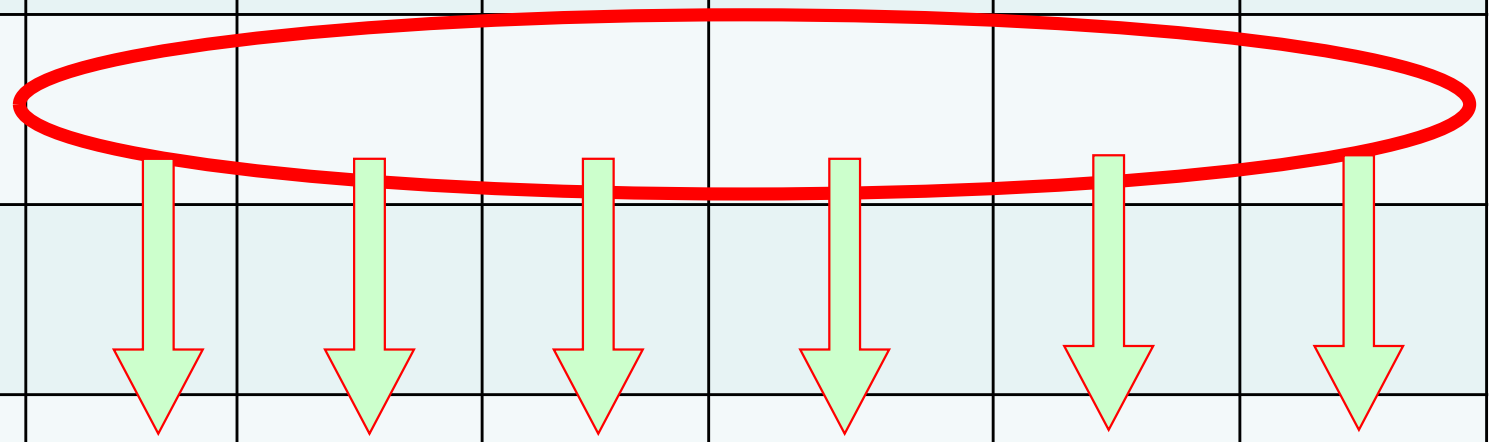
Harmonize vertically and horizontally to make sure the whole framework still makes sense

	What	How	Who	Where	When	Why
Scope						
Business Model						
System Model						
Technology Model						
Detailed Representation						
Functional Enterprise						



Next we look into reference architecture to determine any technology models can be re-used

	What	How	Who	Where	When	Why
Scope						
Business Model						
System Model						
Technology Model						
Detailed Representation						
Functional Enterprise						



Reuse as many models as possible

Week 2 Homework 1

- Select a potential system that you might want to do the conceptual design for your group project
- If you don't have any system in mind yet, then consider
  - UCLA registration system, Westwood traffic flow system (broader than traffic signals), Your automobile system, Student Healthcare "system", …
- Analyze row 1 & 2 of the Zachman Framework
  - Consider each column
  - If any column information is uncertain, then skip it
- Create a spreadsheet just like the example we just did

Group Discussion Exercise

- In your group, select a system that all of you may know
- Analyze the Zachman Framework Row 1 and 2
  - Describe what information should be included in each model (specific column and row)
- Be prepared to outbrief your discussion
  - Appoint one or more students as your team spokesperson(s)

Plan to get back in about 15-20 minutes

In Class Group Exercise

- Get together with your team
- Select a potential system that your group might be considering to do the conceptual design
- If you don't have any system in mind yet, then consider
  - UCLA registration system, Westwood traffic flow system (broader than traffic signals), Your automobile system, Student Healthcare "system", …
- Analyze row 1 & 2 of the Zachman Framework
  - Consider each column
  - If any column information is uncertain at this point, your team should investigate later

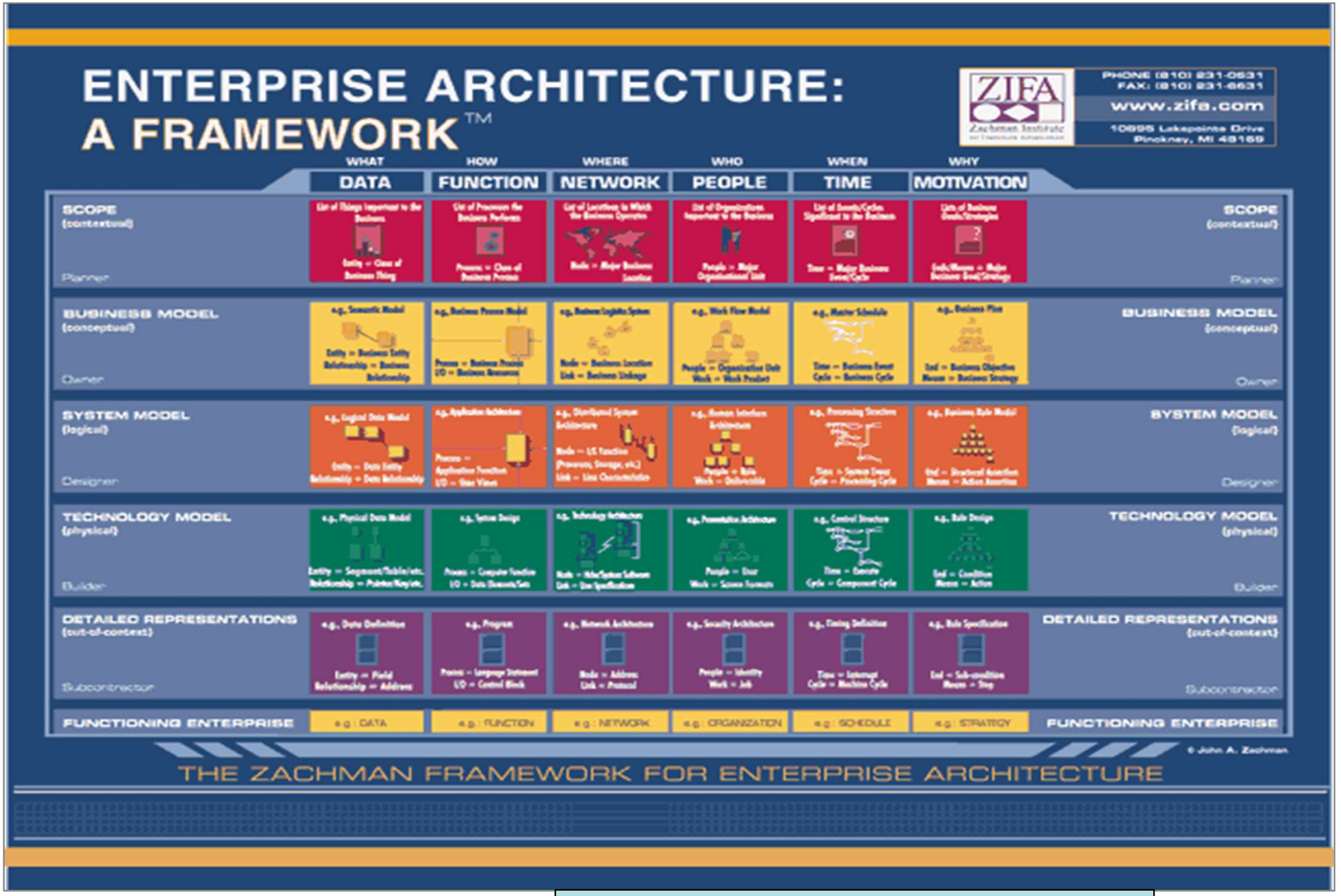
Breakout Session Considerations

- Who are the stakeholders?
- Time Frame the system is needed – When
- Why is it needed?
- How would the system work?
  - Its Capabilities
- Where would the system operate?
  - Is a communication network needed?
- What data does it need and would generate?

It is possible some of them not applicable

Zachman Framework for Enterprise Architecture

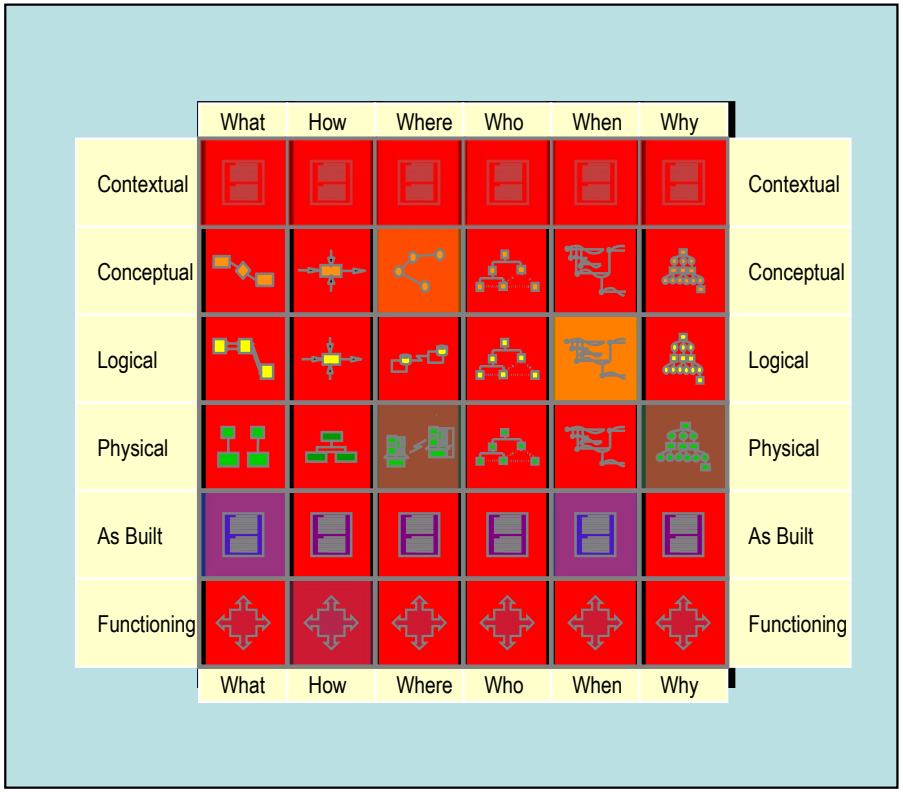
This one is for more technical users



Used with permission

Zachman Framework: Framework Rules*

- The columns have no order
- Each column has a simple, basic model
- The basic model of each column must be unique
- Each row represents a distinct, unique perspective
- Each cell is unique
- The composite or integration of all cell models in one row constitutes a complete model from the perspective of that row
- The logic is recursive



*John Zachman, J.F. Sowa; *Extending and Formalizing the Framework for Information Systems Architecture*, IBM Systems Journal, Vol. 31, No. 3 (1992); IBM Publication G321-5488

Key Design Considerations

- Complete solutions
- Life cycle focus
- System of system environment

Providing Complete Solutions

- Functionality and performance
- Weight, size, power consumption
- Ease of use – user interface, ease of installations and integrations
- Training
- Reliability, maintainability, and supportability
- Safety and Security
- Cost : Development cost, procurement cost, cost of ownership, …
- Schedule : development schedule, production schedule

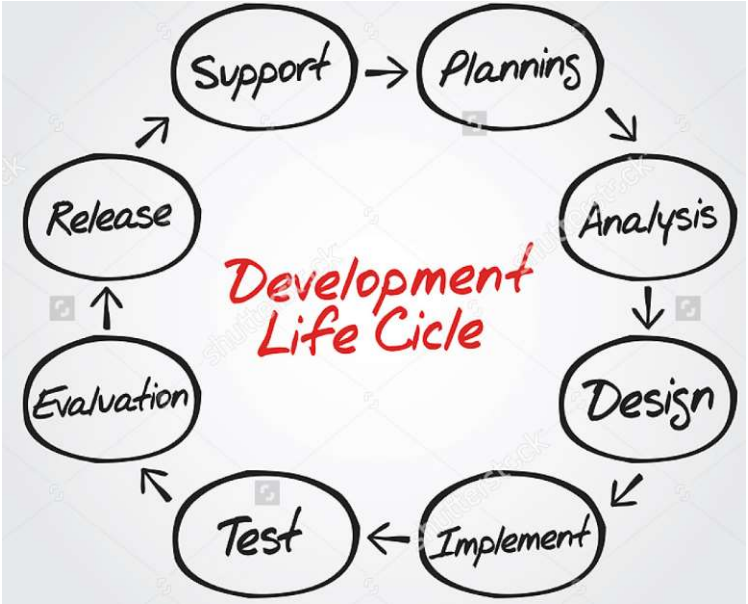
Example for Complete Solution: Smart Phones

- Functionality: Mobile phone, Short messages, e-mails, music, Internet access, …
- Light weight, small (low profile), battery life
- Ease of use: try iPhone
- Training: Not needed
- Reliability, maintainability: product warrantee, short life cycle (2-4 years)
- Safety and Security: Microwave energy safety, data security
- Cost: Must be cost competitive
- Schedule: Timing of product release are critical.

