Supporting Development of Complex System through Model-Based Systems Engineering (MBSE)

Dr. Oscar A Mondragon

Director MS in Software Engineering

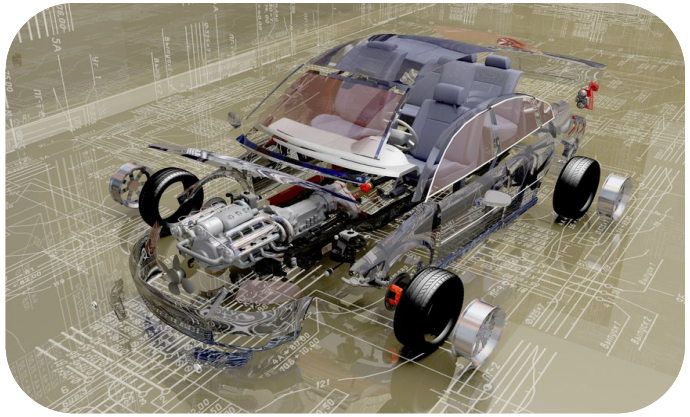
Computer Science Department

The University of Texas at El Paso

Challenges of Systems Development

From where does system design currently emerge?

emerges from pieces, rather than from **architecture**
➔ **systems** are:
breakable,
difficult & **complex** to **test** and **operate**



The pace of change

- Greater pace of change
- Reduce time to deliver solutions

Knowledge & Investment

- Lost at project lifecycle phase
- Development cost go up
- Late discovery of design problems

System complexity

Increased due to:
- Languages, technology,
- Global information flow.

Mission complexity growth

- Growing faster than our ability to manage it
- Inadequate specifications
- Incomplete verification.

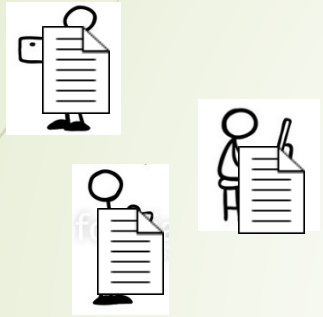
Demands of capability

Systems development has not kept pace with
the demands to deliver more capability in less time
➔ traditional methods and development teams often fail to deliver

Source: "Essentials of IBM Rational Rhapsody for Systems Engineers" course from IBM Corporation 2012 & INCOSE Vision 2025

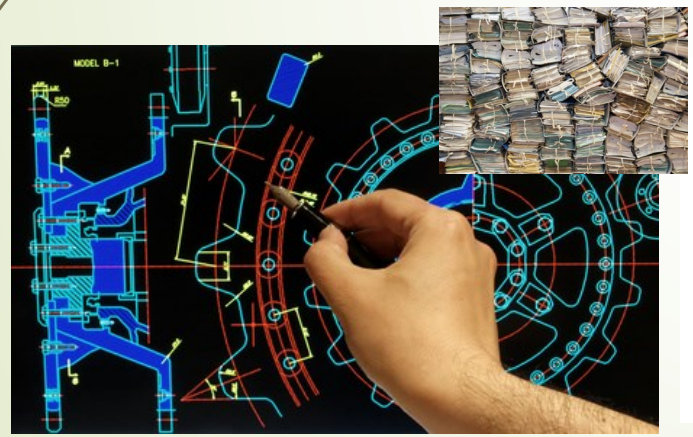
Document-driven communication

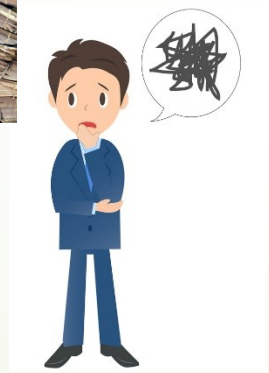
Development is largely document-driven.



