MODULE 2:
HOLISTIC DESIGN APPROACH FOR AUTONOMOUS SYSTEMS – APPLICATION OF SYSTEMS ENGINEERING CONCEPTS

Nicholas Ho
Institute of System Science, NUS

About Nicholas Ho

- nicholas.ho@nus.edu.sg
- Lecturer at NUS ISS; Courses covered include:
  - Robotic Systems
  - Autonomous Robots and Vehicles
  - Human-Robot System Engineering
- BEng and PhD degree from School of Mechanical Engineering, NUS
- Specialized in architecture, design & development
  - Artificial Intelligence
  - Augmented/Virtual Reality
  - Internet-of-Things (IoT) & Cyber-Physical System (CPS)

Contents

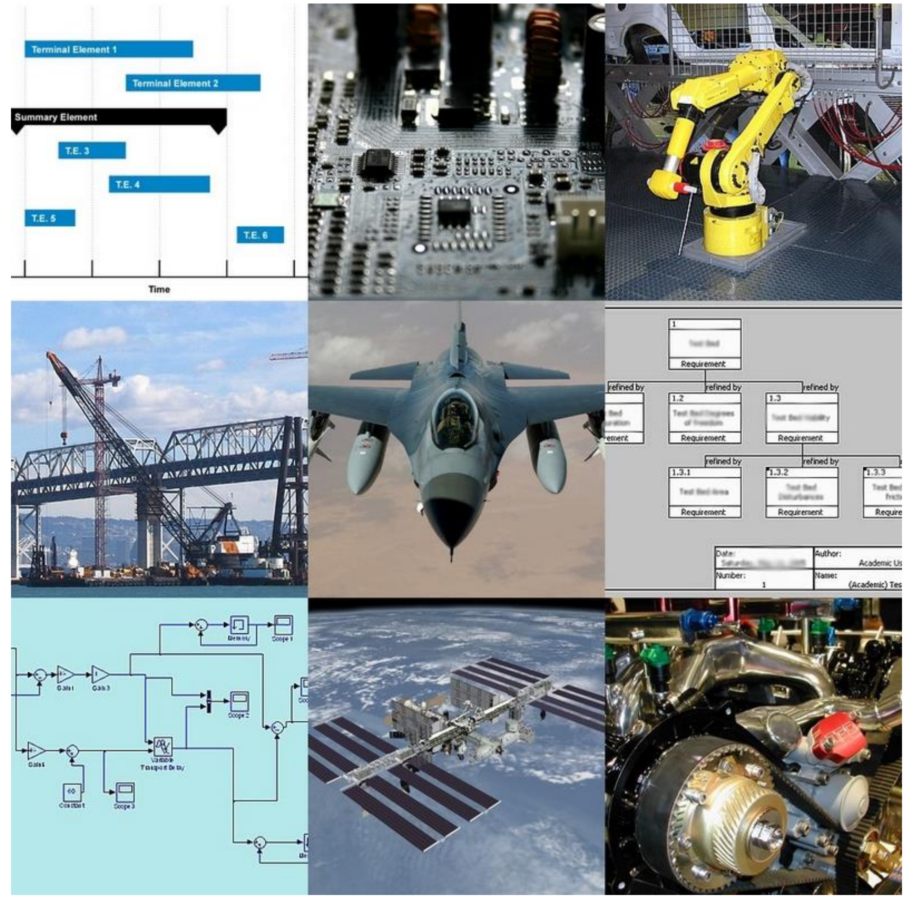
1. Fundamentals of Systems Engineering in relevance to autonomous systems
2. Systematic, holistic design approach for autonomous systems
3. <u>Workshop:</u> Designing a use case for autonomous systems using a holistic design approach

CHAPTER 1:
FUNDAMENTALS OF SYSTEMS ENGINEERING IN RELEVANCE TO AUTONOMOUS SYSTEMS

What is Systems Engineering?

ISO 29990:2010

- An interdisciplinary field of engineering aimed at **designing/managing complex projects**
- **Uses a host of tools and methods** to better comprehend and manage complexity in systems
- Offers great potential for **designing/managing autonomous systems** while also driving innovation



What is Systems Engineering?

What technical fields are involved in Autonomous Vehicles? How do we integrate them seamlessly?

??? Mechanical Engineering ???

??? ???

What is Systems Engineering?

Tools include:

1. System architecture
2. System model, Modeling, and Simulation
3. Optimization
4. System dynamics
5. Systems analysis
6. Statistical analysis
7. Reliability analysis
8. Decision making

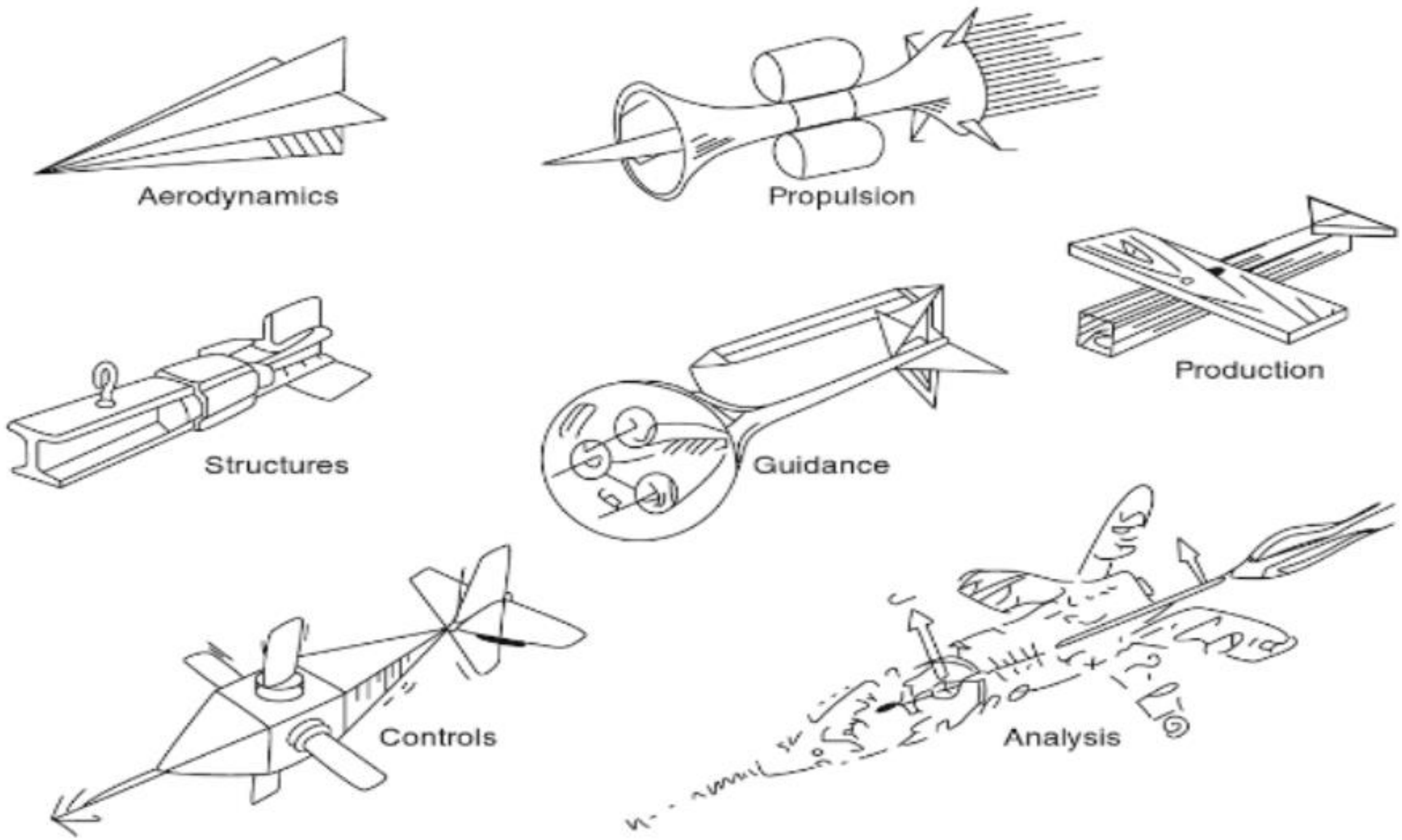
The International Space Station is an example of a very complex system requiring Systems Engineering

Importance of Systems Engineering

Adds value to product development

- Allows a systems engineer to **consider all objectives for the project** (e.g. business case, stakeholder needs, user preferences, regulatory compliance, production issues, technical limitations)
- Also **serves as an interface among different engineering disciplines**; integrated components from various teams of engineers
- **Key risk management tool** in the development of complex systems