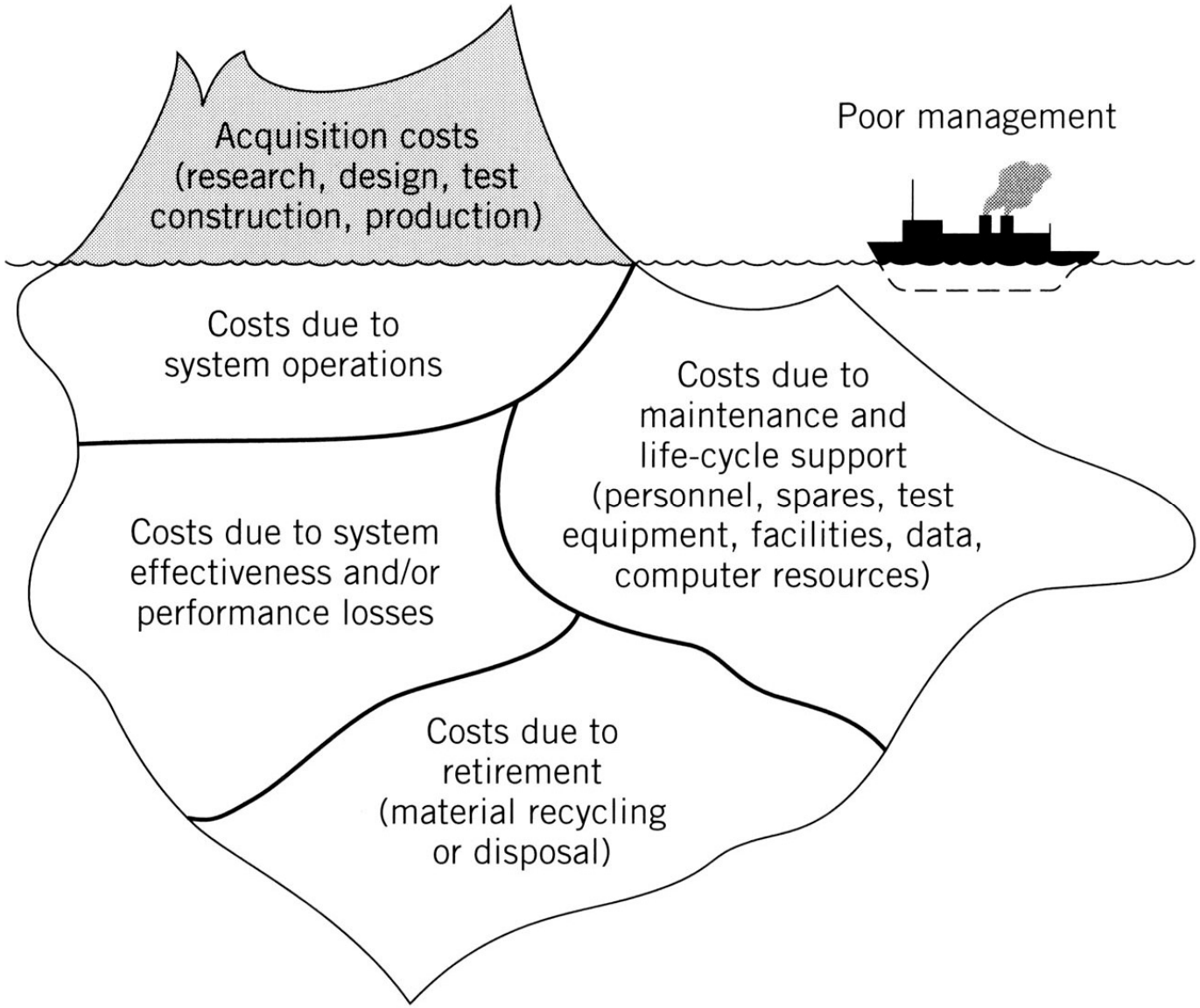
Life Cycle Focus

- Life Cycle cost
- Design to accommodate future system evolutions such as requirement changes, technology and parts obsolescence, system migrations.
  - Ease of modifications and upgrades
  - Ease of re-integrations and regression tests

➢ Plan for continuous improvements on supportability and sustainability
  - Increased reliability
  - Improved maintainability
  - Reduced logistics footprint

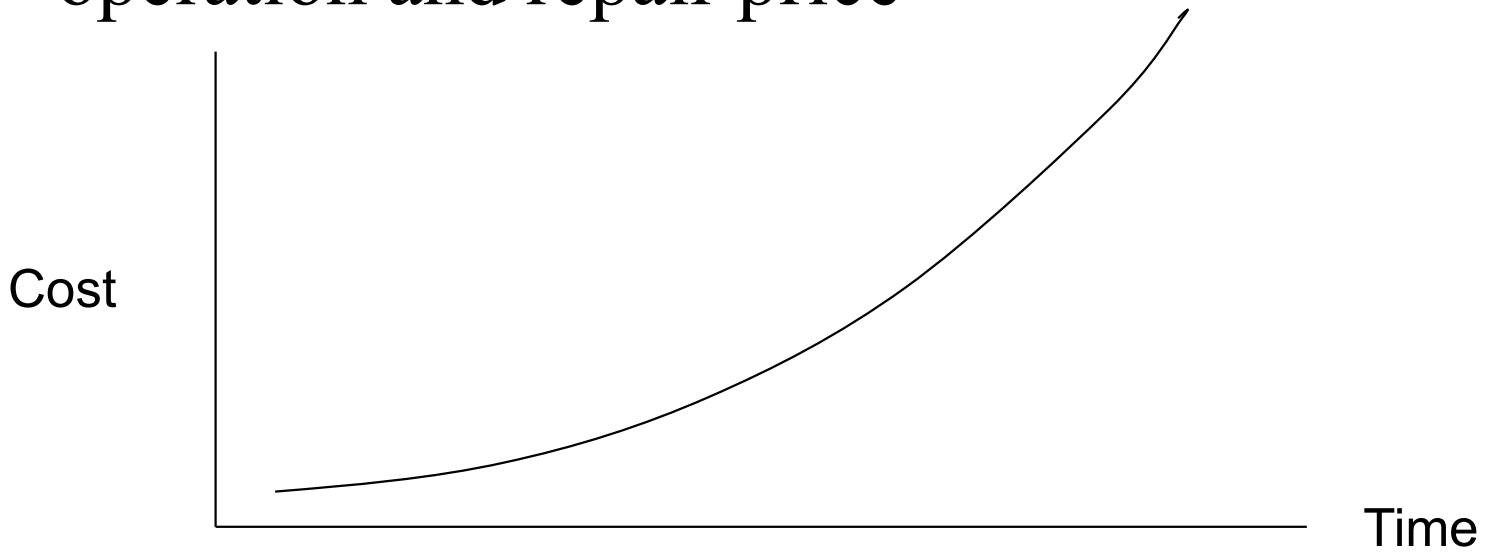


Total Cost Visibility



Total Cost in Utilization Phase

- It is more than developing a system
- Most of cost incurred during the operation and maintenance phase
  - Automobile example – purchase price versus operation and repair price

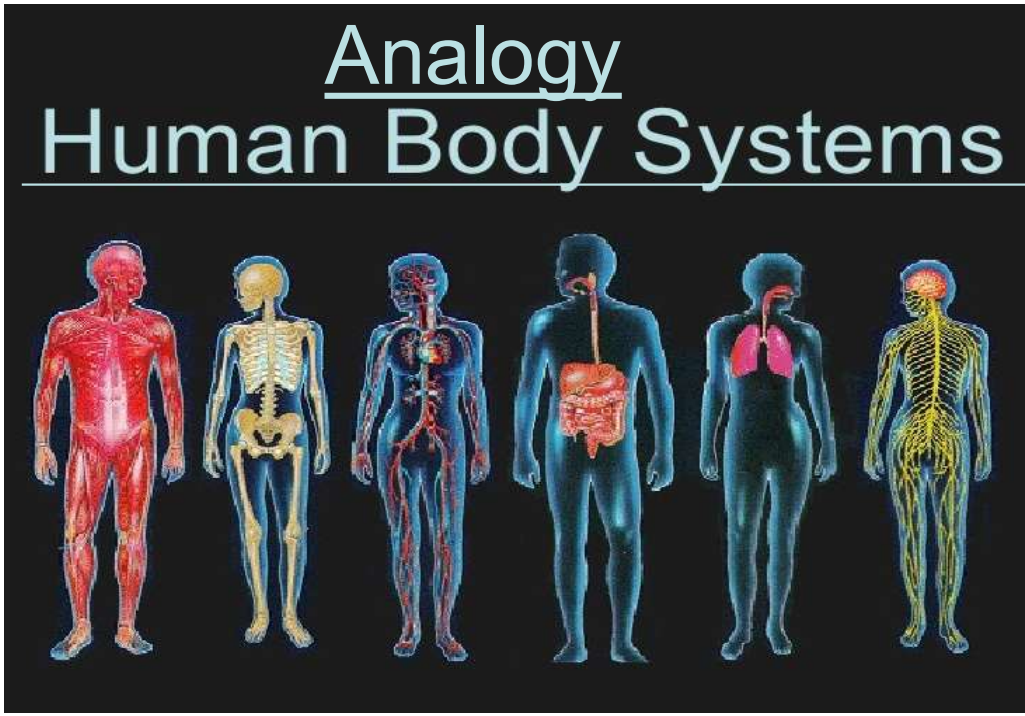


System of Systems Environment

- Almost all systems will be deployed as part of a large System of Systems (SOS)
  - Role of each individual system in the delivery of user capabilities
  - Demands of the SOS incorporated into the system requirements and development process

Check List for System of Systems (SOS)

- SOS context: Are there special required capabilities to become part of the existing or planned SOS?
- Design for integration: Is the system architecture open and adaptable to enable reconfiguration and integration into a SOS?
- Interface: Have the SOS interface requirements been adequately defined?

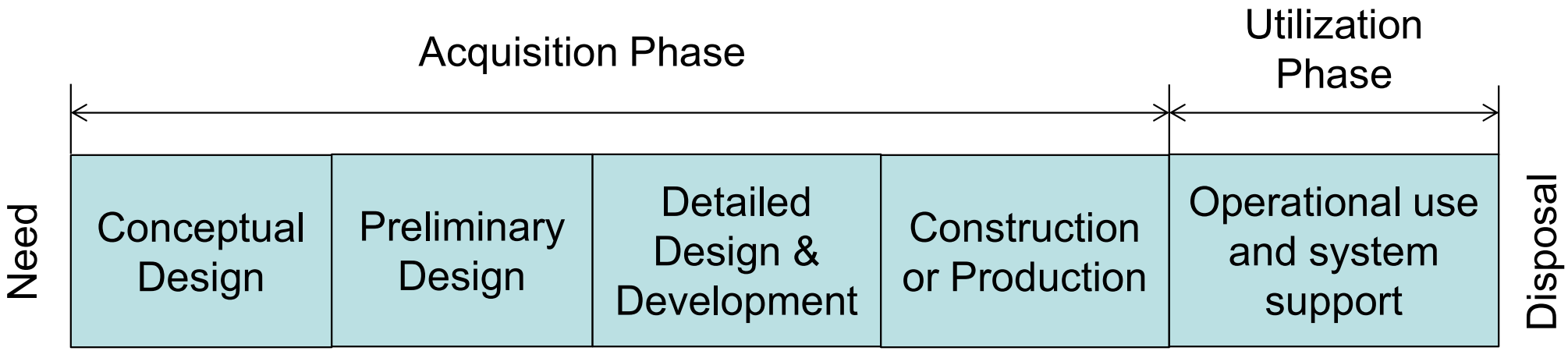


In Class Group Exercise – take 15 minutes

- For the system that you did for the Zachman analysis – select one of your team member input
  - Determine what should be considered for a "Complete Solution"
  - What might be the "life cycle focus"?
  - Any system of systems consideration?