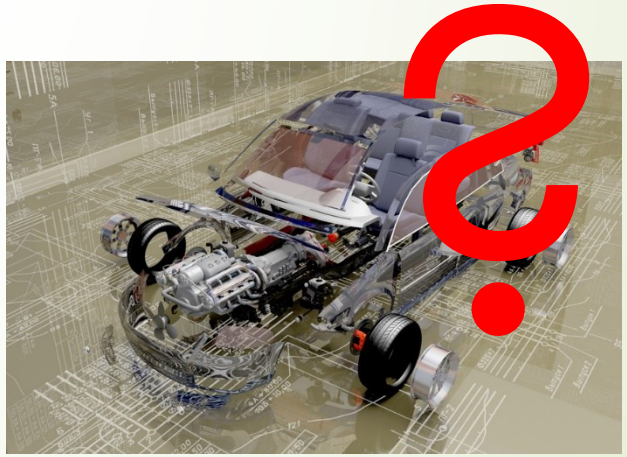
Documentation is developed based upon the needs of the customer.

Then submitted into a stack of documents for review and analysis.





Stack of documents is sent downstream
Architecture & Design are then created

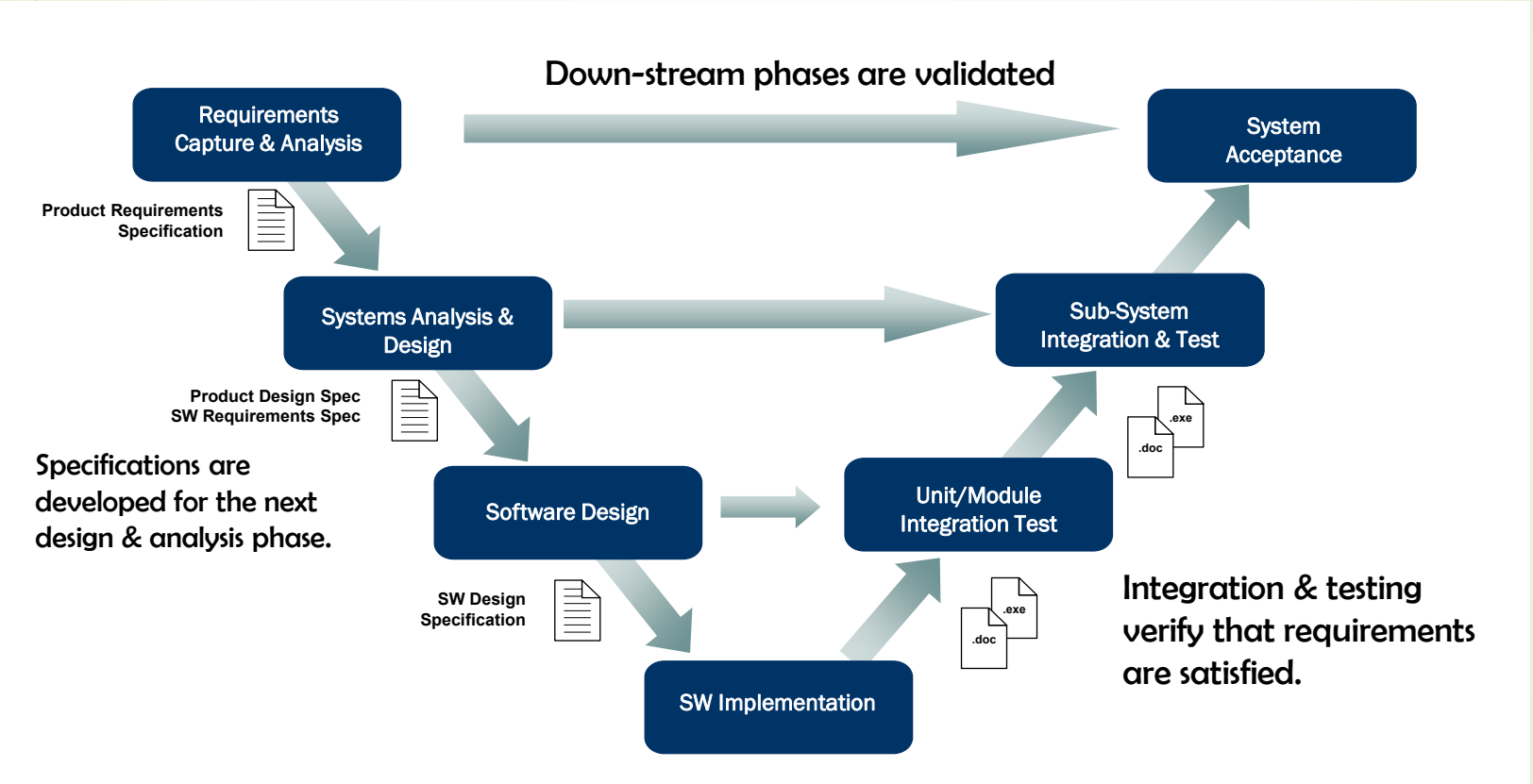


System design is accepted and built

But is the system correct?