Importance of Systems Engineering

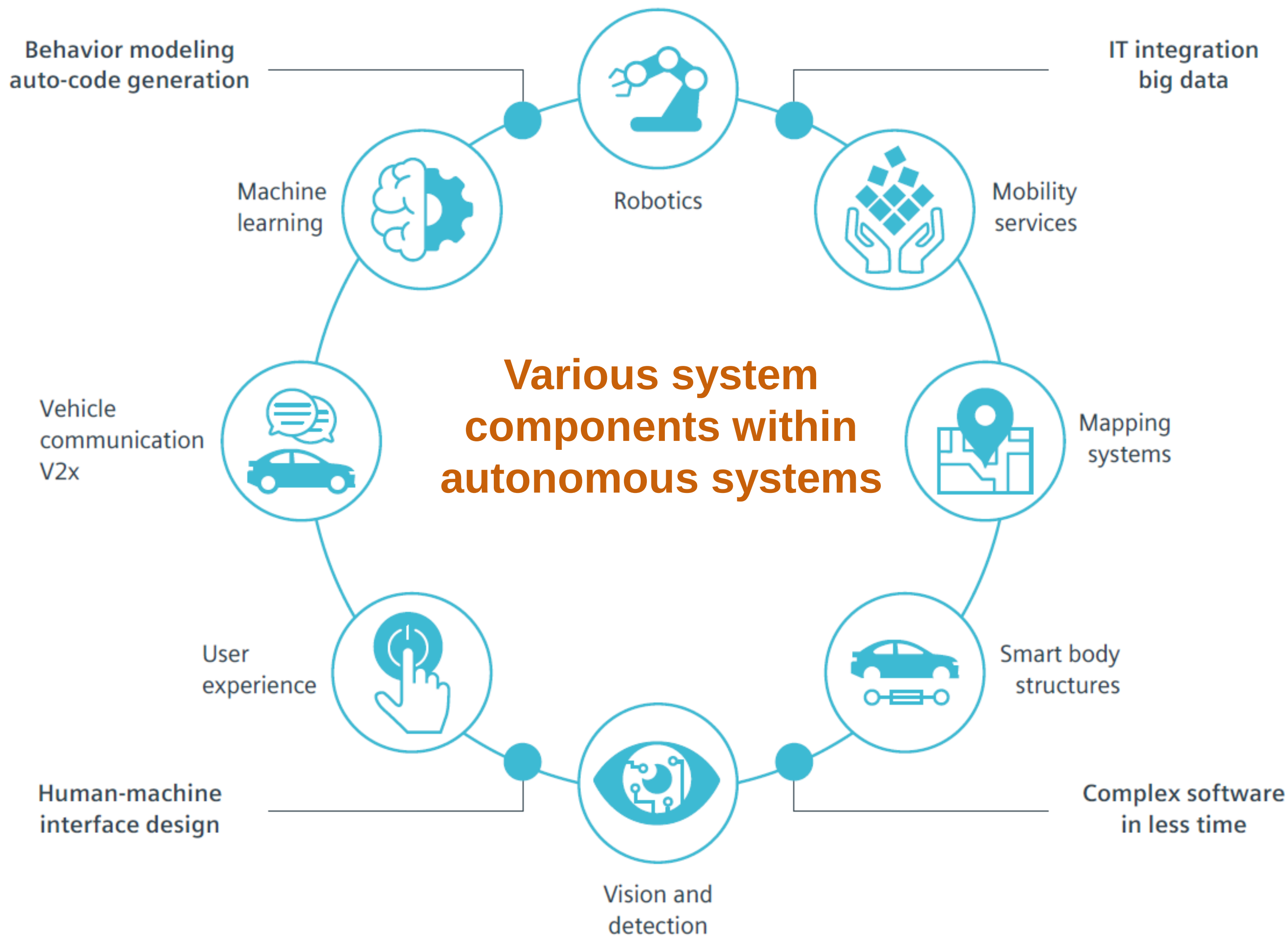


Ideal missile design as seen by various engineering teams

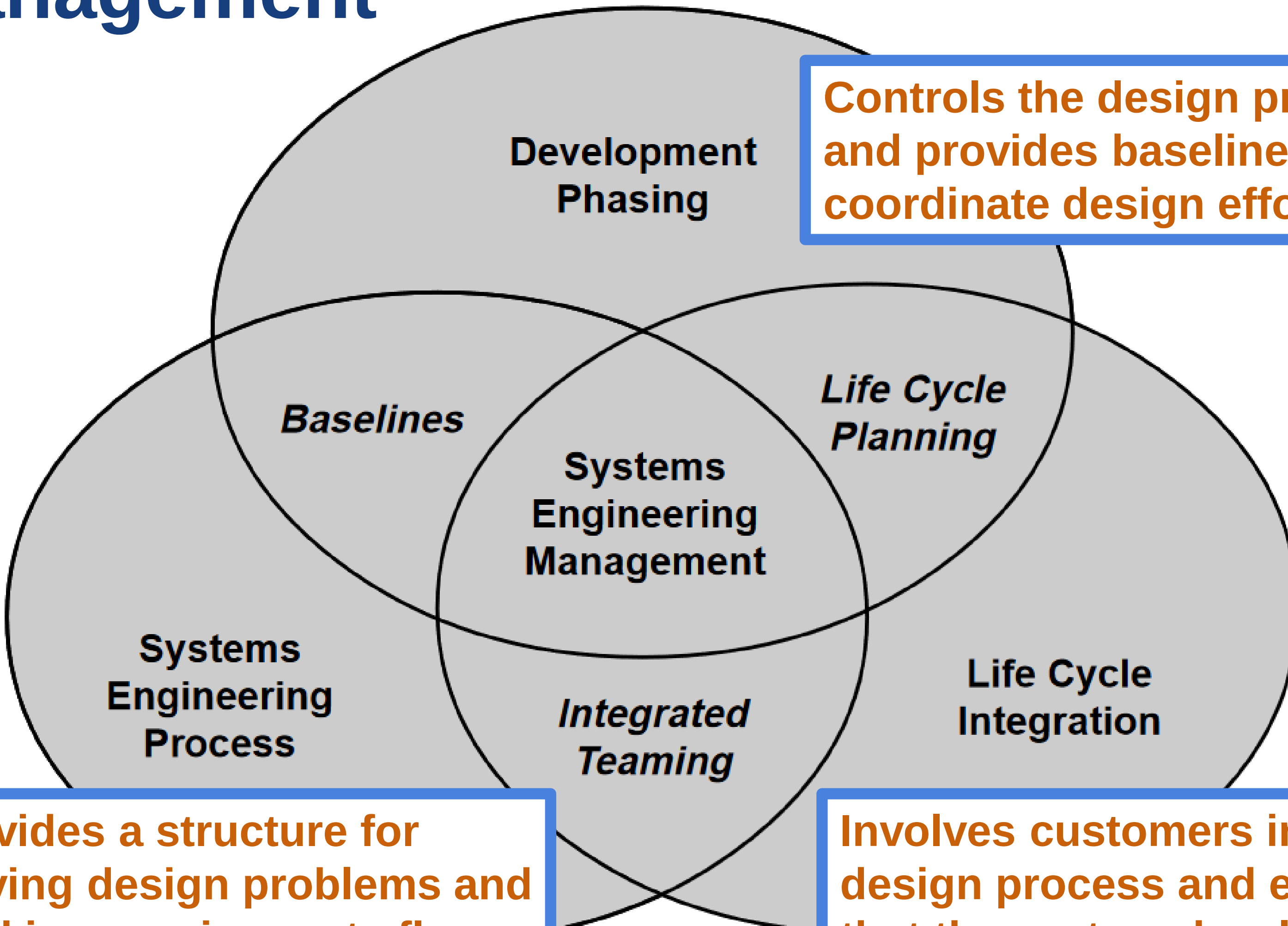
Importance of Systems Engineering

- **Emerging autonomous systems with increasing complexity**
- **Autonomous systems generally comprise of:**
  1. **Hardware elements** ranging from well-known mechanisms (e.g. wheels and axels on self-driving cars, motors on robotic arm) to
  2. **Innovative electronics** (e.g. LIDAR systems, RADAR, Stereo cameras and advanced sensors) to
  3. **Revolutionary software packages** (e.g. deep-learning algorithms for image recognition and navigation)
- **Risks and challenges** associated with autonomous systems may best be managed through a systems engineering approach

Importance of Systems Engineering



Systems Engineering Management

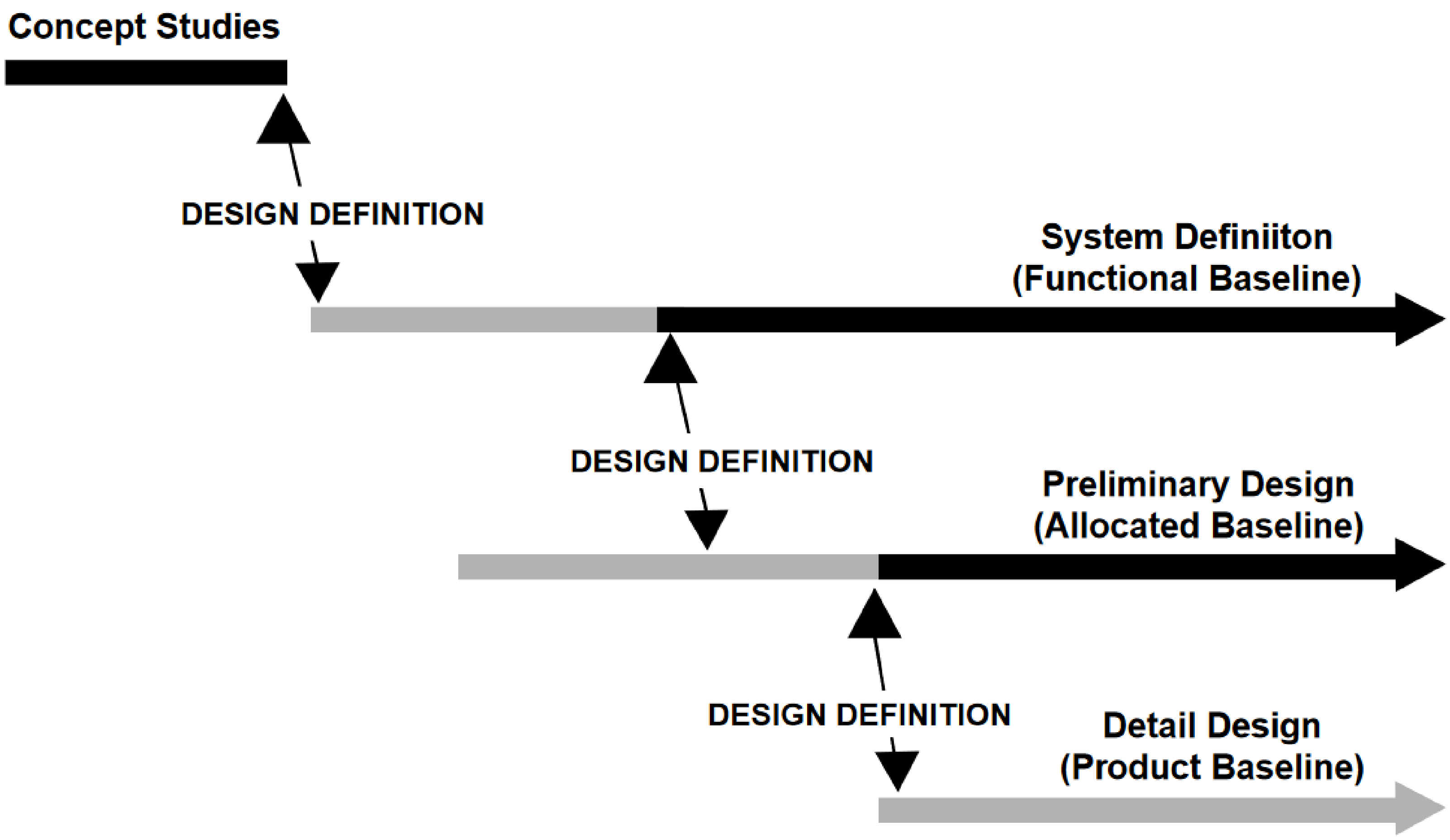


Controls the design process and provides baselines that coordinate design efforts

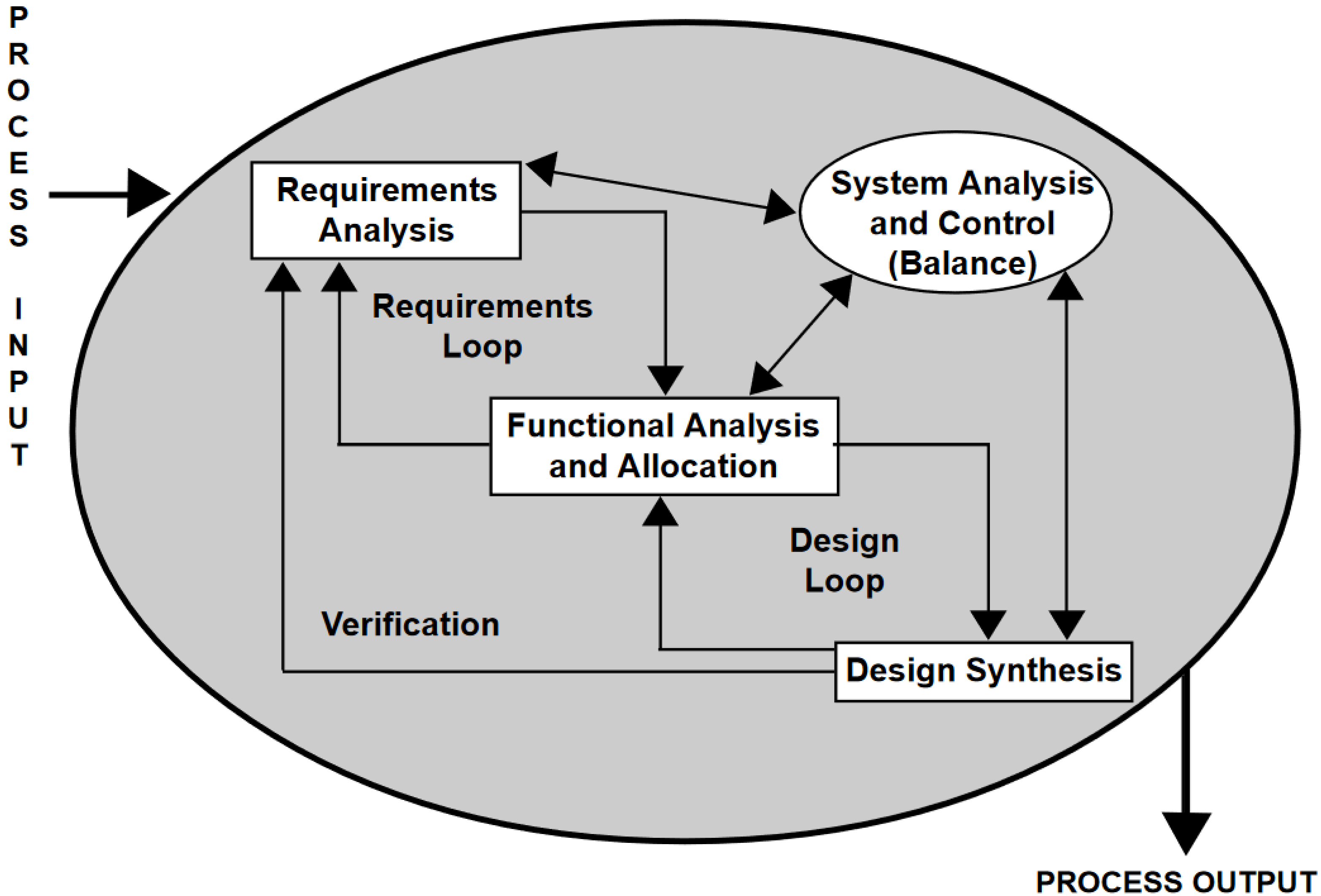
Provides a structure for solving design problems and tracking requirements flow through the design effort

Involves customers in the design process and ensures that the system developed is viable throughout its life