System Life Cycle

- MIL-STD-499B



A draft standard released in mid 1990's.

MIL STD 499B

The Phases

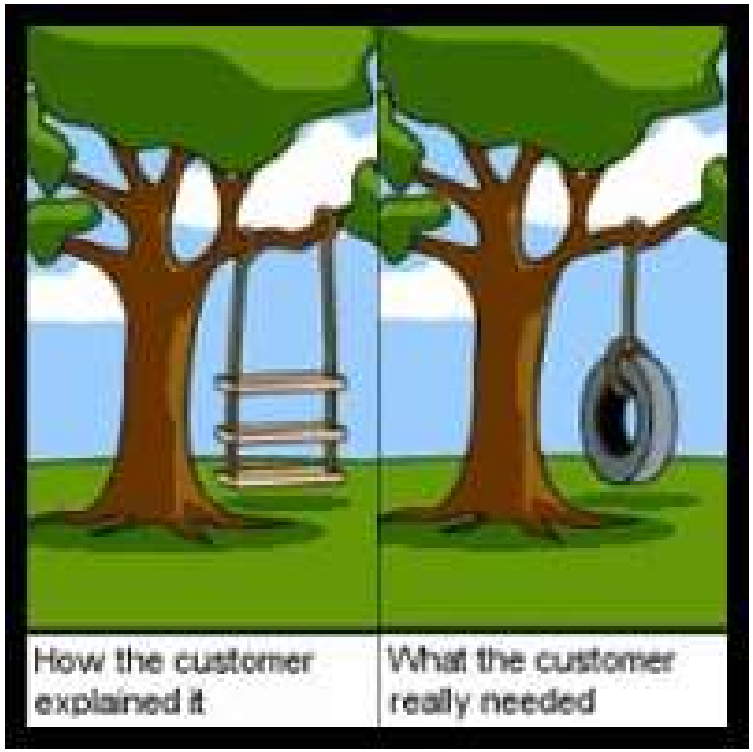
- Need
  - Wish list from users
  - Perceived customer wish list by marketing
- Conceptual Design
  - Functional Requirements well defined problem statement
  - **Functional baseline**
    - Describe the whats and whys of the system
    - Remove the ambiguity
- Preliminary Design
  - Convert the functional baseline into a preliminary definition of the system configuration or architecture – Hows of the system
  - **Allocated baseline**

The Phases (cont)

- Detailed Design and Development
  - Design the details
  - **Product baseline** – System defined by products
  - Some prototyping needed to confirm fitness
- Construction and Production
  - System components produced
  - Formal test and evaluation
- Utilization and Operation
  - Maintenance
  - Modification
- Disposal
  - Activities at the end of system life – could be costly

Requirements Engineering

- Simple statement of need translated into a number of statement of requirements
- Form the basis for functional design and architecture
- Forward traceability
  - System level requirements to detailed design
- Backward traceability
  - Individual design justified by at least one higher level requirement