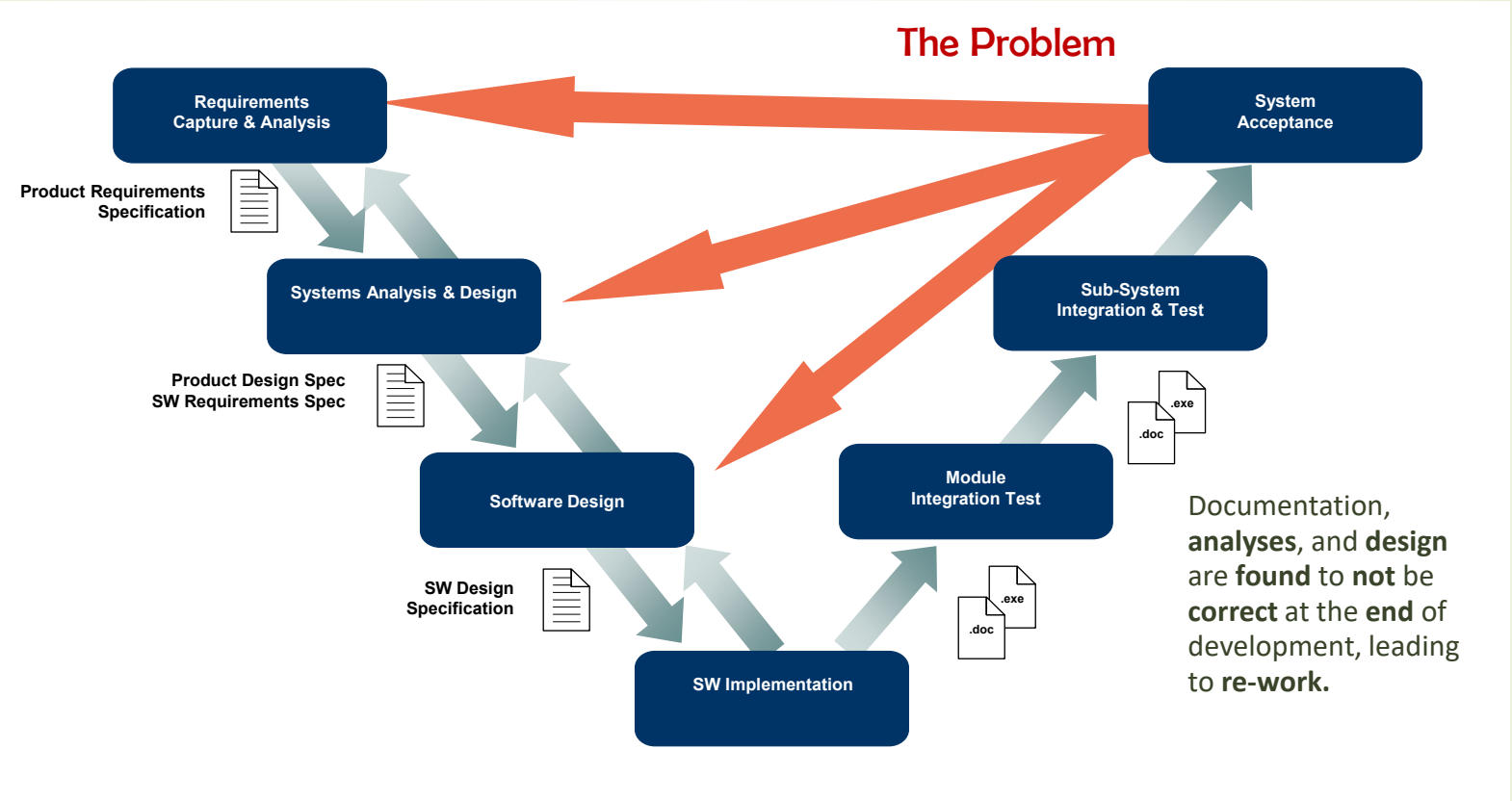
Document-driven development - how it should work …

Traditional development assumes design input is both fully and correctly defined.