Development Phasing



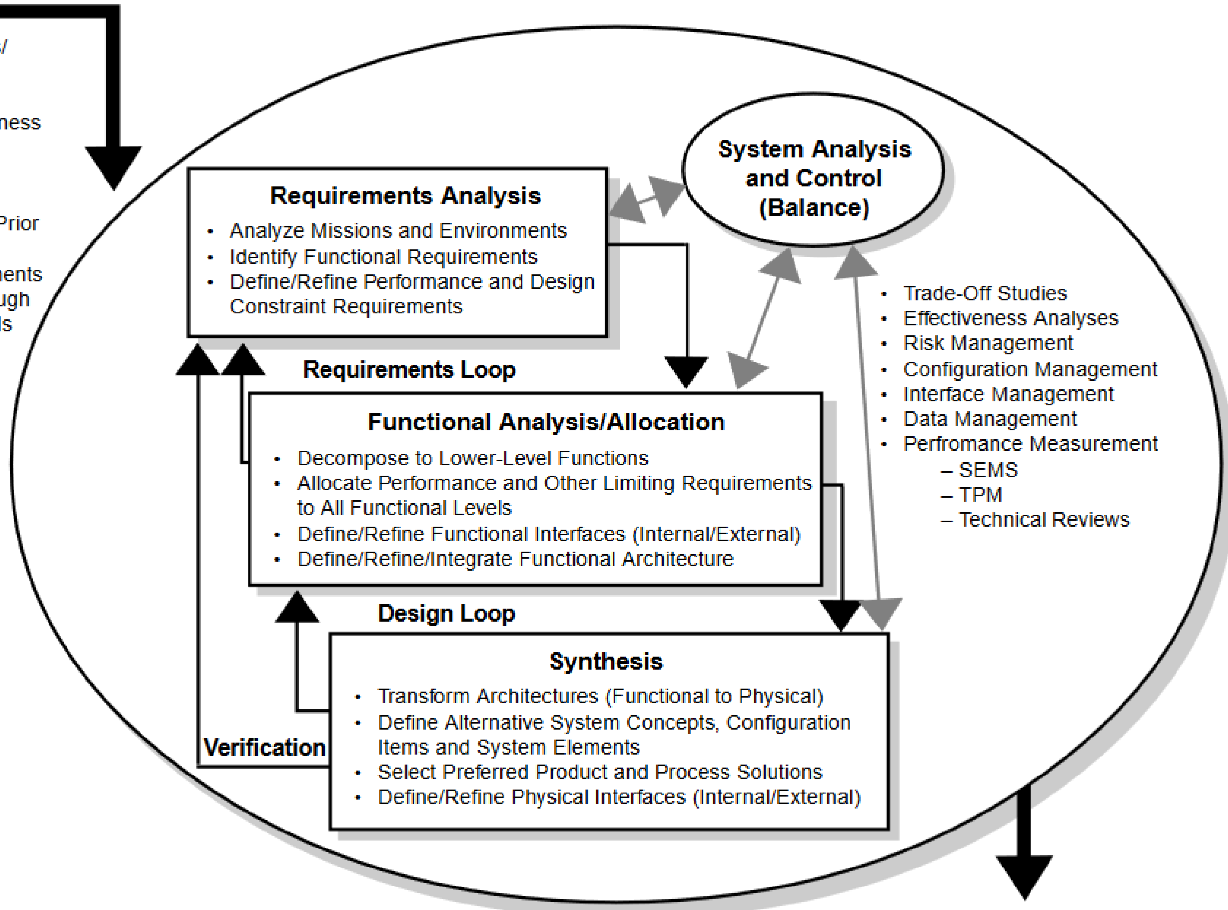
The Systems Engineering Process



The Systems Engineering Process

Process Input

- Customer Needs/Objectives/ Requirements
  - Missions
  - Measures of Effectiveness
  - Environments
  - Constraints
- Technology Base
- Output Requirements from Prior Development Effort
- Program Decision Requirements
- Requirements Applied Through Specifications and Standards



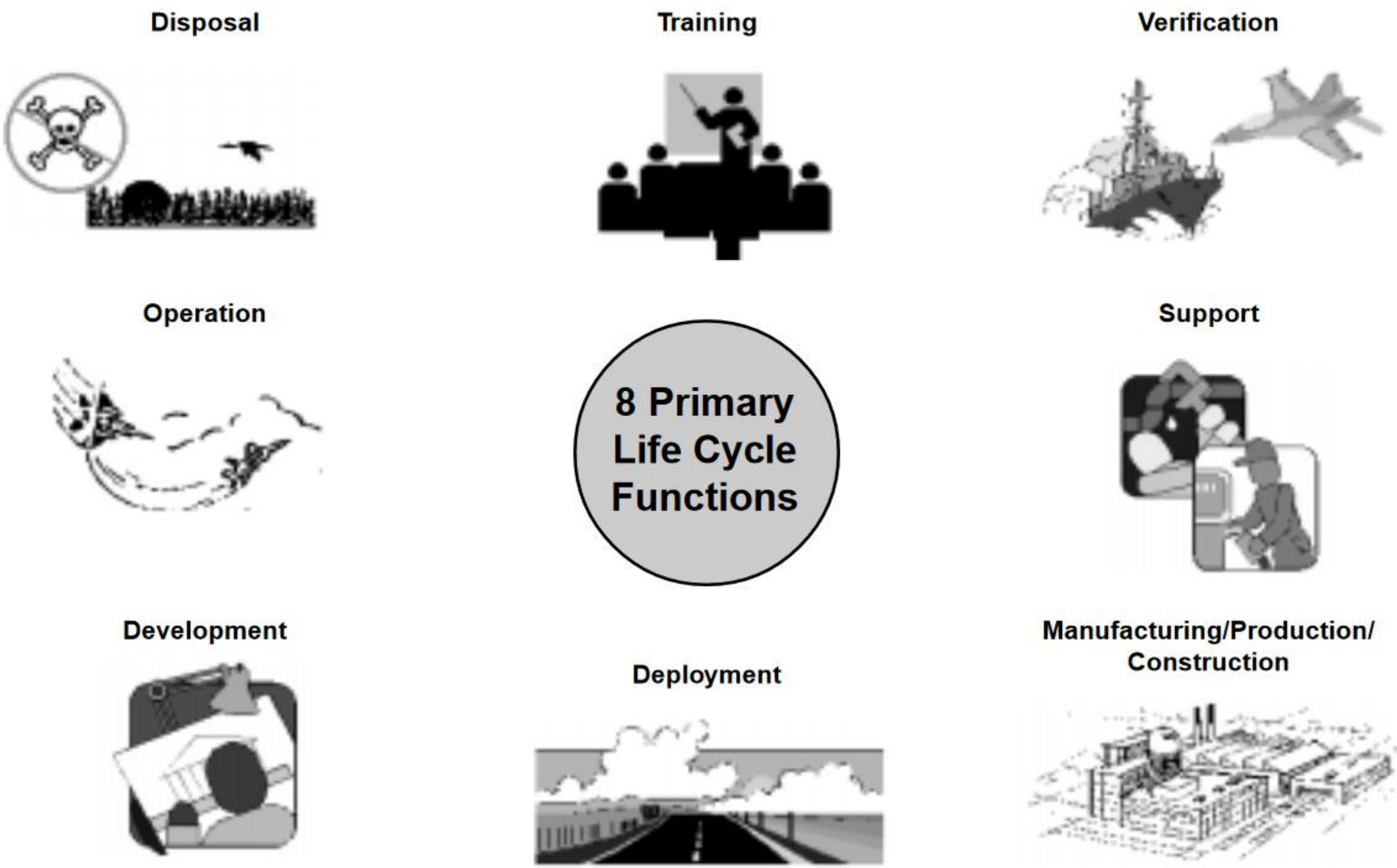
Process Output

- Development Level Dependent
  - Decision Database
  - System/Configuration Item Architecture
  - Specifications and Baselines

Related Terms:

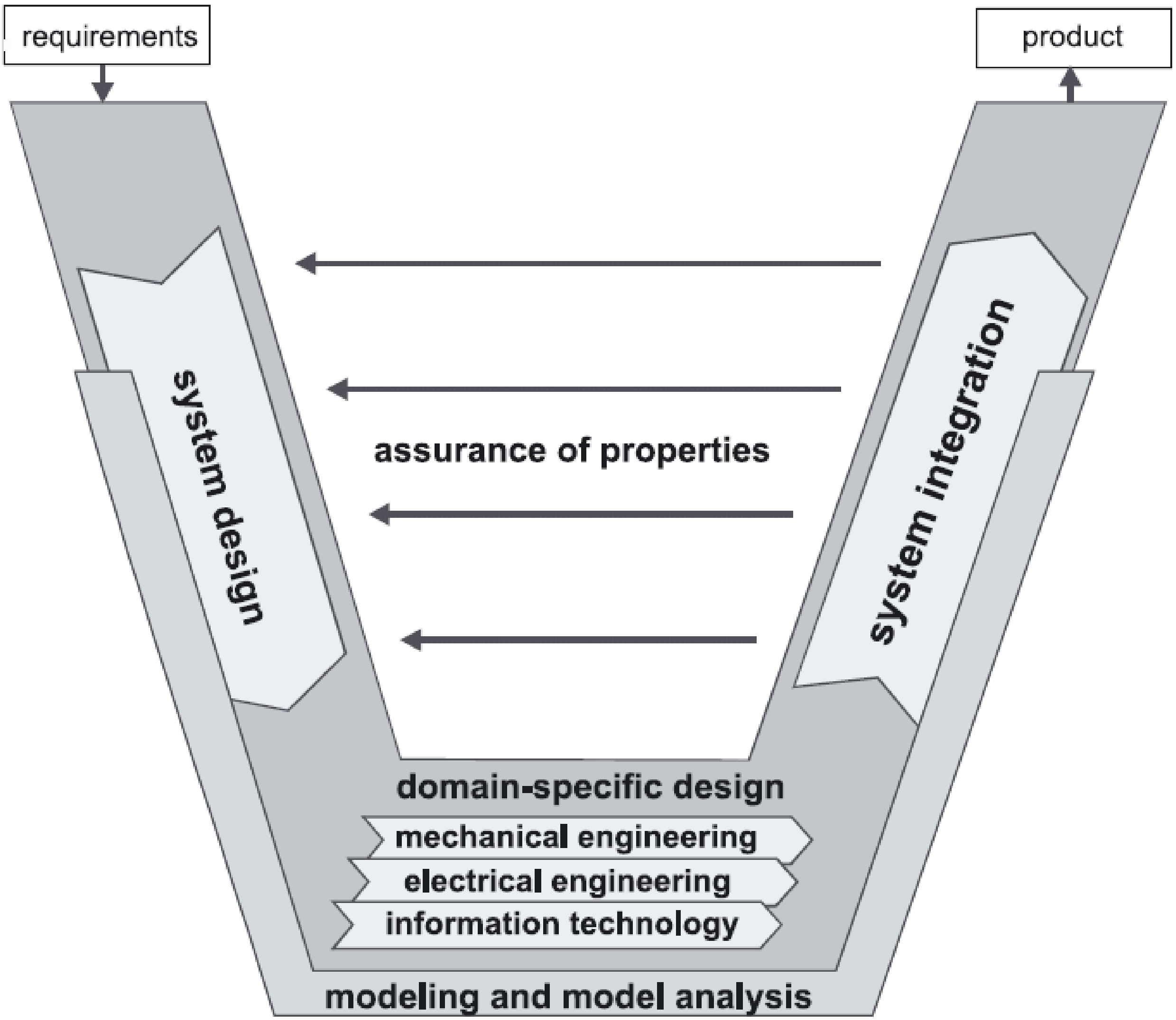
Customer =	Organizations responsible for Primary Functions
Primary Functions =	Development, Production/Construction, Verification, Deployment, Operations, Support, Training, Disposal
Systems Elements =	Hardware, Software, Personnel, Facilities, Data, Material, Services, Techniques