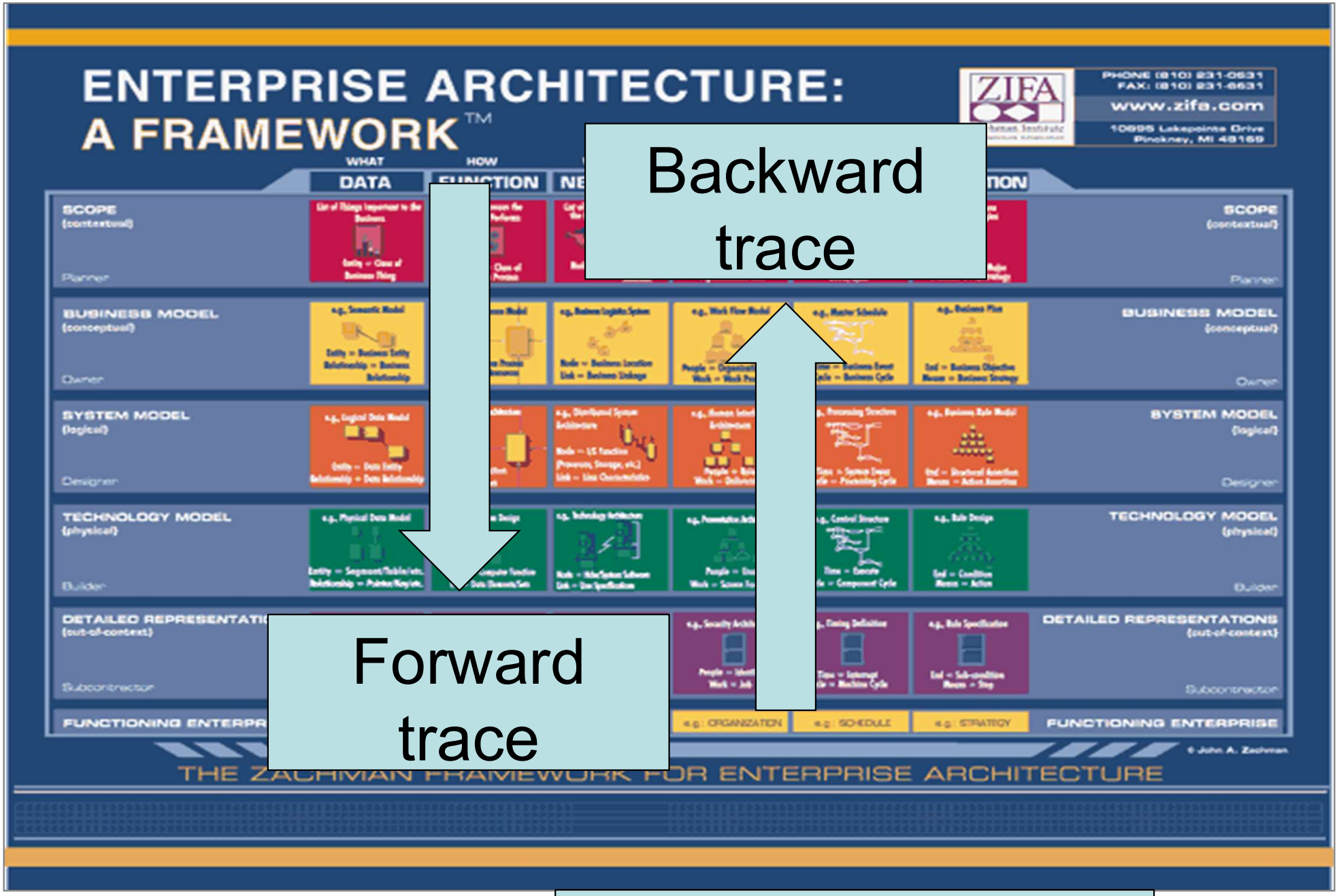


Zachman Framework for Enterprise Architecture

This one is for more technical users

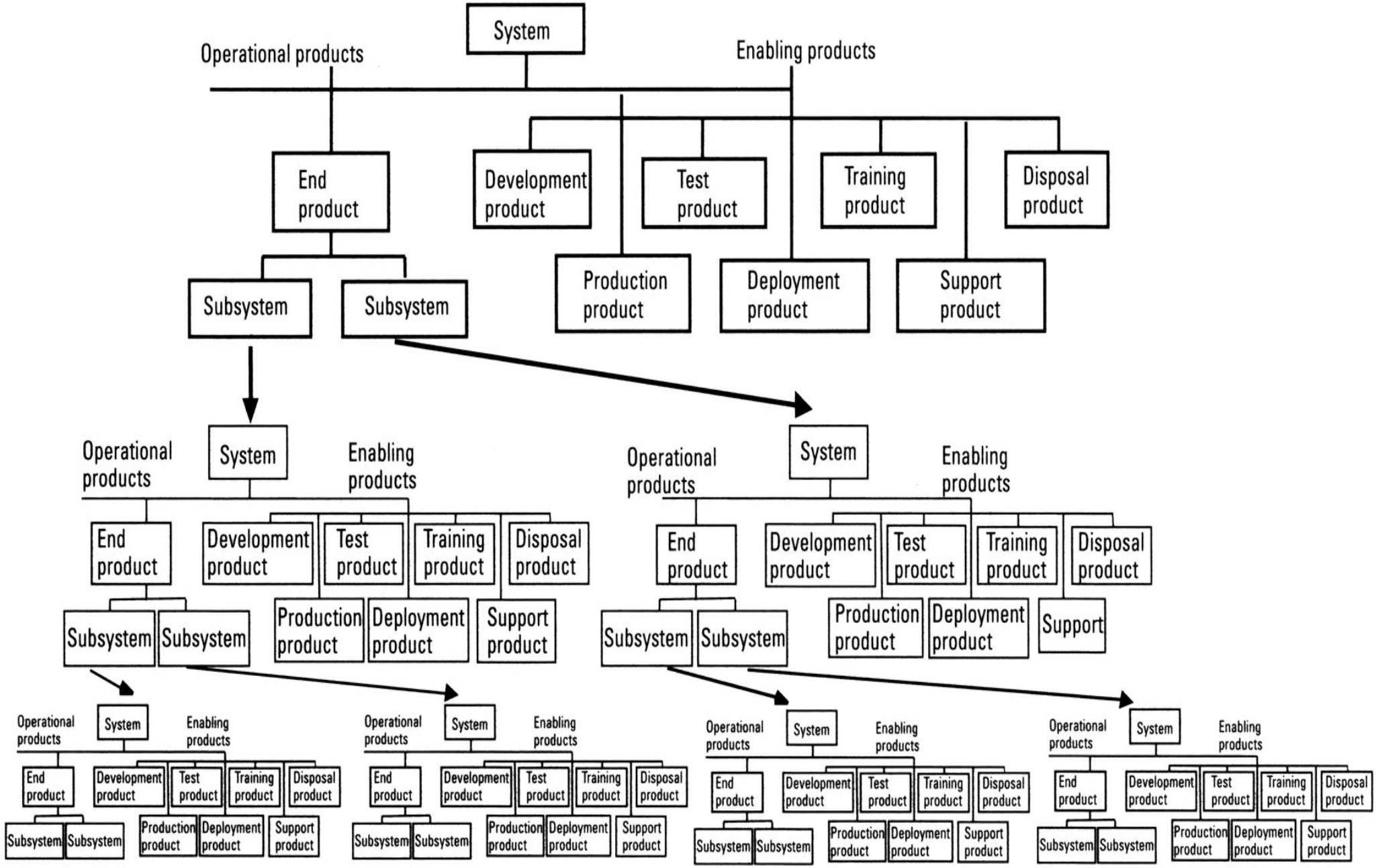
UCLA

SYSTEM ENGINEERING

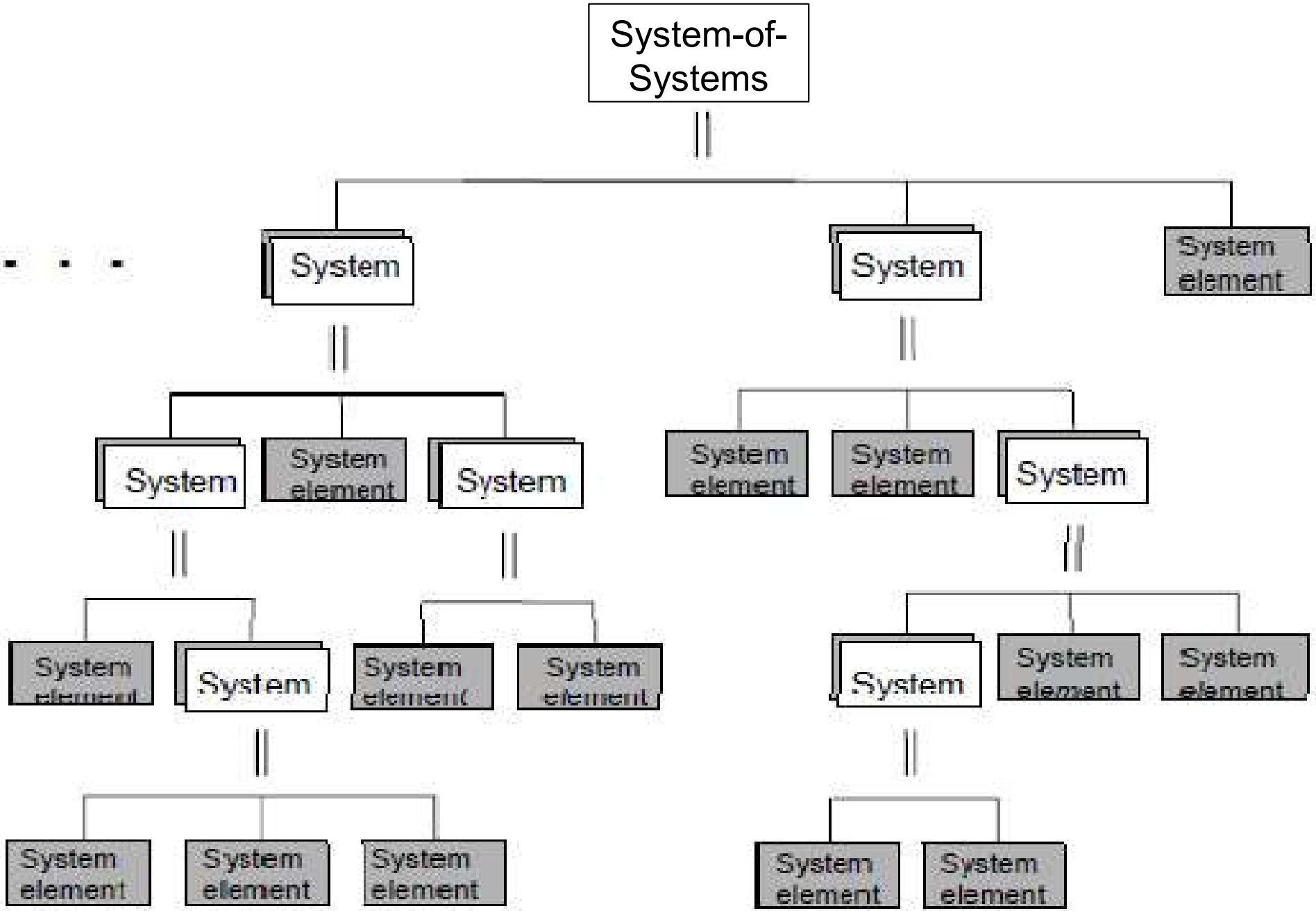


Used with permission

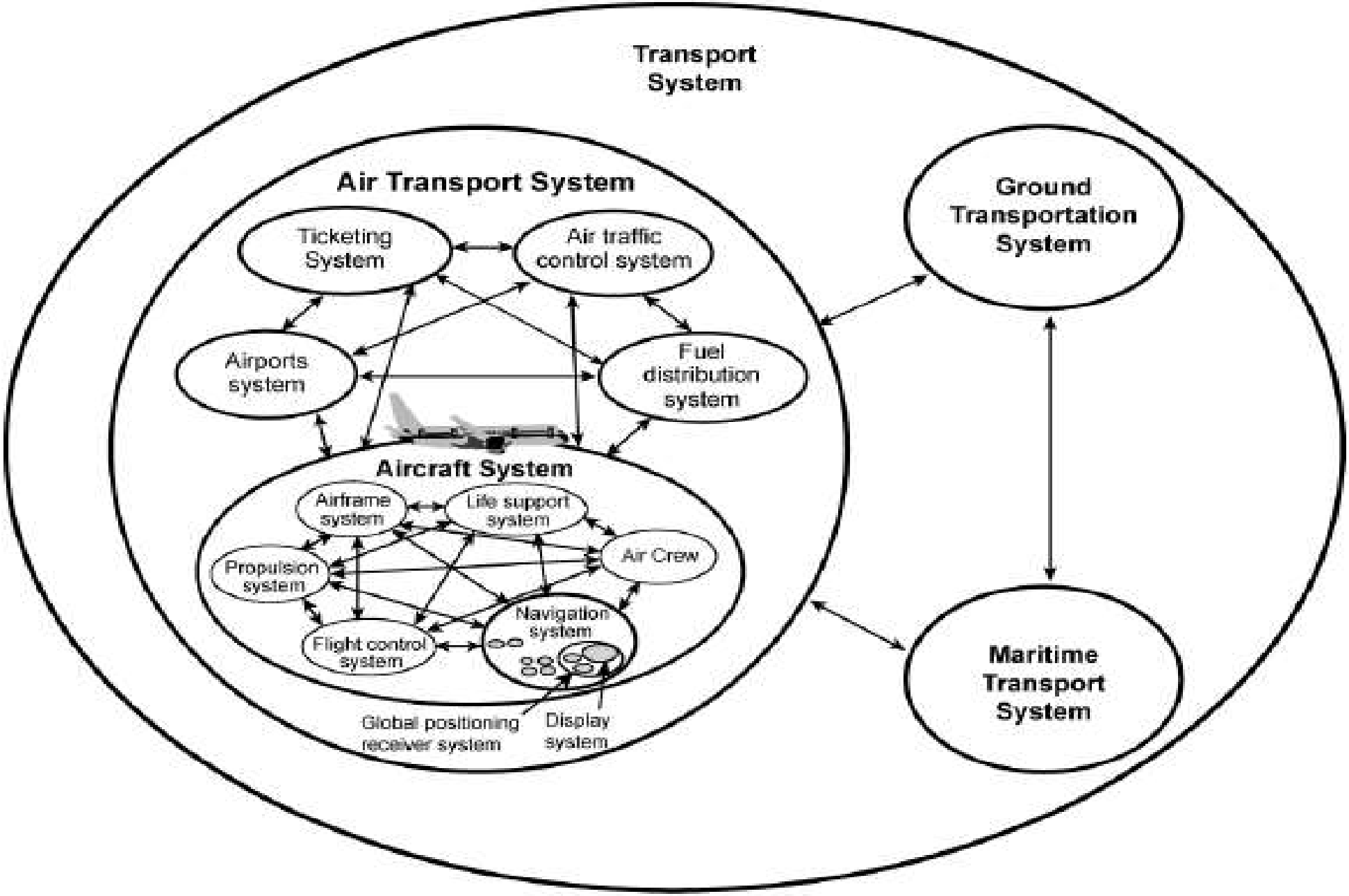
The complexity of systems



Hierarchy Within a System



Example of System of Systems – Air Transportation System

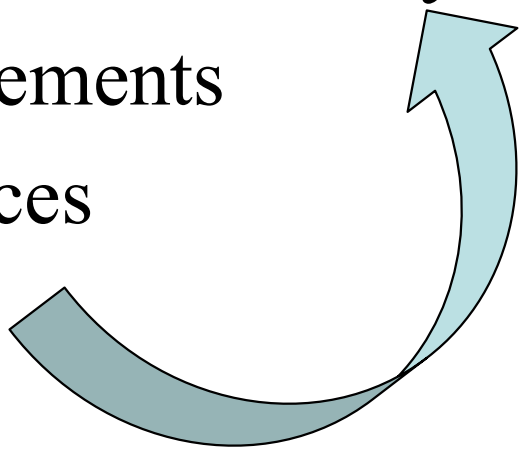


In Class Group Exercise – take 15 minutes

- Select a system
  - Use the system you selected for the previous group exercise
  - If that system is not suitable for this exercise, then use a smart phone system
- Write down around seven to ten "sub-systems" within your "system"
  - What does each sub-system do?

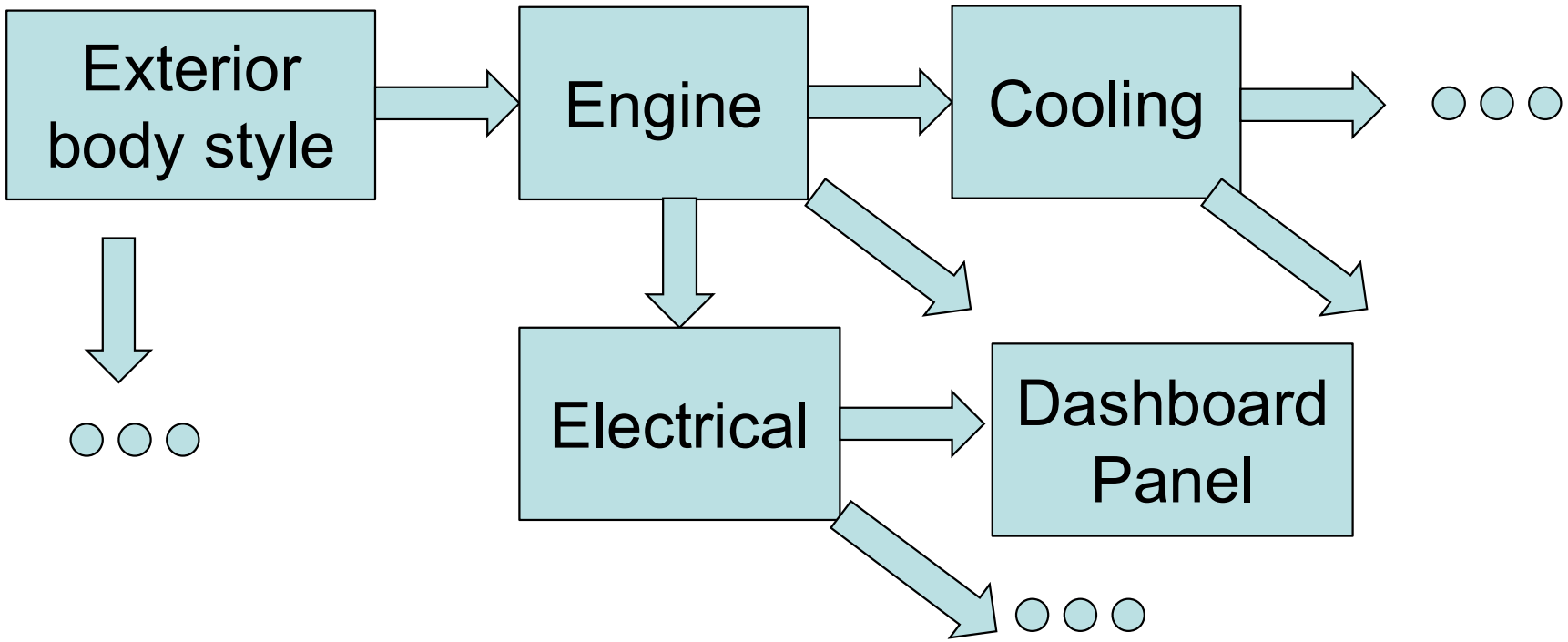
Top-Down Development

- Understand the whole system first
  - Requirements
  - Interfaces
- Break down to subsystems
  - Requirements
  - Interfaces
  - Repeat



Bottom-up Approach

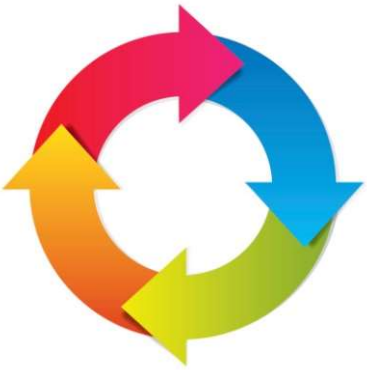
- Example:
  - Building automobiles – the old way
    - before GM bought Hughes



Weight, space, fuel efficiency, …

Need Both Top-down and Bottom-up Approaches

- Bottom-up approach – works well on simpler systems
  - Traditional engineering design method
  - Known components are assembled into a system, and then tested
  - Iterate this process until the system meets the desired criteria
- Top-down approach – necessary for more complex systems
  - Understand the system level requirements (through analysis)
  - Break down into subsystems and then into components
  - Flow down the requirements from system level to subsystems and then to components
  - Build and integrate the system using bottom-up approach
- Often the designs are the results of several iterations of both top-down and bottom-up approaches