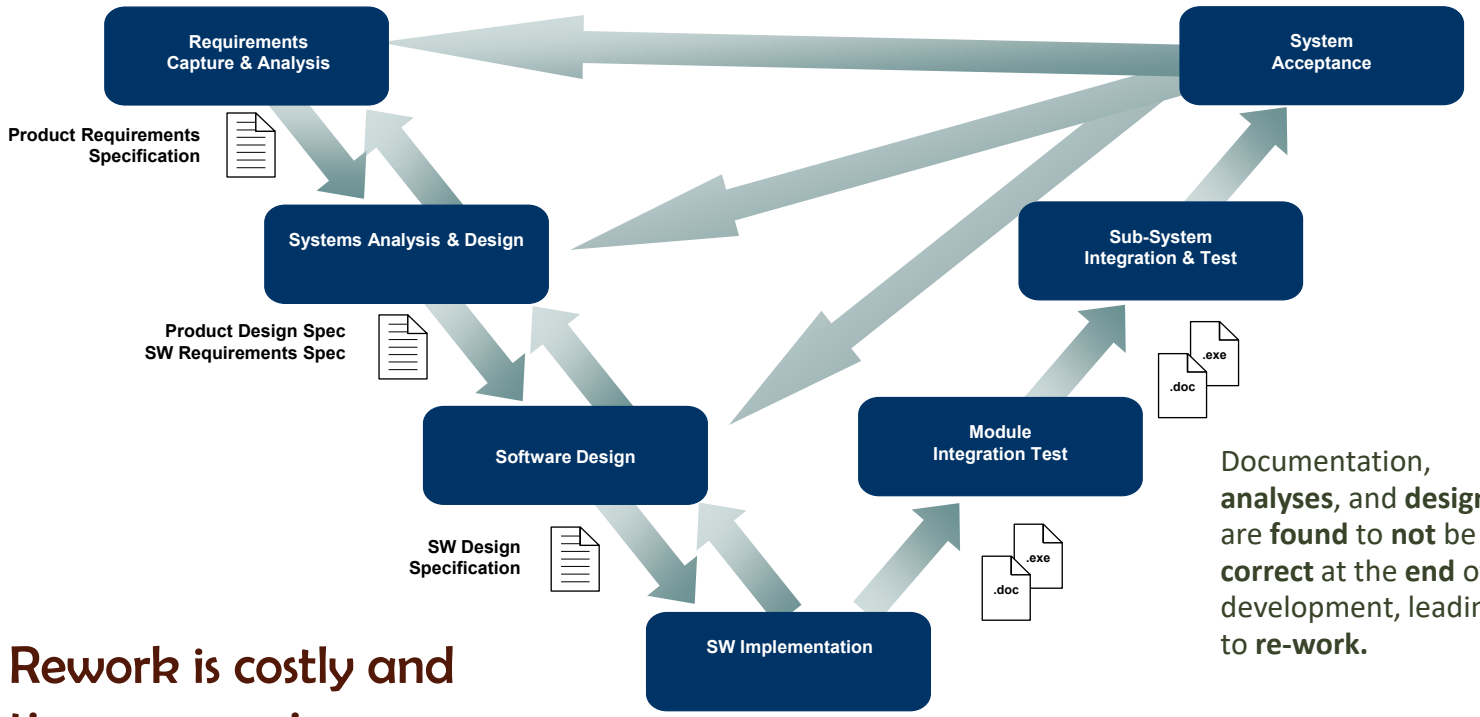
Source: "Essentials of IBM Rational Rhapsody for Systems Engineers" course from IBM Corporation 2012 & INCOSE Vision 2025

Document-driven development - how it should work …



Source: "Essentials of IBM Rational Rhapsody for Systems Engineers" course from IBM Corporation 2012 & INCOSE Vision 2025

The Problem



Documentation, **analyses**, and **design** are **found** to **not** be **correct** at the **end** of development, leading to **re-work.**