Life Cycle Functions



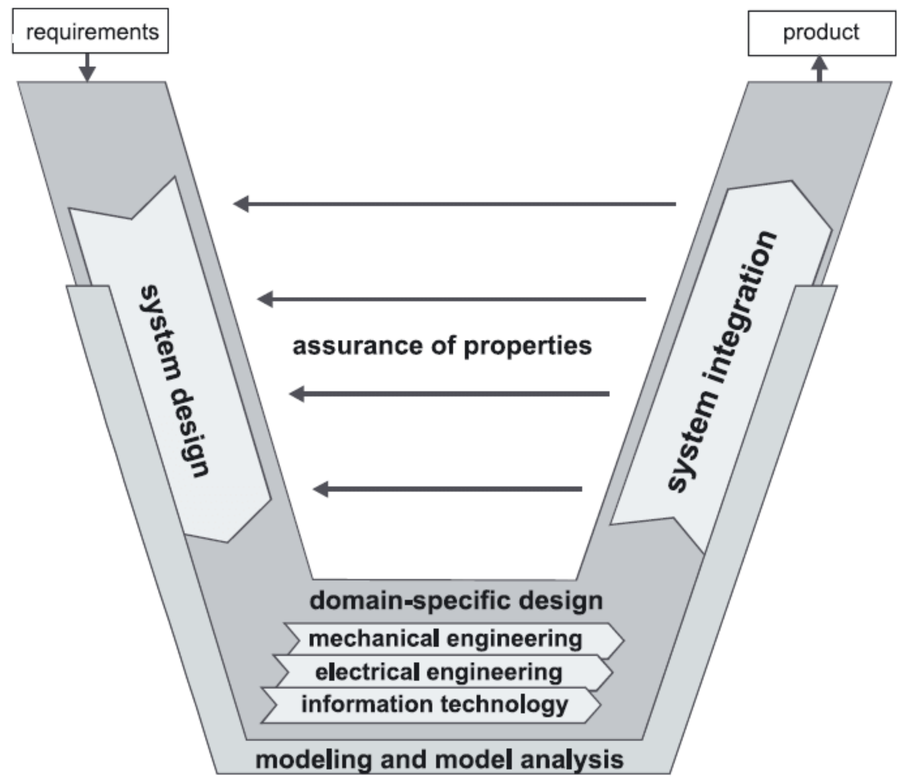
CHAPTER 2:
SYSTEMATIC, HOLISTIC DESIGN APPROACH FOR AUTONOMOUS SYSTEMS

Introducing the V-Model



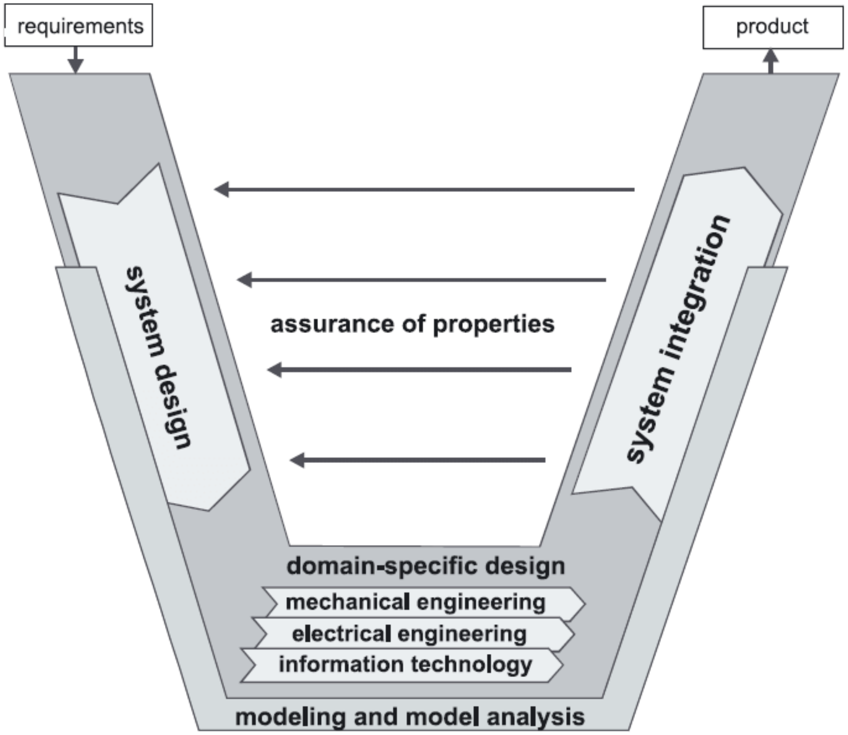
Introducing the V-Model

- **A visual representation of the main steps of systems engineering**
  - ➢ Lower blocks → more specific details
  - ➢ Higher blocks → broader perspective
  - ➢ Blocks at the same level correspond to one another
- **Describes the activities to be performed and the results that have to be produced during product development**

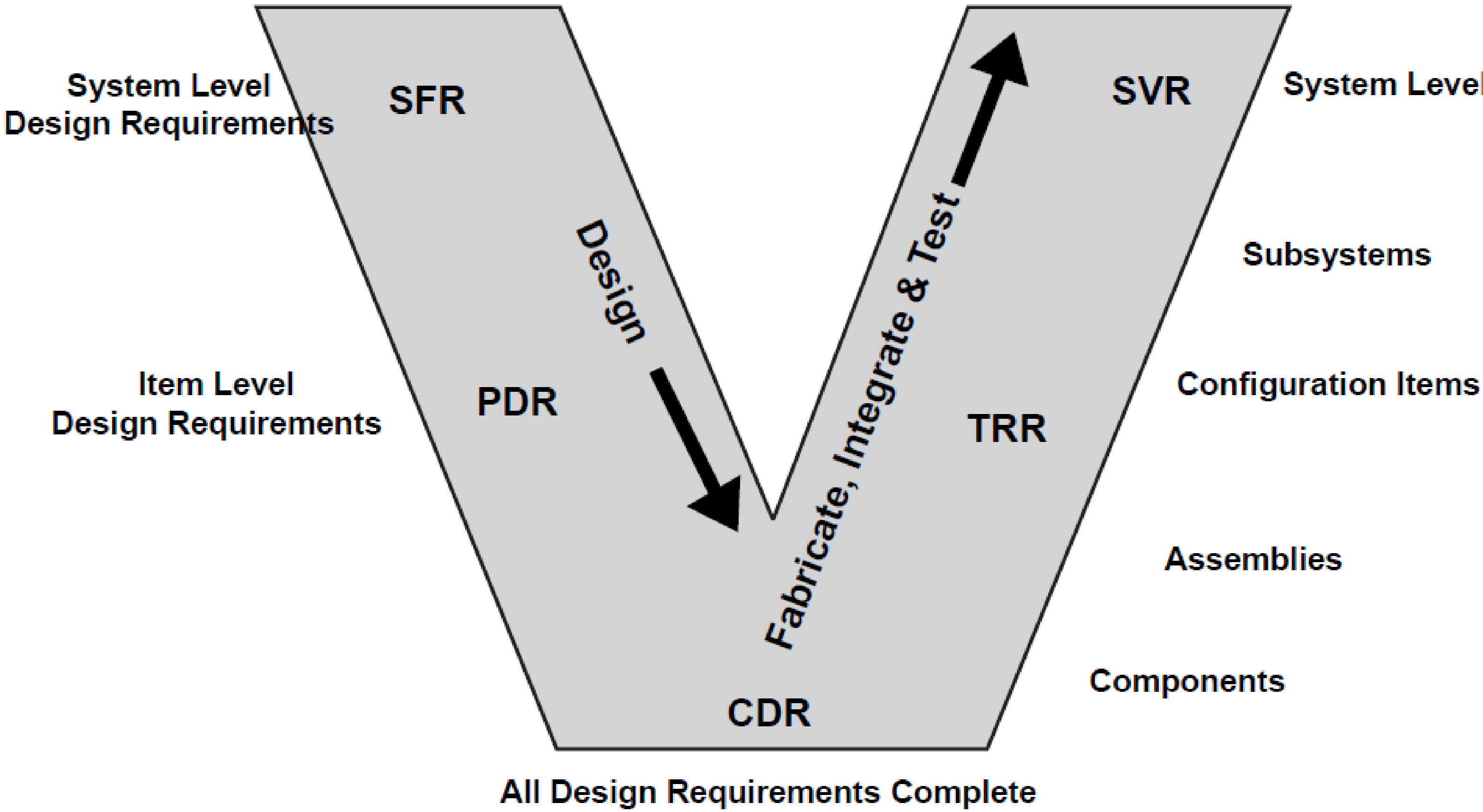


Introducing the V-Model

- **Provides guidance for the planning and realization of projects**
- **Objectives:**
    1. Minimization of project risks
    2. Improvement and guarantee of quality
    3. Reduction of total cost over the entire project and system life cycle
    4. Improvement of communication between all stakeholders (i.e. user, acquirer, supplier and develop)

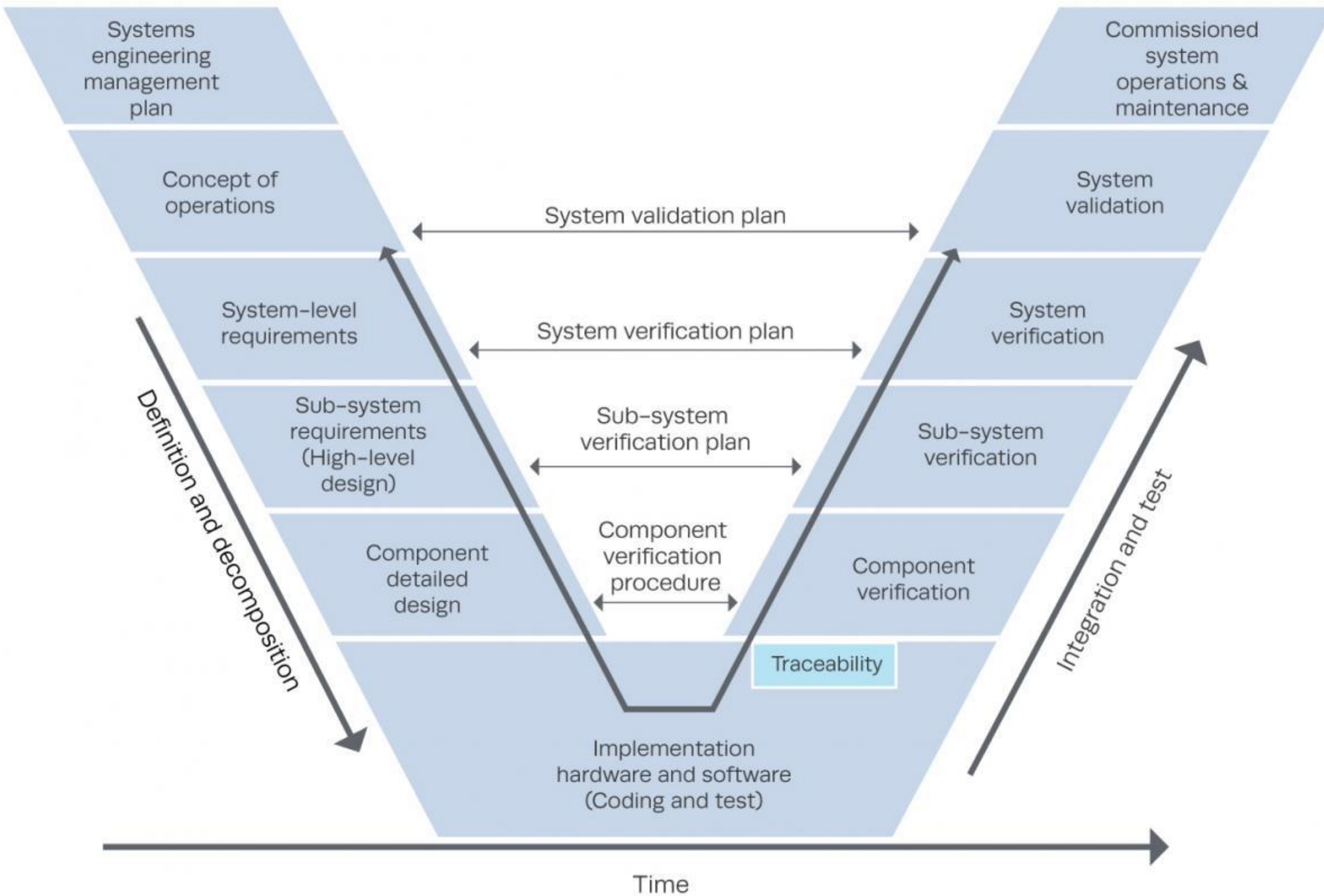


Introducing the V-Model



SFR = System Functional Review
PDR = Preliminary Design Review
CDR = Critical Design Review
TRR = Test Readiness Review
SVR = System Verification Review