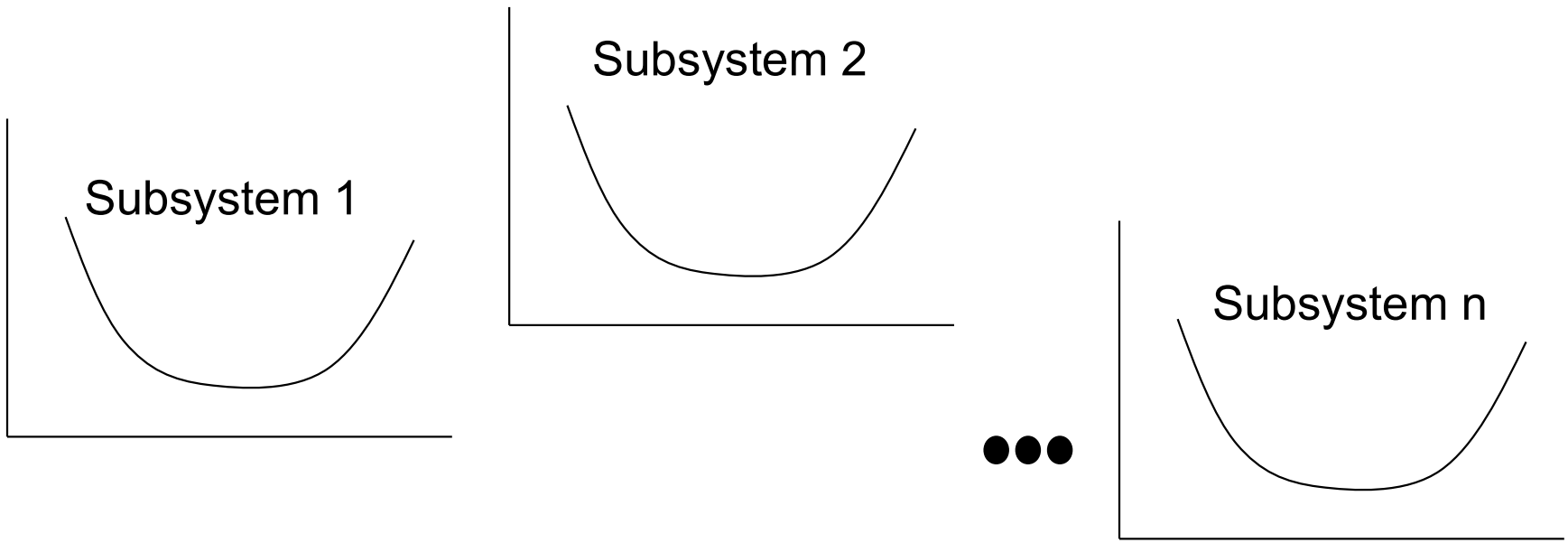


System Optimization

- Combine many optimized subsystems

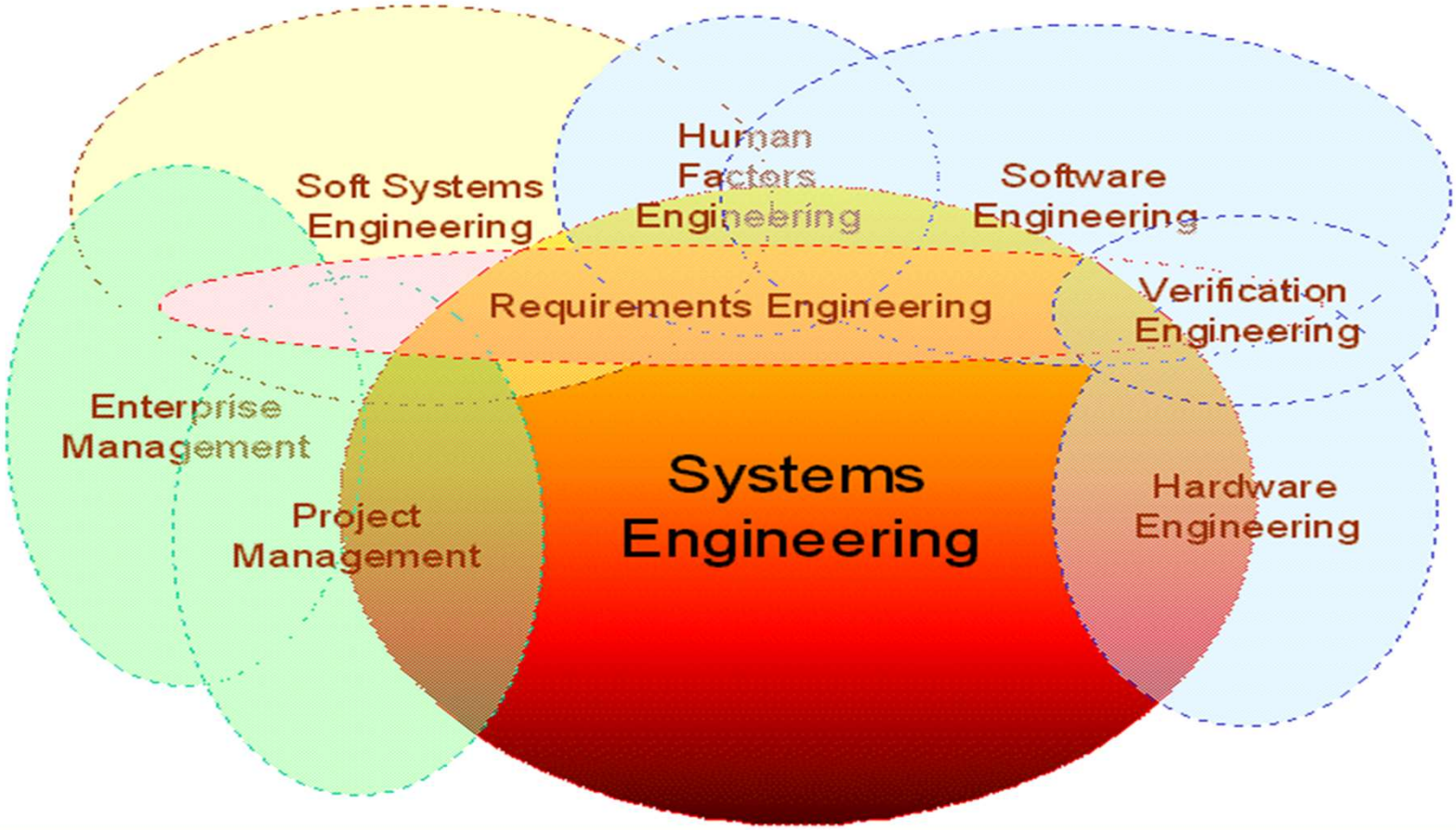
$\neq$

Optimized whole system

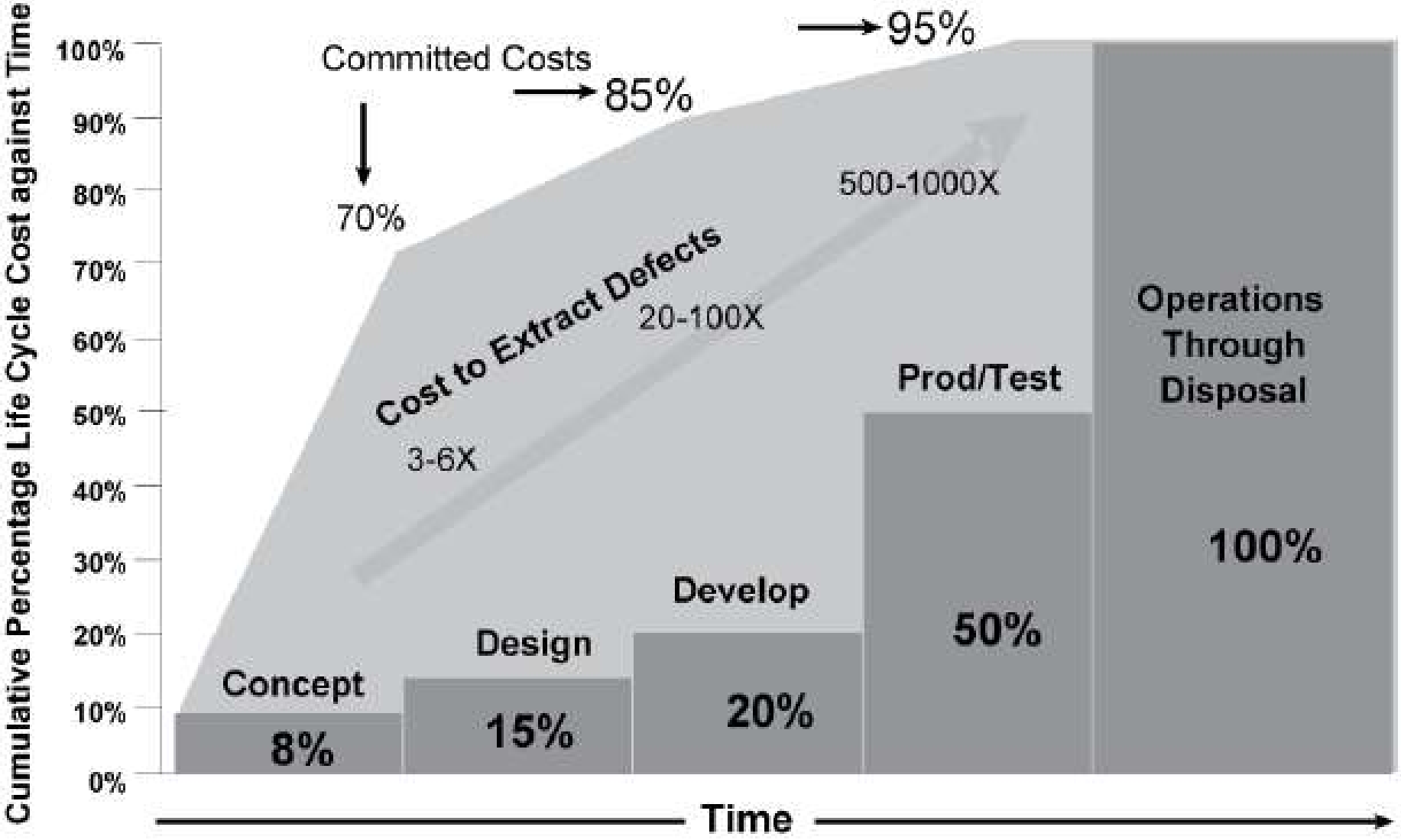


Integration of Disciplines and Specialties

- System Engineering manages and integrates multitude of technical and non technical disciplines and specialties
- Breakout the tasks so each can do their design