Introducing the V-Model

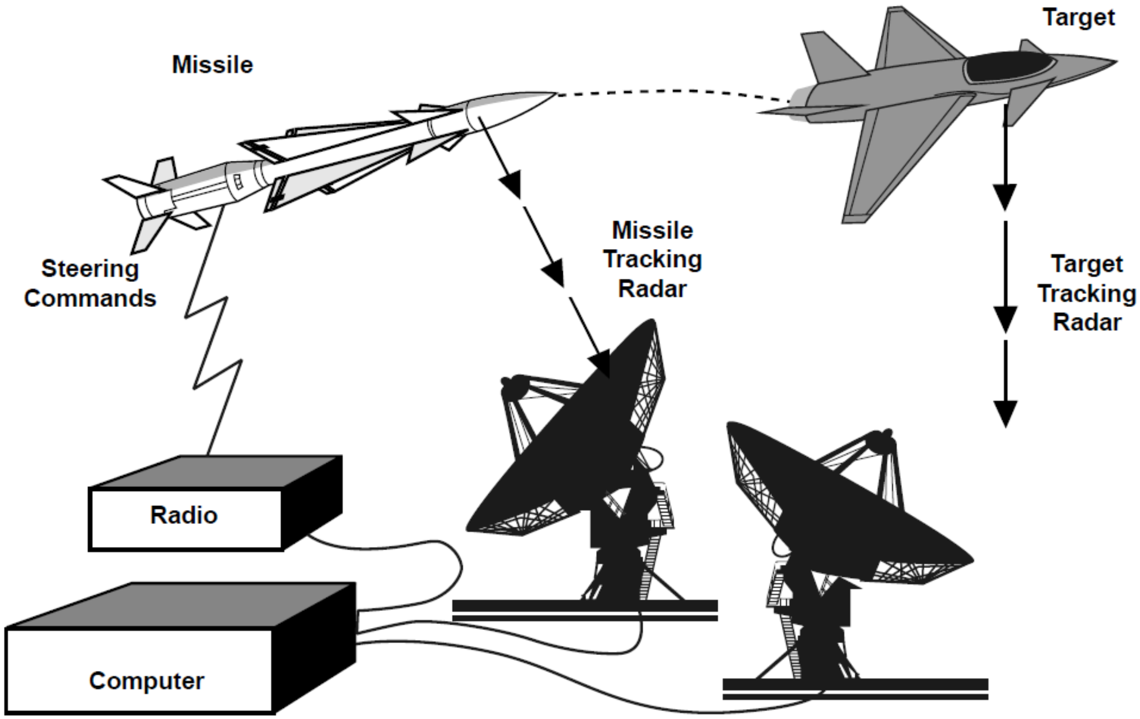


Management Plan

- Takes an initial product idea or identified problem and **plans how the project will proceed**
- Includes **identifying initial budgets, project timelines, and necessary personnel**

Concept of Operations

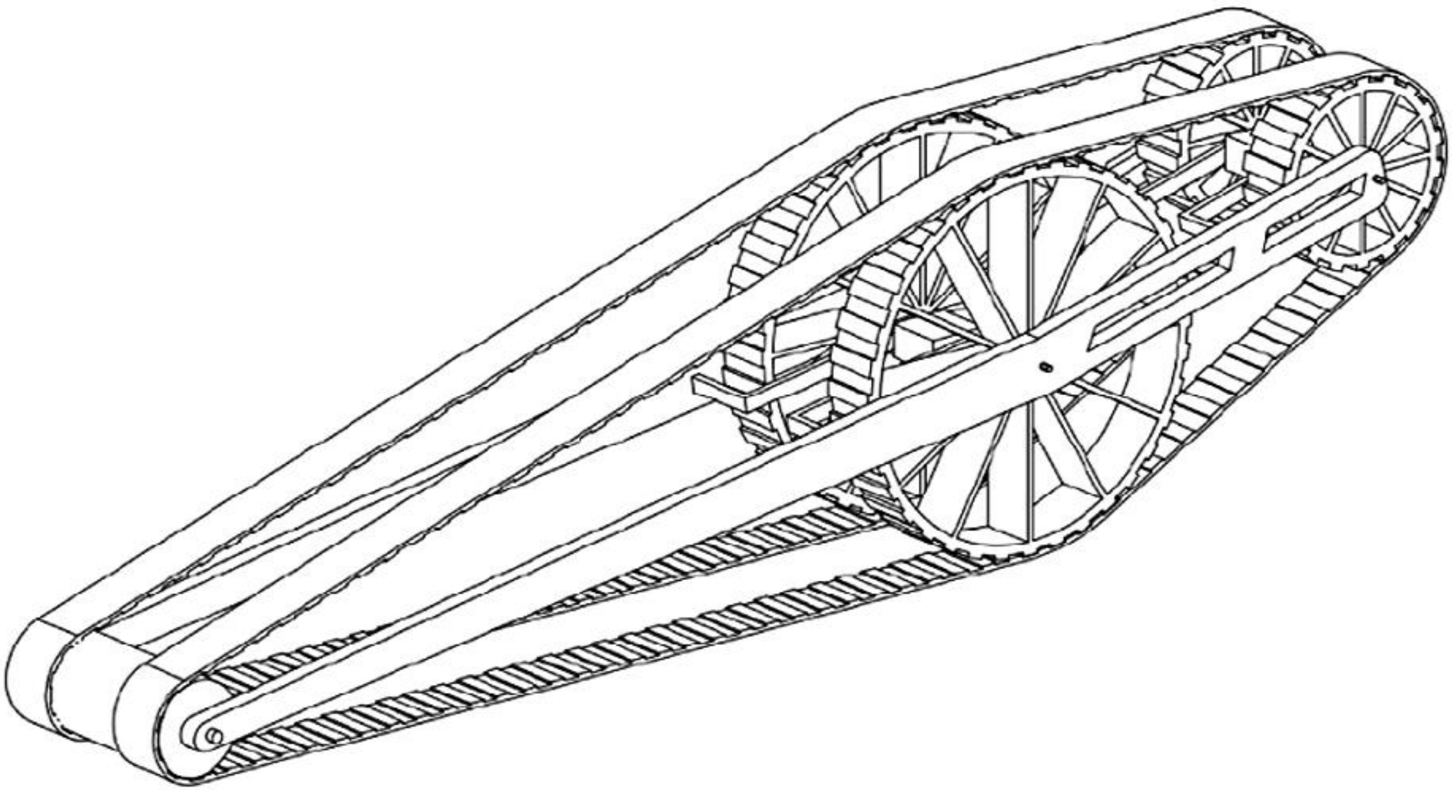
A concept of operations **defines who will use a system, what it will do, where it will do those things, and, to the extent relevant to an end user or important stakeholder, how it will accomplish those tasks**



External Command Guidance System

Concept of Operations

Can be in a drawing form:



Requirements

- “*A requirement is a **singular documented need—what a particular product or service should be or how it should perform**. It is a statement that identifies a **necessary attribute, capability, characteristic, or quality of a system in order for it to have value and utility to a user**.*”
- **Good requirements are unambiguous, measurable, testable, and traceable statements**
- Define the **criteria that must be met** by designs and implementation, and are directly tested on the right side of the V-Model

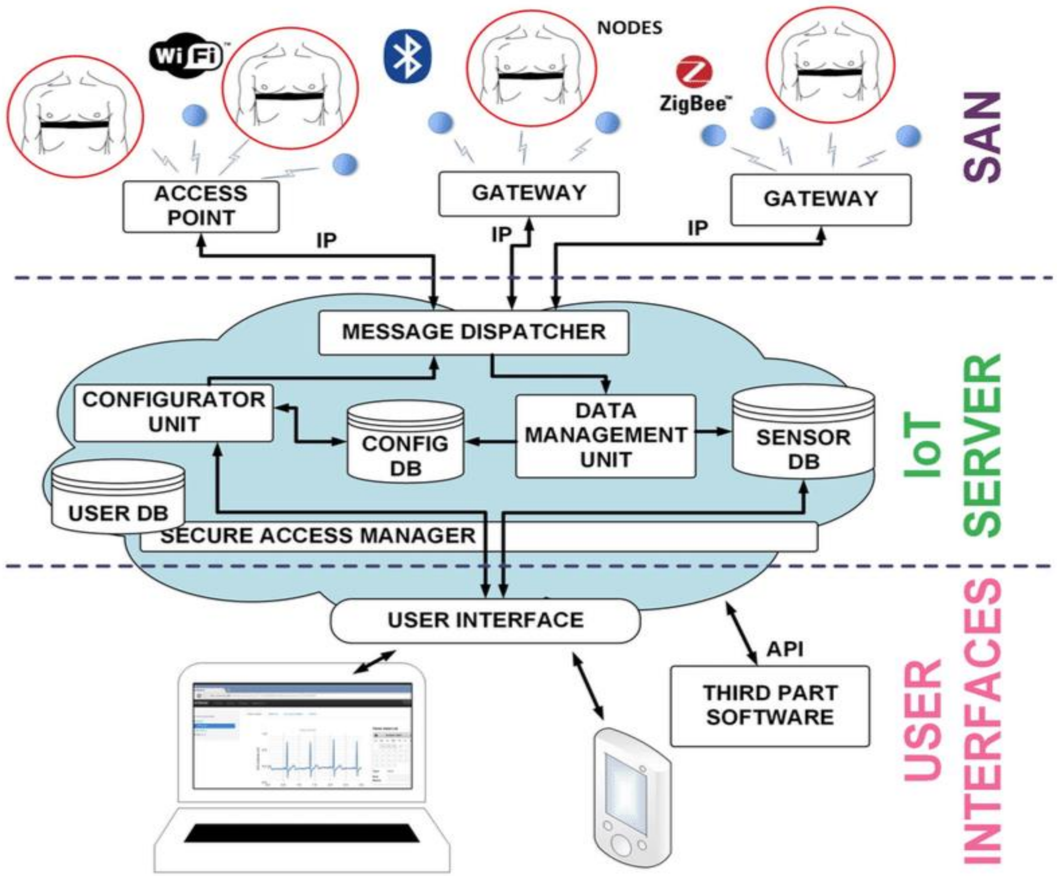
Requirements

Setting Final Specifications based on customer needs:

	METRIC	Units	Value
1	Attenuation from dropout to handlebar at 10hz	dB	>12
2	Spring pre-load	N	650
3	Maximum value from the Monster	g	<3.4
4	Minimum descent time on test track	s	<11.5
5	Damping coefficient adjustment range	N-s/m	>100
6	Maximum travel (26in wheel)	mm	43
7	Rake offset	mm	38
8	Lateral stiffness at the tip	kN/m	>75
9	Total mass	kg	<1.4
10	Lateral stiffness at brake pivots	kN/m	>425
11	Headset sizes	in	1.000 1.125
12	Steertube length	mm	150 170 190 210 230
13	Wheel sizes	list	26in
14	Maximum tire width	in	>1.75
15	Time to assemble to frame	s	<45
16	Fender compatibility	list	Zefal
17	Instills pride	subj	>4
18	Unit manufacturing cost	US$	<80
19	Time in spray chamber w/o water entry	s	>3600
20	Cycles in mud chamber w/o contamination	k-cycles	>25
21	Time to disassemble/assemble for maintenance	s	<200
22	Special tools required for maintenance	list	hex
23	UV test duration to degrade rubber parts	hours	>450
24	Monster cycles to failure	cycles	>500k
25	Japan Industrial Standards test	binary	pass
26	Bending strength (frontal loading)	MN	>100

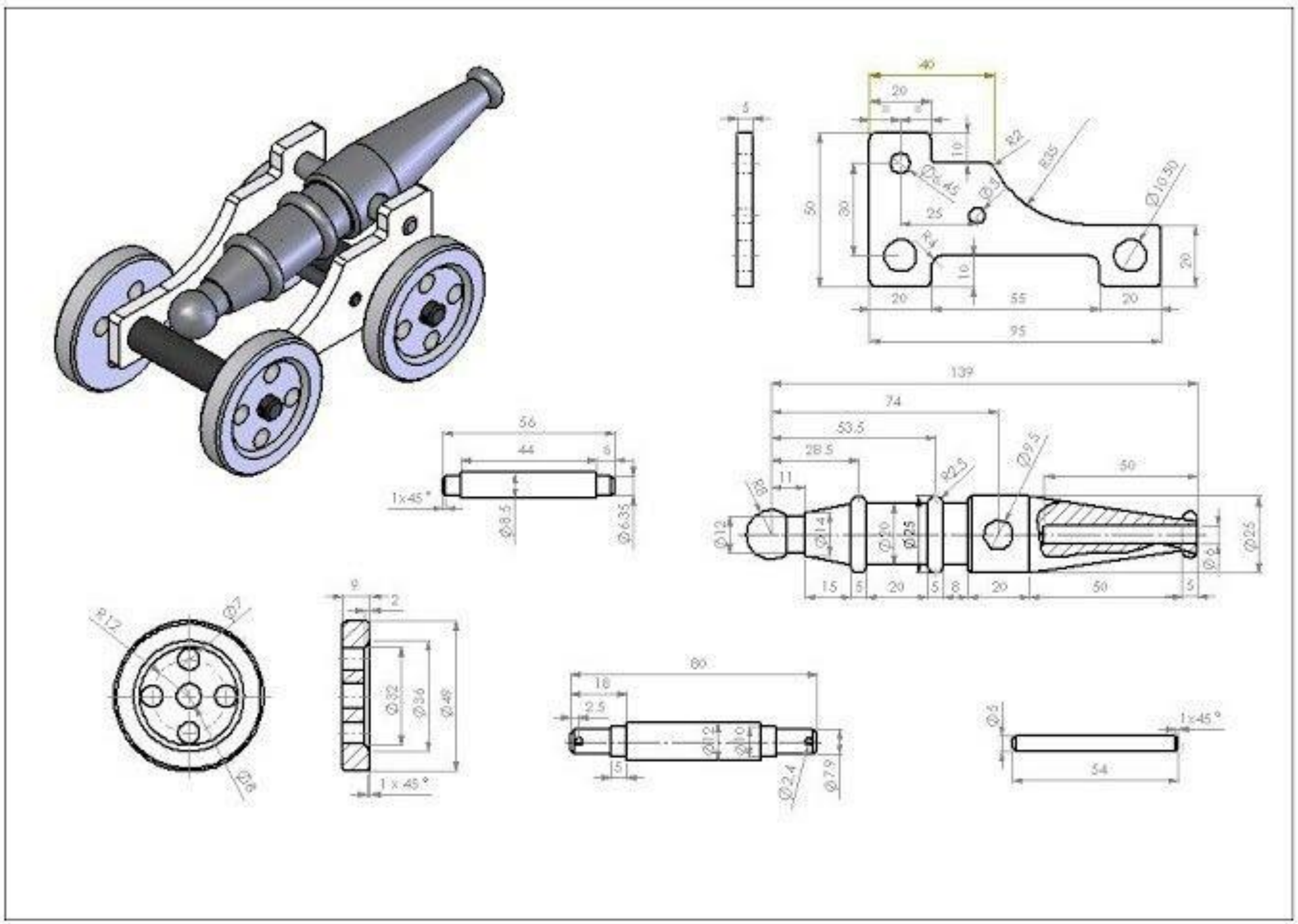
Detailed Design

Determining the **system's architecture** that summarizes the integration of components into a unified system



Detailed Design

Can be in a CAD drawing form:

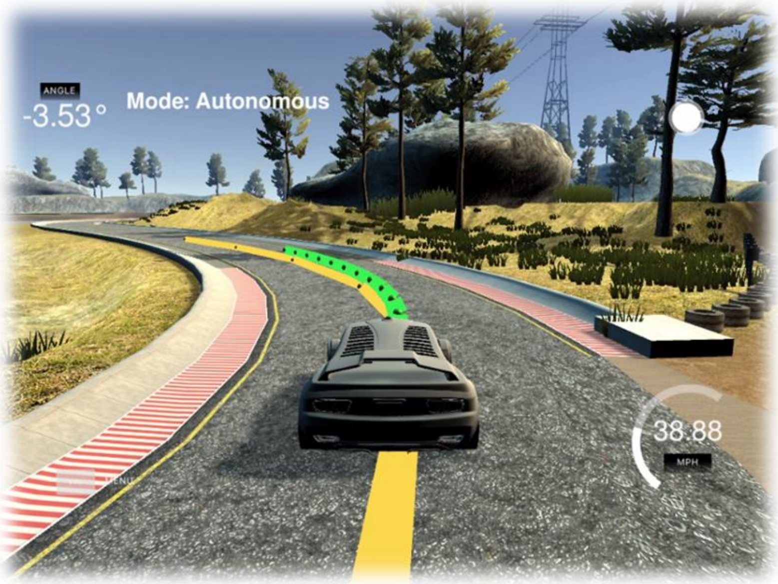


Implementation

- **Actual building of the prototype and software (coding)**
- **Prototype can come in 2 forms:**
  - Virtual (simulated)
  - Physical (manufactured)
- **Convert** all previously generated design and specification documents **into a functional system**

Implementation

Virtual vs Physical Prototype



Verification

- **Verify for the following 3 levels in sequence:**
  - Component, sub-system and system
- **Component verification** – ensure individual components meet each respective requirement (e.g. LIDAR sensor for AV, individual keys for PC)
- **Sub-system verification** – ensure sub-systems meet their respective requirements (e.g. combined sensory system for AV, entire keyboard for PC)
- **System verification** – ensure the system as a whole meet the requirements (e.g. AV, PC)

Verification

Component

Sub-system

System

System Validation

- Ensuring that the full system—made up of integrated components—**meets the user's needs as outlined in the concept of operations**
- **Tests have to end off with real target users in the real operational environment**
- **Should test all possible scenarios faced by the system**, including environmental influences, varieties of user input, and all the ways that system components might fail

Validation: "Are you building the right thing?" VS Verification: "Are you building it right?"

Operation & Maintenance

- Applicable for finalized product; not applicable for prototypes
- Involves **deploying the system to real-world users**
- **Continued monitoring** to ensure that production models comply with requirements and user needs, eliciting **feedback** from users, and guiding updates or alterations
- **Unforeseen or underestimated risks** which become prominent with real use can be **identified and addressed**
- **Continued need for a systems engineering perspective** even after product reaches market (e.g. Smart Phones)

Operation & Maintenance

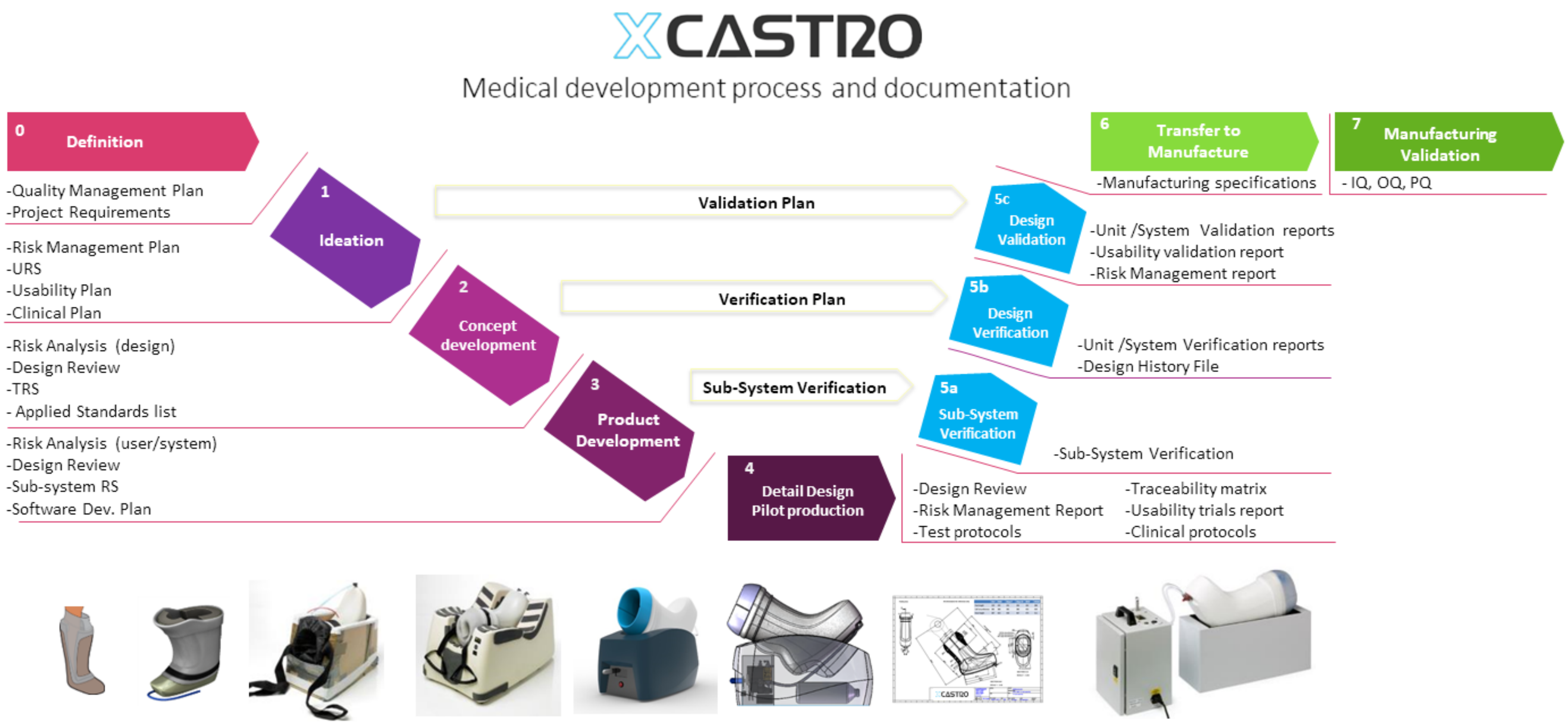
Other Notes on the V-Model

- **In practice, systems engineering is <u>not as linear</u> as the V-Model approach** suggests, but the model provides **a useful basic understanding of key activities**

- Other systems engineering frameworks like the **“spiral model”** rearrange the same steps and core activities into a more flexible or iterative model

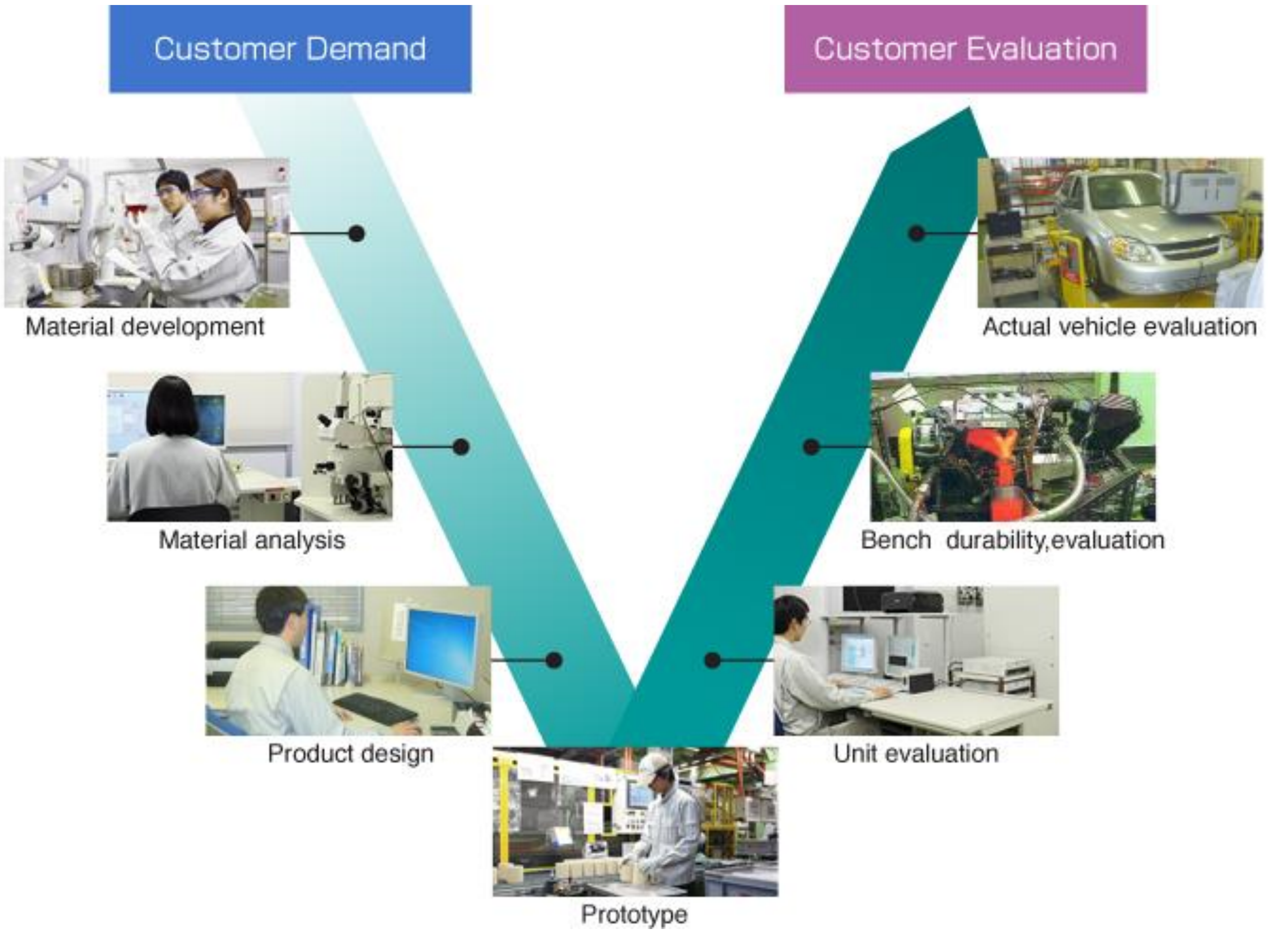
Application Example

Medical devices development process (V-Model)