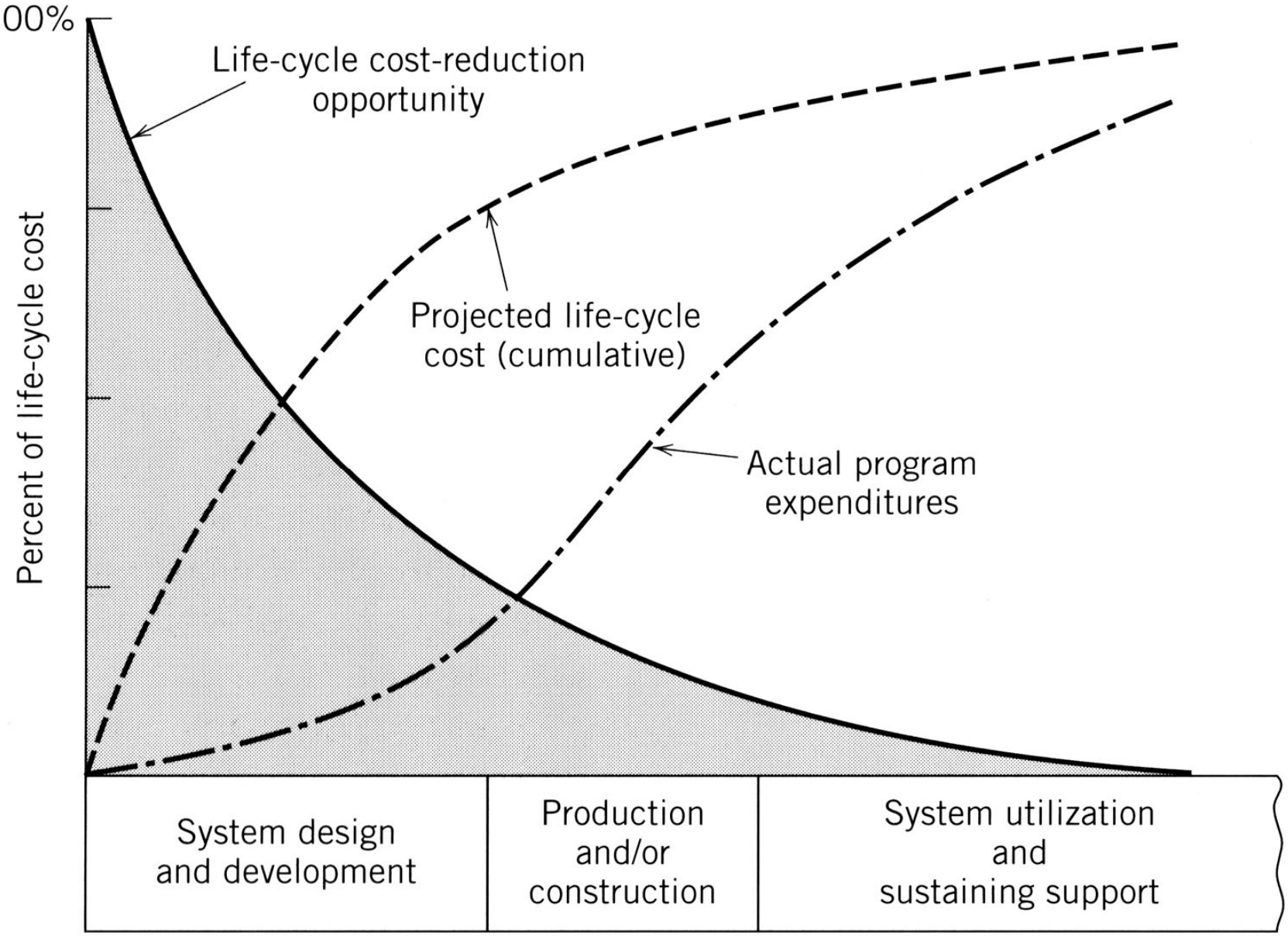
Life Cycle Cost Against Time



Source: INCOSE Systems Engineering Handbook

Systems Engineering Benefits

- Cost Impact at early phases

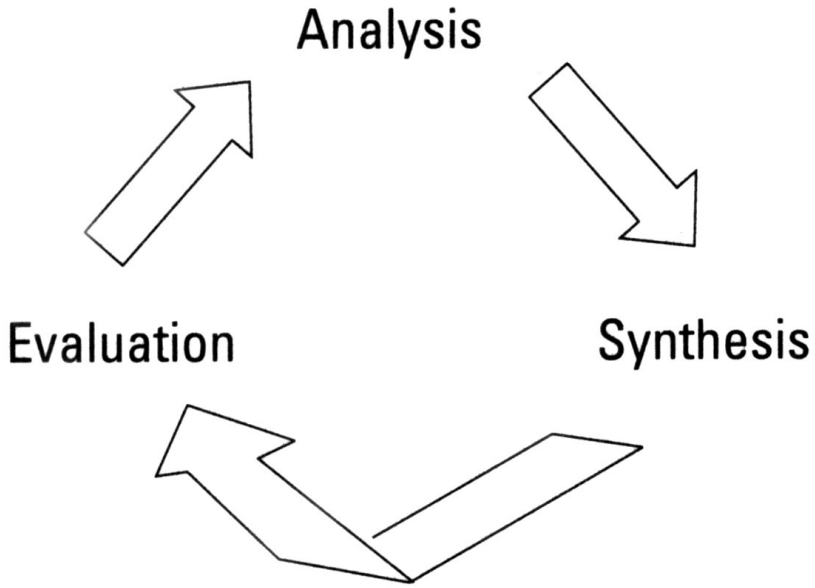


Basic System Engineering Tools

Analysis – Determine the functions that are needed

Synthesis – Identify (design) solutions

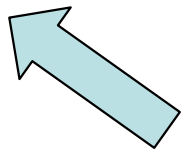
Evaluation – Trade studies on the solutions



- This is an iterated process
- Each step moves to the next level of detail, until the complete system design is finalized

Analysis

- Investigate the requirements
- Identify the essential functions
- Answer what, how well and why questions relative to design
- Define lower level requirements
  - Derived requirements
- Create Functional Flow Block Diagram (FFBD)



More on this in a later lecture

Synthesis

- Determines the "How"
- Produce a design to meet the requirements
- Iteratively
  - Define the functional design
  - Consider different technical approaches
  - Best approach selected then go to the next level of details
  - Repeat until complete system design is finalized

Evaluation

- Investigate and trade alternative designs
- Make decision to meet "all requirements"
  - Identify where the discrepancies are
  - Discrepancy triggers further analysis and synthesis
- Select and confirm the desired approach to design

In Class Breakout Session – 15 to 20 minutes or a Lecture Homework

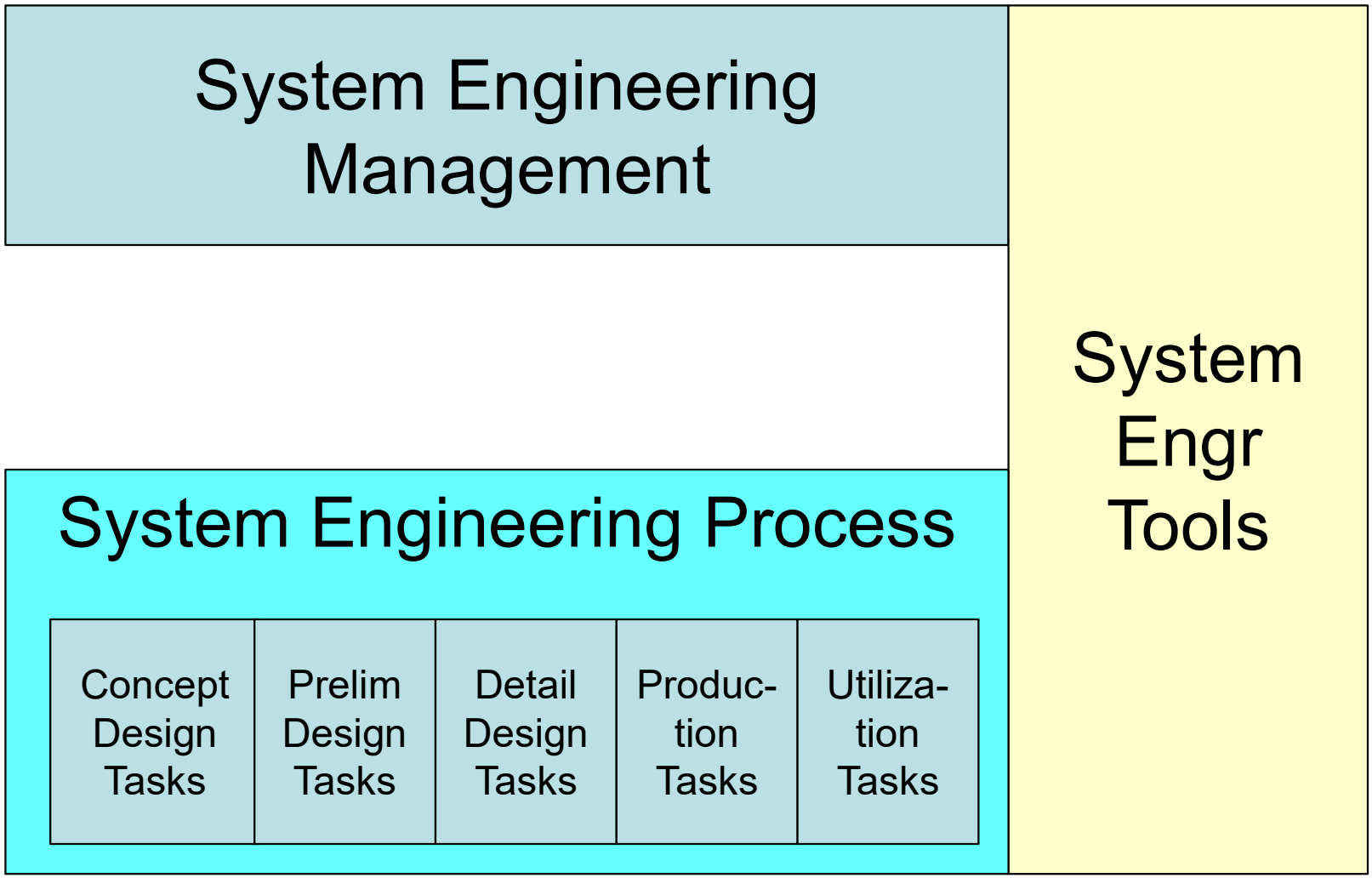
- For the system and subsystems that you analyzed during the lecture
  - State the system and subsystems that your team has discussed
  - Identify one "subsystem" within your system
  - Analyze the next level of "sub-sub-systems"
    - List 3 to 5 sub-sub-systems
    - What does each do?
      - Are there any derived requirements associated with each sub-sub-systems?

Systems Engineering Framework

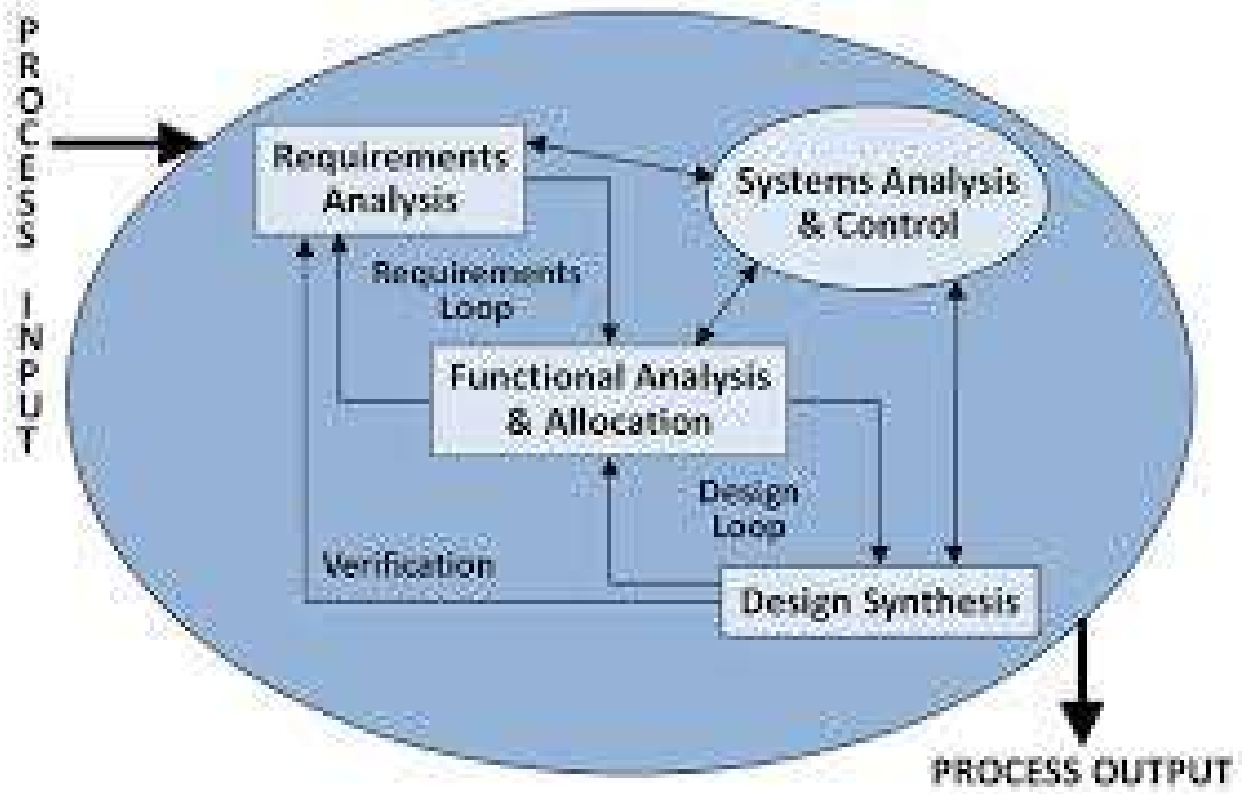
- Relationship of systems engineering process, management and tool
  - Process is the "hows" of systems engineering
  - Management is the directing, monitoring and reporting of the systems engineering effort
  - Process and management support by tools