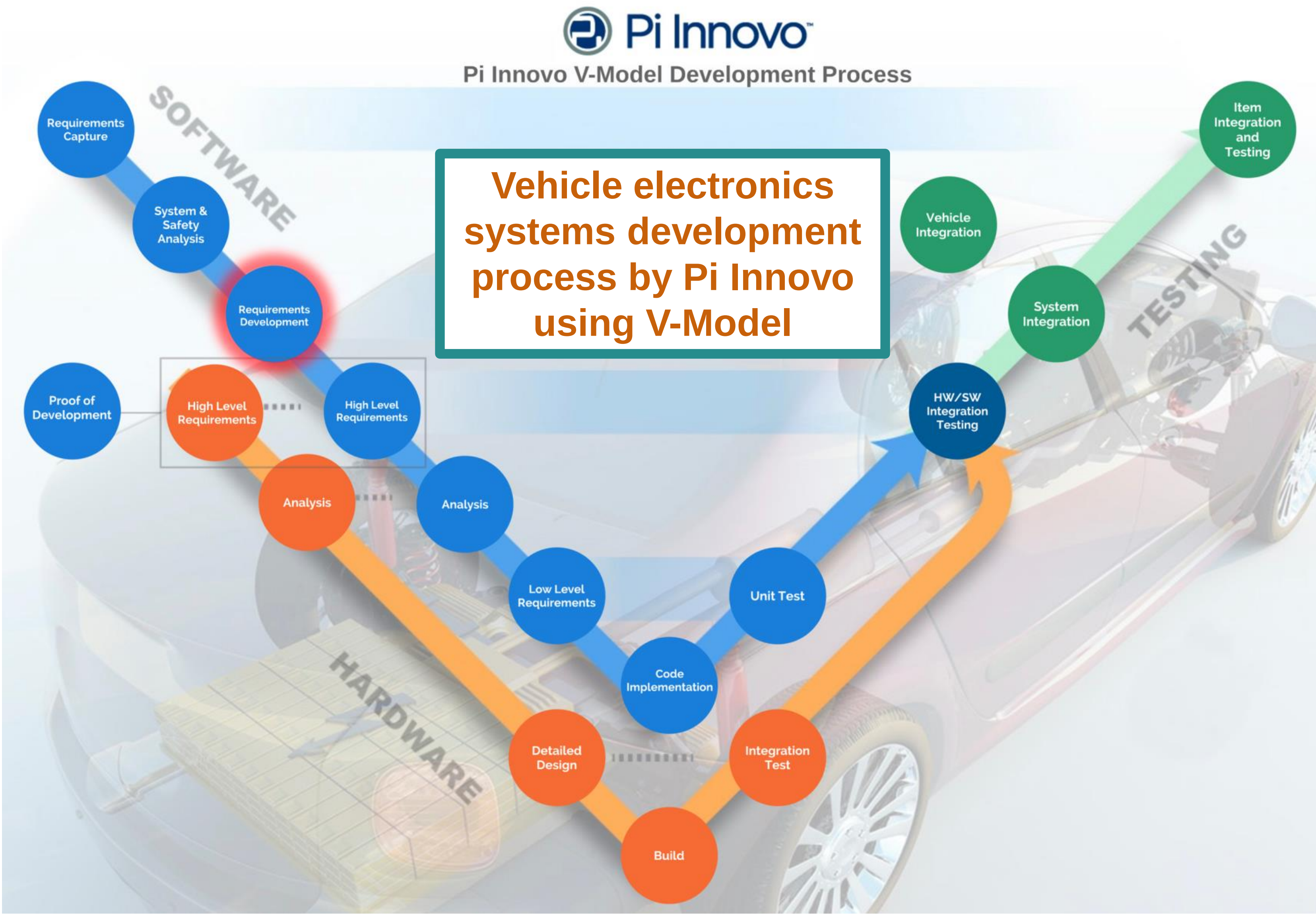


Application Example

Automotive catalyst development process by Cataler using V-Model:



Application Example



Case Study Example

Using V-Model to design and develop an integrated IoT system for home use; to monitor and control environmental conditions at home using this system:

Recall V-Model Steps:

1. Management Plan
2. Concept of Operations
3. System-level Requirements
4. Sub-system-level Requirements
5. Detailed Design
6. Implementation
7. Verification
8. System Validation
9. Operation & Maintenance

Case Study Example

Step 1: Management Plan

- Objective is to build an IoT (Internet-of-Things) system for the aforementioned uses
- **Plans:**
  - **Total budget for development** → $10,000 (assuming that similar current commercial product is selling at this price); **breakdown of budget** among manpower, materials, software-related services, equipment
  - **Project timeline (i.e. development time)** → 6 months (assuming that this is average development time for similar products); **Gantt chart planning** from Step 2 to Step 9
  - **Necessary personnel** → project manager, software engineers, electrical engineers, data engineers, mechanical engineers (based on the product idea or identified problem, and allowable budget)

Case Study Example

Step 1: Management Plan

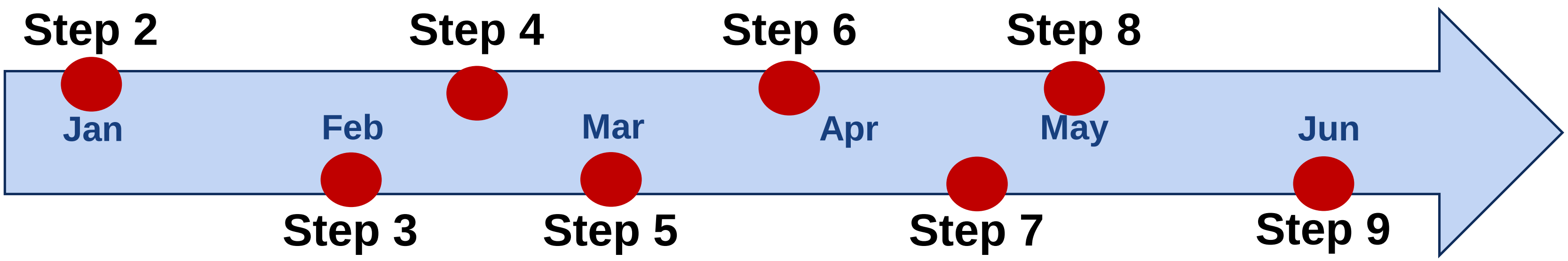
Breakdown of budget:

Items	Budget Allocation
Manpower	$5,000
Materials	$900
Software-related services	$600
Equipment	$1,500
Others	$2,000
Total:	**$10,000**

Case Study Example

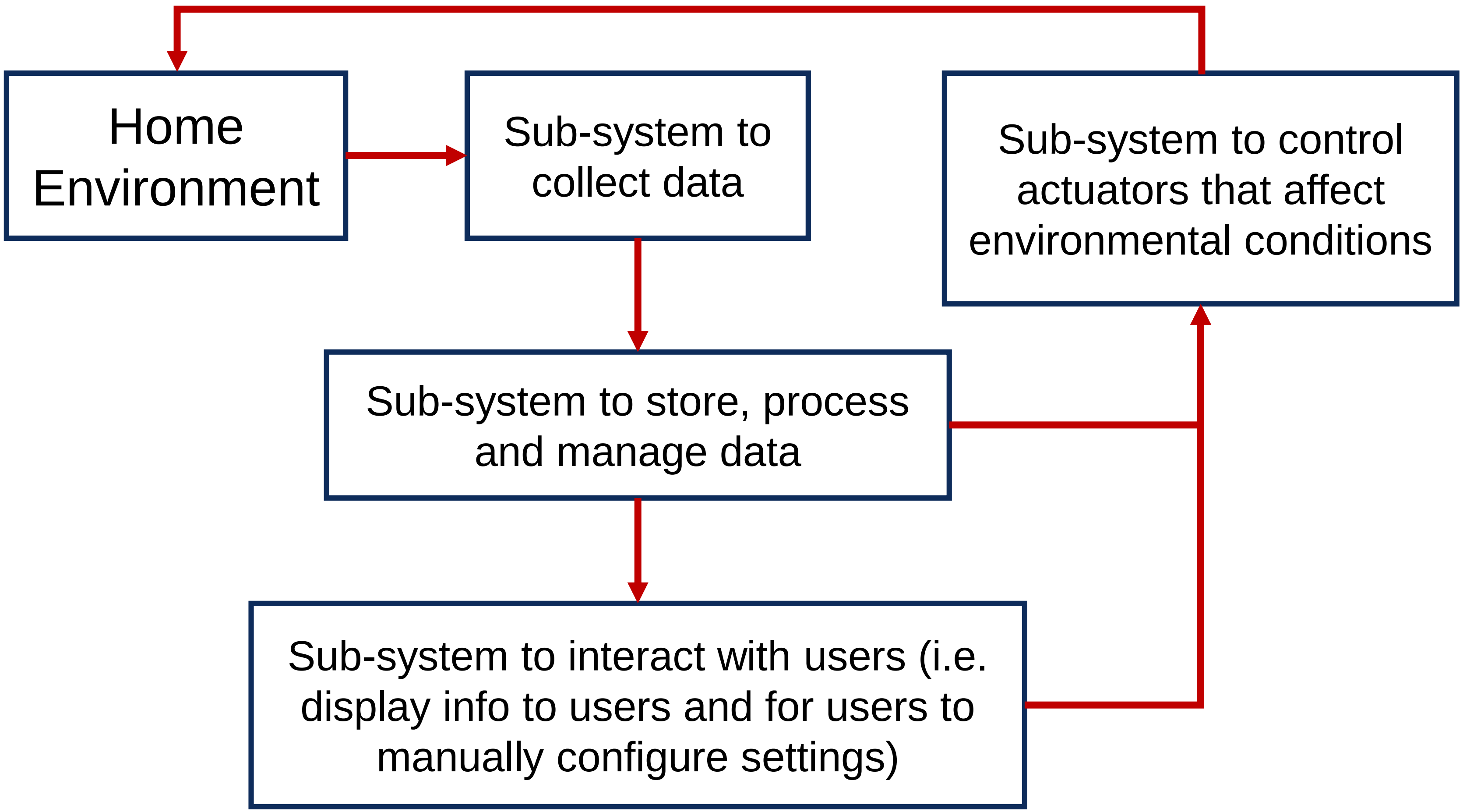
Step 1: Management Plan

Gantt chart planning for 6 months:



Case Study Example

Step 2: Concept of Operations



Case Study Example

Step 3: System-level Requirements

3 sub-systems required:

1. Layer 1 → Sensors & Actuators System (i.e. SAN)
2. Layer 2 → IoT Server System
3. Layer 3 → User Interface (UI) System

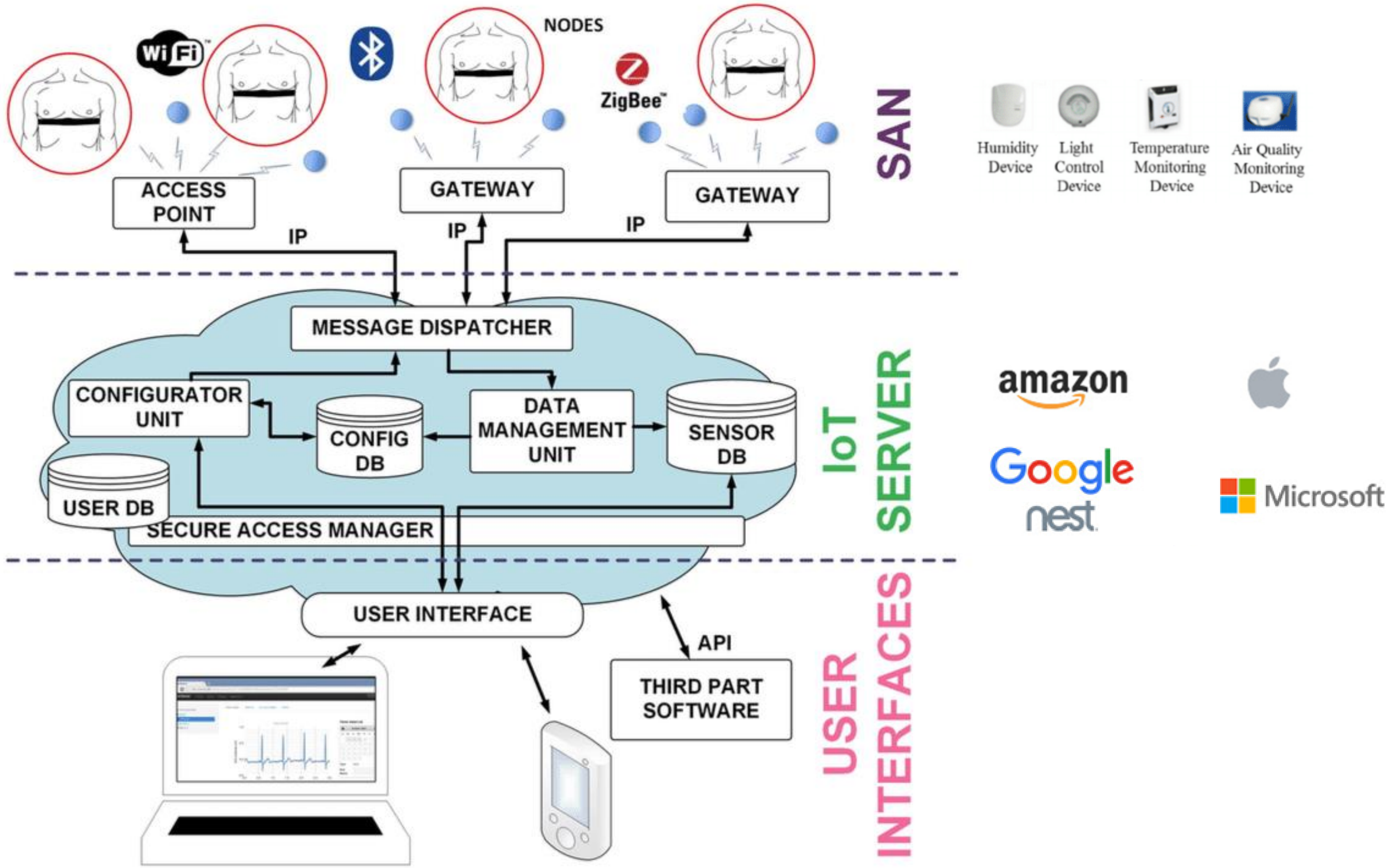
Step 4: Sub-system-level Requirements

The following components required for each sub-system:

1. Layer 1 – Temperature & Humidity sensors, Temperature & Humidity control devices, Light sensors, Light control devices, Air quality monitoring device, gateways/routers
2. Layer 2 – message dispatcher, sensor database, configuration database, user database, data management unit, configurator unit, secure access manager
3. Layer 3 – laptop, smart phones

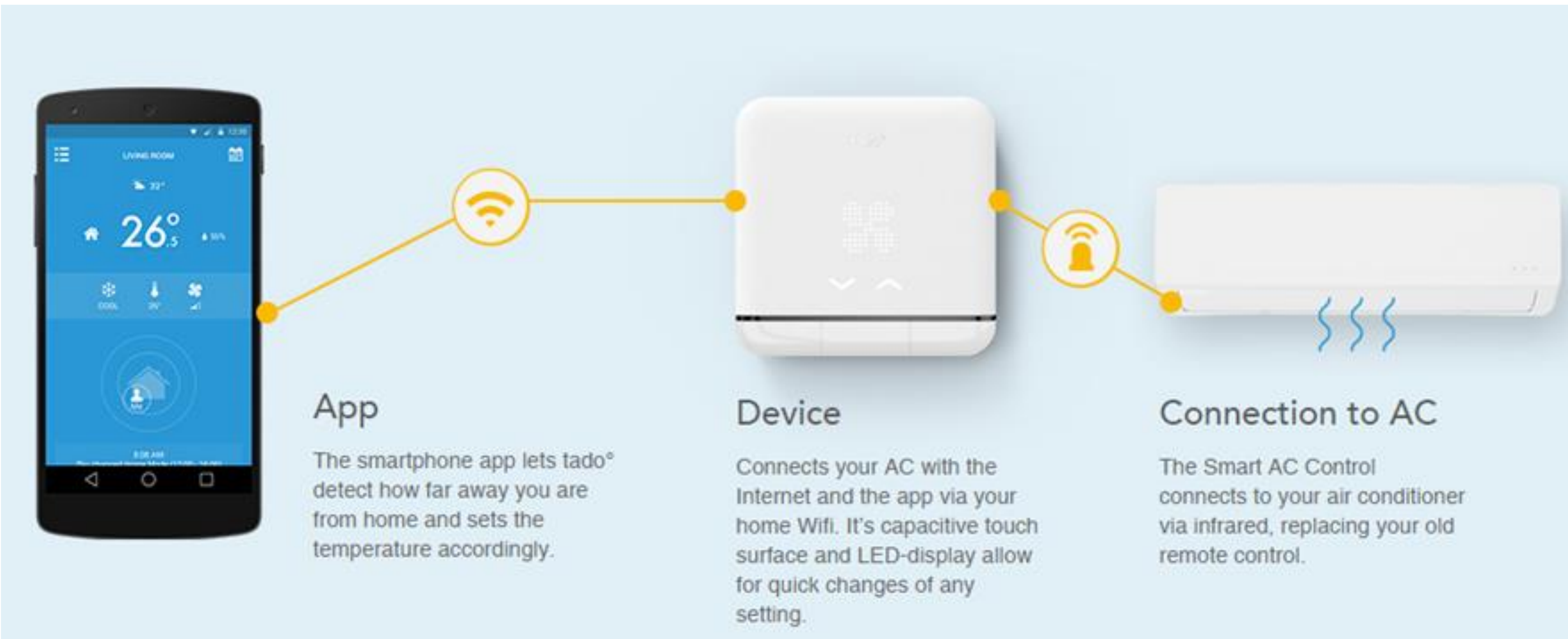
Case Study Example

Step 5: Detailed Design



Case Study Example

Step 6: Implementation (Temp & Humidity E.g.)



Case Study Example

Step 7: Verification

- **Component verification** – ensure individual components meet each respective requirement (e.g. Temperature & Humidity sensors, Temperature & Humidity control devices, Light sensors, Light control devices)
- **Sub-system verification** – ensure sub-systems meet their respective requirements (i.e. SAN, IoT Server, UI System)
- **System verification** – ensure the system as a whole meet the requirements (i.e. the integrated IoT system)

Case Study Example

Step 8: System Validation

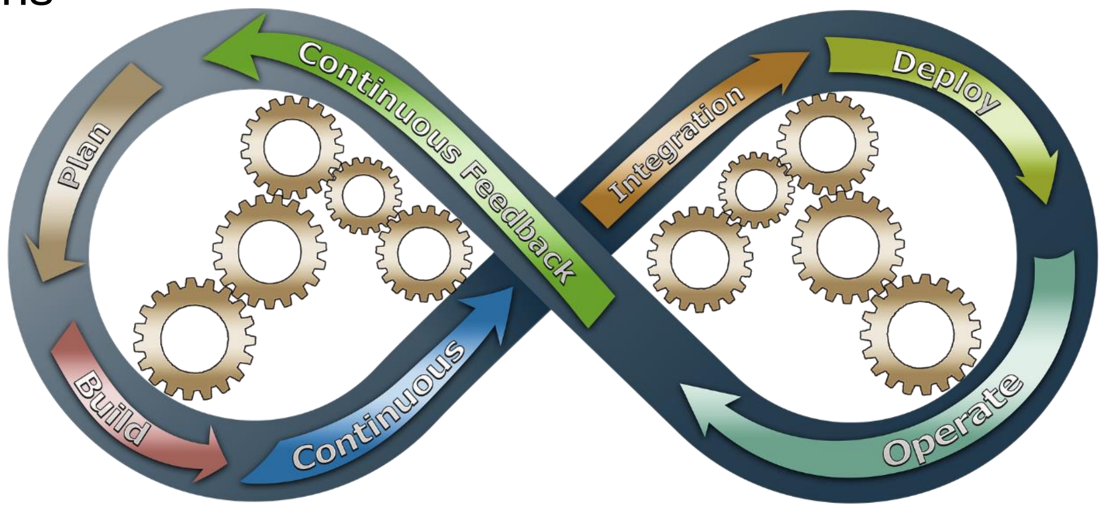
- **Pilot tests and surveys for user feedback**
- **Various possible test scenarios**
  - ➢ Various weather conditions (e.g. rain, sunny, cloudy, snowing, monsoon, thunderstorms, windy, hazy)
  - ➢ Different house types (e.g. Hdb, condo, private estates, studio apartments)

Case Study Example

Step 9: Operation & Maintenance

Plan after product launch:

- Continued monitoring to gather feedback from users
- On a lookout for new services by service-providers (e.g. cloud-related services) and for more affordable/quality sensors/actuators
- Annually revise product according to feedback after product launch; newly improved versions



<u>WORKSHOP:</u> DESIGNING A USE CASE FOR AUTONOMOUS SYSTEMS USING A HOLISTIC DESIGN APPROACH

Group Project 1

Let's assume that the idea of autonomous cars is totally new in the automotive industry. **And imagine you are a systems engineer being tasked by your boss to work on the development of a new product: <u>self-driving cars</u>.**

Given that the total budget for this project is $500,000 and the allowable development time is 10 years. You are expected to make other decisions on your own.

<u>Use the V-Model</u> to aid you in your product development process. You may refer to the previous case study example and the other module notes as a guide. **<u>The full details of the project requirements are stated on the next few slides.</u>** You are required to present your group work at the end of your discussions.

Group Project 1

Full details of the project requirements for each step (For the interest of time, focus more on Steps 2 to 6!):

1. Management Plan

- **Budget breakdown** among materials, software-related services, equipment and miscellaneous costs; **assume manpower is tapped onto existing employee pool**
- **Gantt chart** denoting project timeline from Step 2 to 9
- **Breakdown of necessary personnel** based on the product idea or identified problem, and allowable budget

2. ***Concept of Operations***

- **Simple block diagram or drawings to illustrate process flow**; specifics are NOT required at this stage (refer to IoT system case example)

Group Project 1

Full details of the project requirements for each step:

3. ***** System-level Requirements *****
   - **State all the sub-systems required**
   - NOT required to state any specifications for this project
4. ***** Sub-system-level Requirements *****
   - **State all the components required for each sub-system**
   - NOT required to state any specifications for this project
5. ***** Detailed Design *****
   - **Detailed block diagram which involves all sub-systems and components, and illustrate the process flow** (refer to IoT system case example)
   - **Specifics are required at this stage**; though NOT required to state the brand of the components

Group Project 1

Full details of the project requirements for each step:

6. *** Implementation ***

- Prototypes are <u>NOT</u> required for this project
- However, you are required to **do a rough sketch of the product (i.e. self-driving car) with the labelled components**

7. Verification

- Since there will not be any prototypes and specification requirements involved in this project, the verification stage only requires you to **draw a flow chart of the 3 verification stages**
- You have to reflect in your flowchart which step you are required to fall back to, in the event that any of the verification stage fails (e.g. if the component verification fails, the development team have to fall back to Step 5 and review the detailed design);
- **Refer to the V-Model as a guide**

Group Project 1

Full details of the project requirements for each step:

8. System Validation

- **Open discussion in class** to simulate test scenarios and user feedback
- **Poll by the group** to gauge on feasibility of your product

9. Operation & Maintenance

- You are **only required to state the plans for this stage** (refer to case study example as a guide)

Group Project 1

Group Project 1

Please upload all your works in Luminus with your names/student IDs as part of the title:

(e.g. DANIEL_KEN_TAN_Group_Project_1)

OR

(e.g. AXX_AXX_AXX_Group_Project_1)

End of Module 2

THANK YOU for your kind attention!

Main References

- https://en.wikipedia.org/wiki/Systems_engineering#Holistic_view
- https://en.wikipedia.org/wiki/V-Model#Specification_stream
- A Systems Engineering Approach to Regulating Autonomous Systems by David Paul Britton
- Systems Engineering Fundamentals by Department of Defense Systems Management College