Examples: Zachman Framework, TOGAF, DODAF, FEAF, ATAM
(quite an alphabet soup)

Related Systems Engineering Disciplines

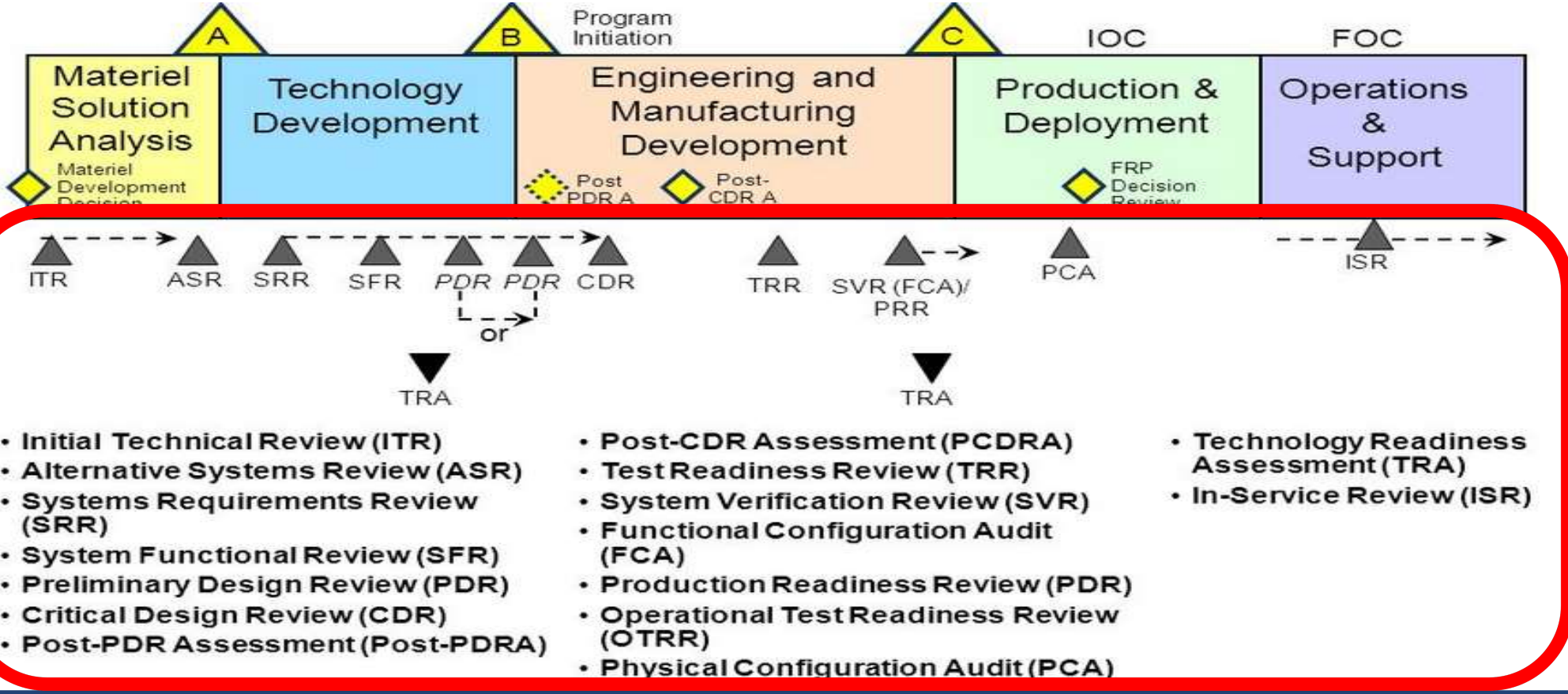


- Apply analysis-synthesis-evaluation iteratively
- This course focus on the acquisition phase
  - The biggest impact to the system
- Look at specific process in the following lectures



Systems Engineering Management

- Cost, Schedule, Technology and Human Resources
- Various reviews during the life cycle
- Go over some of them in the next few lectures for design review



Reviews Are Important – Anecdote

- Mars Climate Orbiter craft lost in 1999
  - Miles versus Km between NASA engineers and contractor engineers
- ``The problem here was not the error, it was the failure of NASA's systems engineering, and the checks and balances in our processes to detect the error. That's why we lost the spacecraft."
  - Edward Weiler, NASA's associate administrator for space science

Systems Engineering Tools

- Requirements breakdown structure, FFBD, work breakdown structure, trade-off analysis, prototyping, simulation, etc.

System Development Models

- Waterfall model
- Vee model
- Spiral model

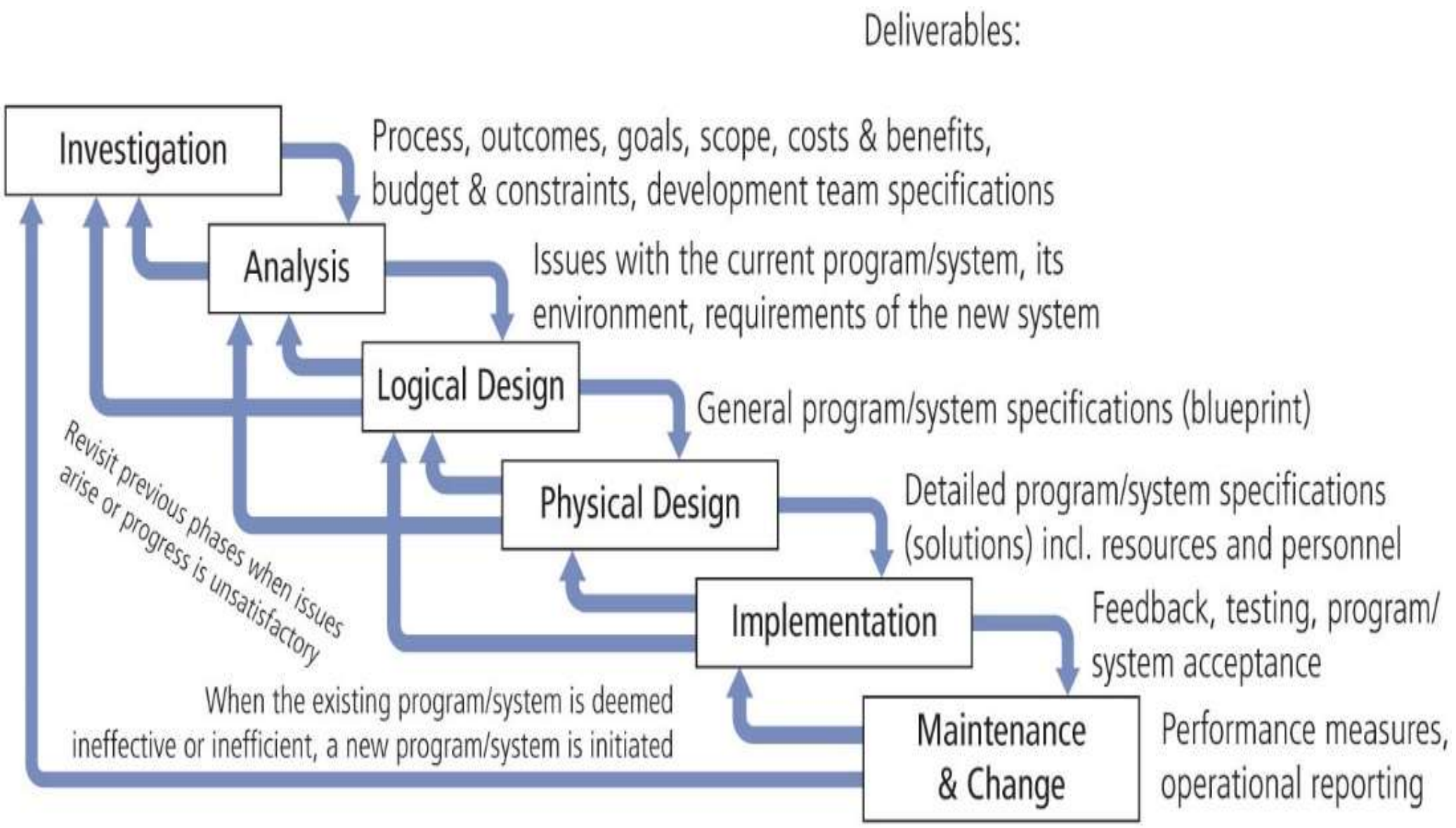
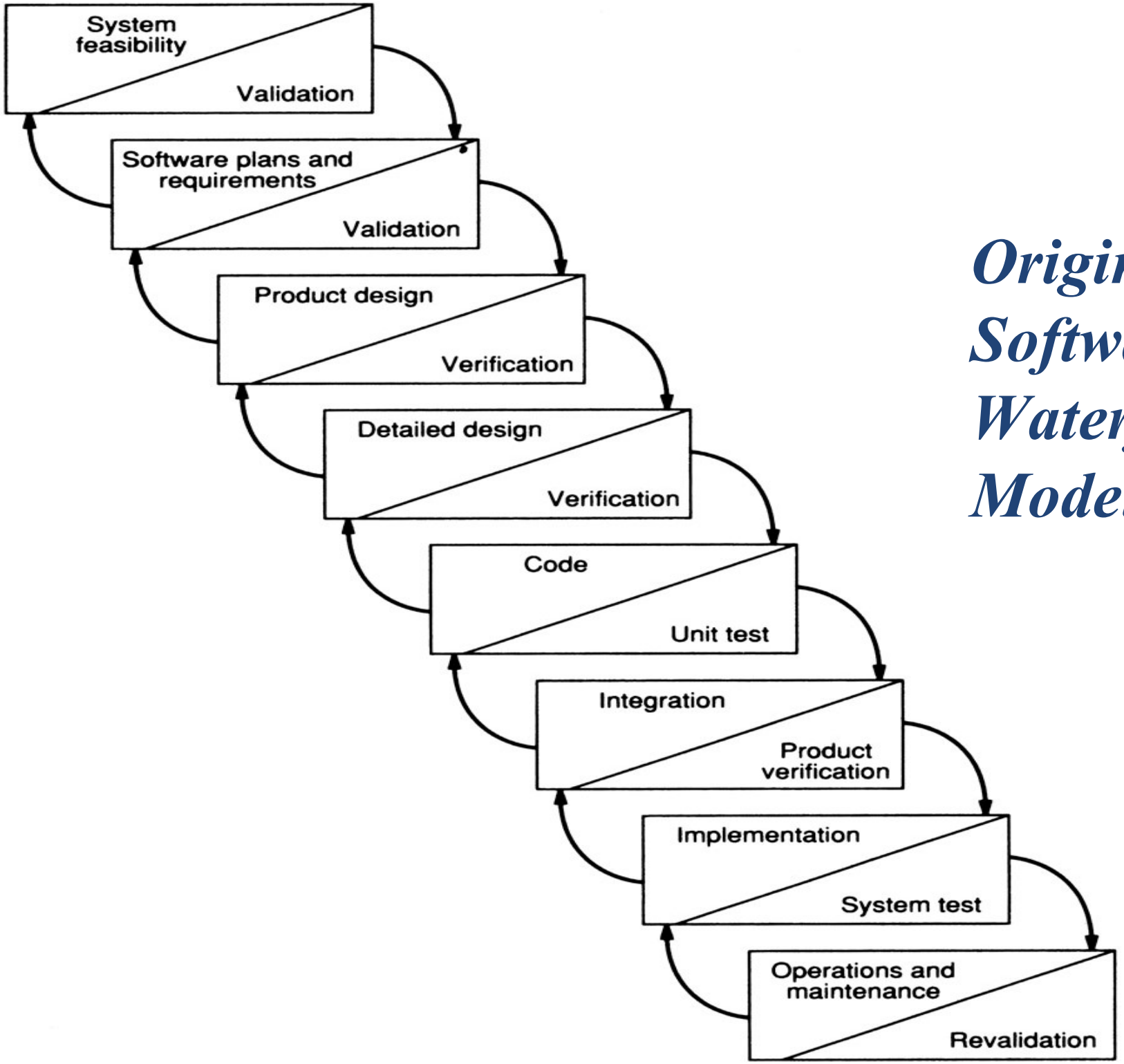
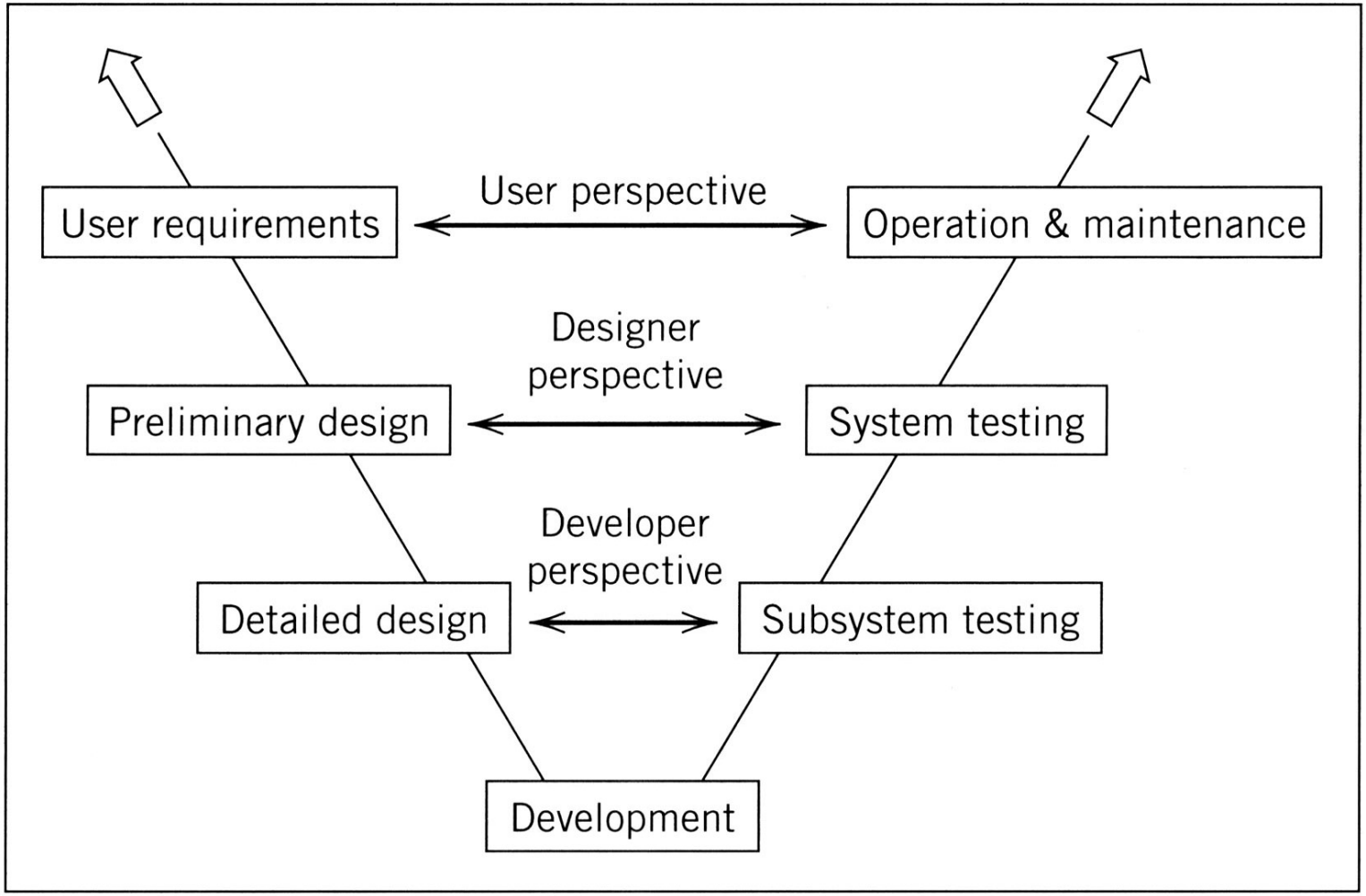


Figure 3-12 SDLC waterfall methodology



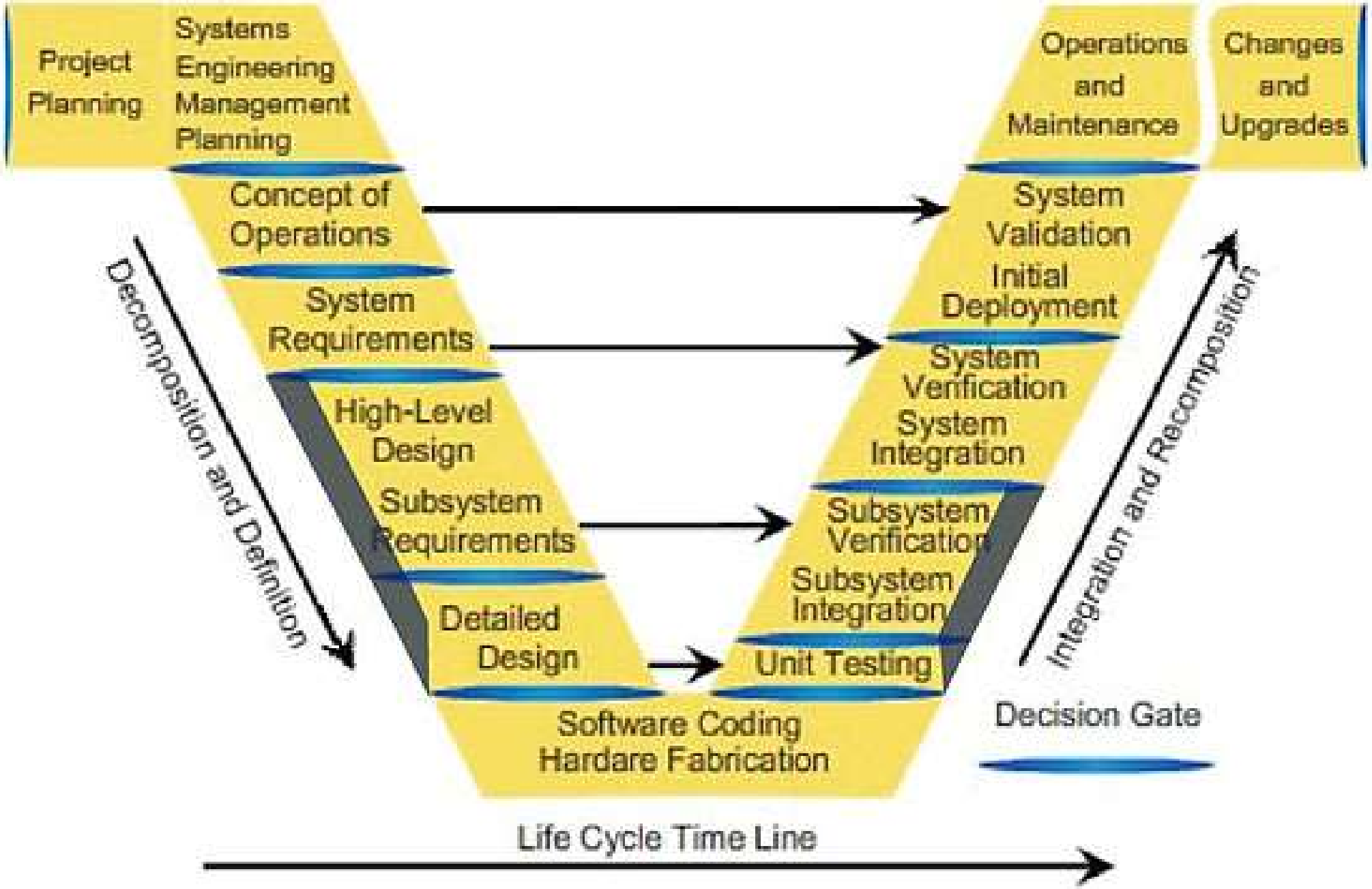
Original Software Waterfall Model

VEE Model

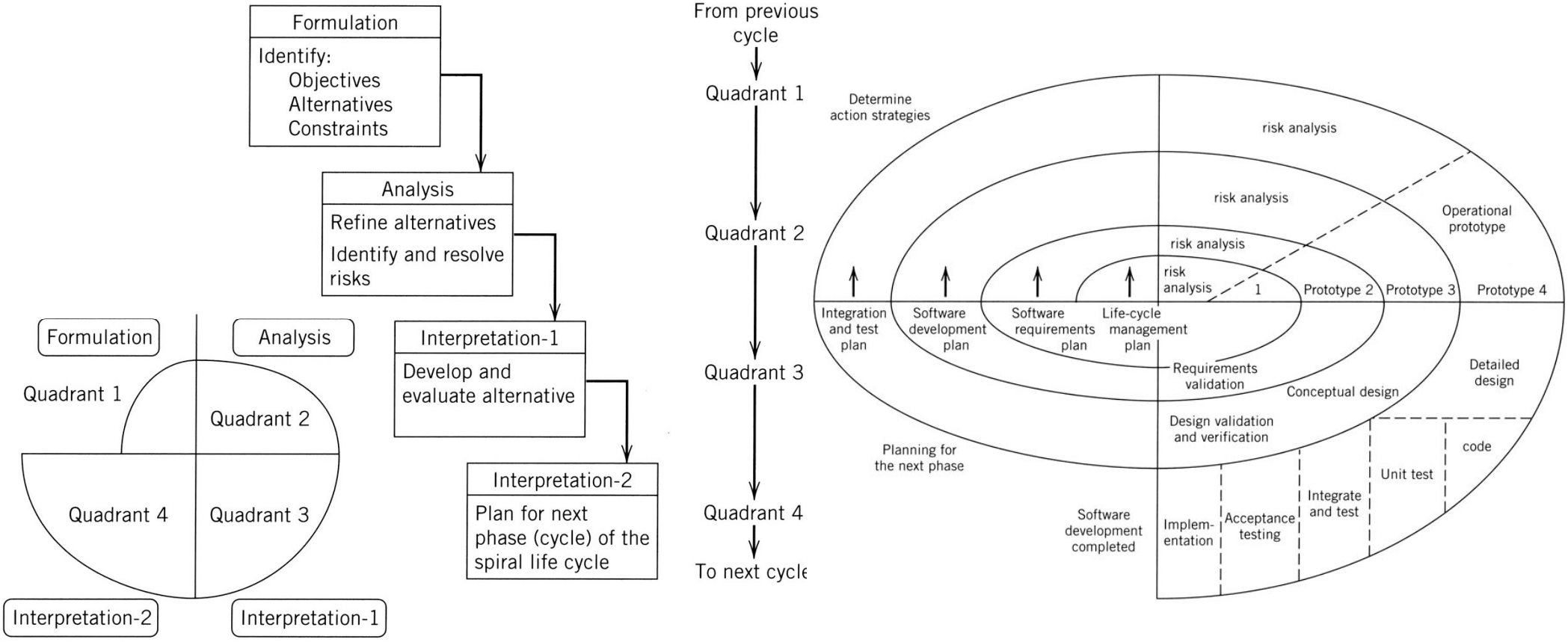


Top down design & Bottom up integration

VEE Model – More Detailed View



Spiral Model



Flow of activity in spiral life cycle

Rapid prototyping for high-risk development

More About Spiral Development Model

- It does not work well on large complex systems
  - It is expensive
  - It takes too long
- It is used on small but difficult projects, when
  - Requirements are not clearly understood
  - Technology is still under development
  - Multiple design approaches need to be evaluated
- It can be very effective when used as part of risk reduction program on critical parts of a large system.

Summary

- System engineering considerations
- Now we will go into the phases in more detail
  - Start with Conceptual Design