Rework is costly and time consuming

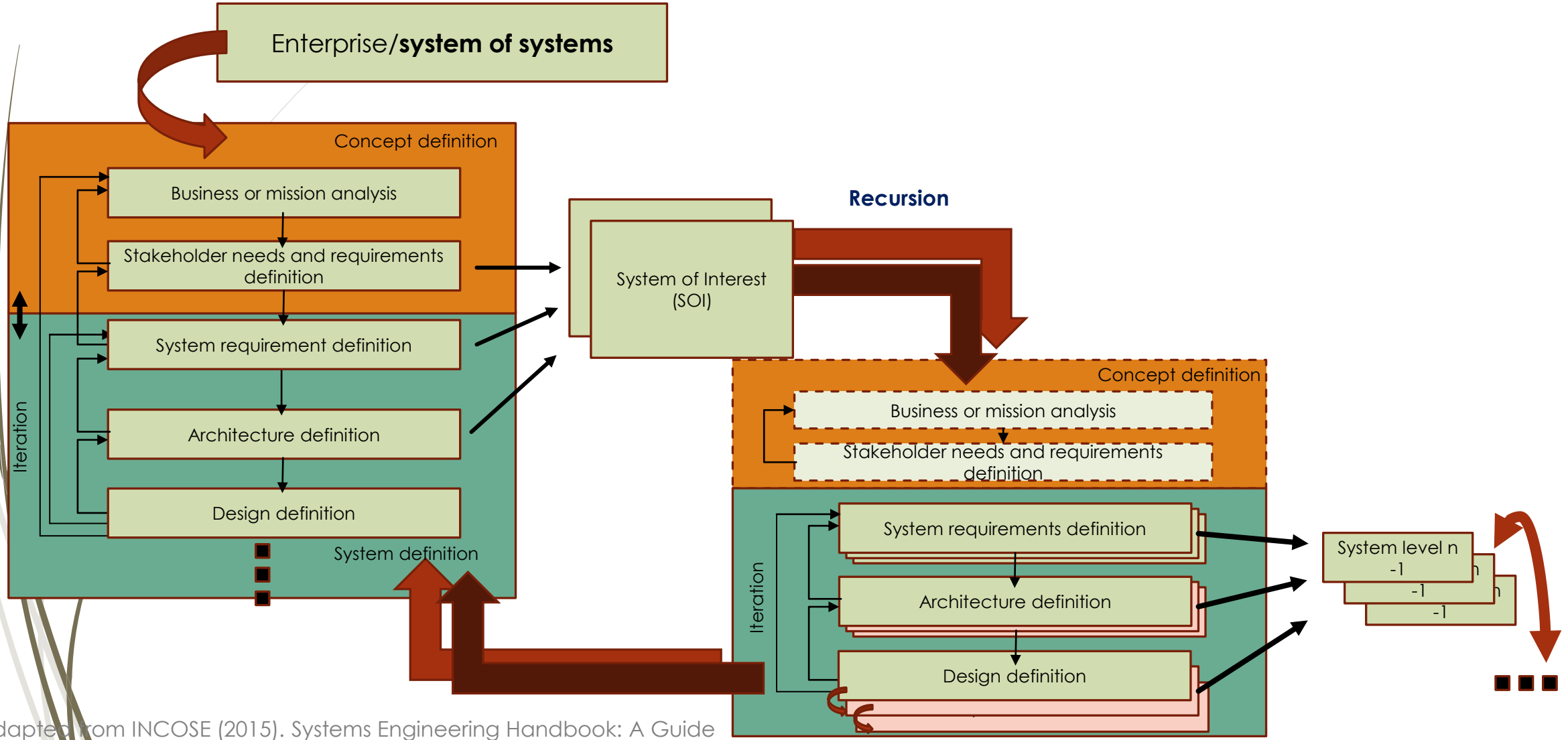
- schedule delays
- **Need** for **change** is typically **discovered late**
- **changes** not **reflected** up or down

Common Issues

- Documents are not updated
- Documentation inconsistent, incorrect, & abandoned
- All the effort and discipline is wasted.

Source: "Essentials of IBM Rational Rhapsody for Systems Engineers" course from IBM Corporation 2012 & INCOSE Vision 2025

Coupling with Complexity: iteration and **Recursion**

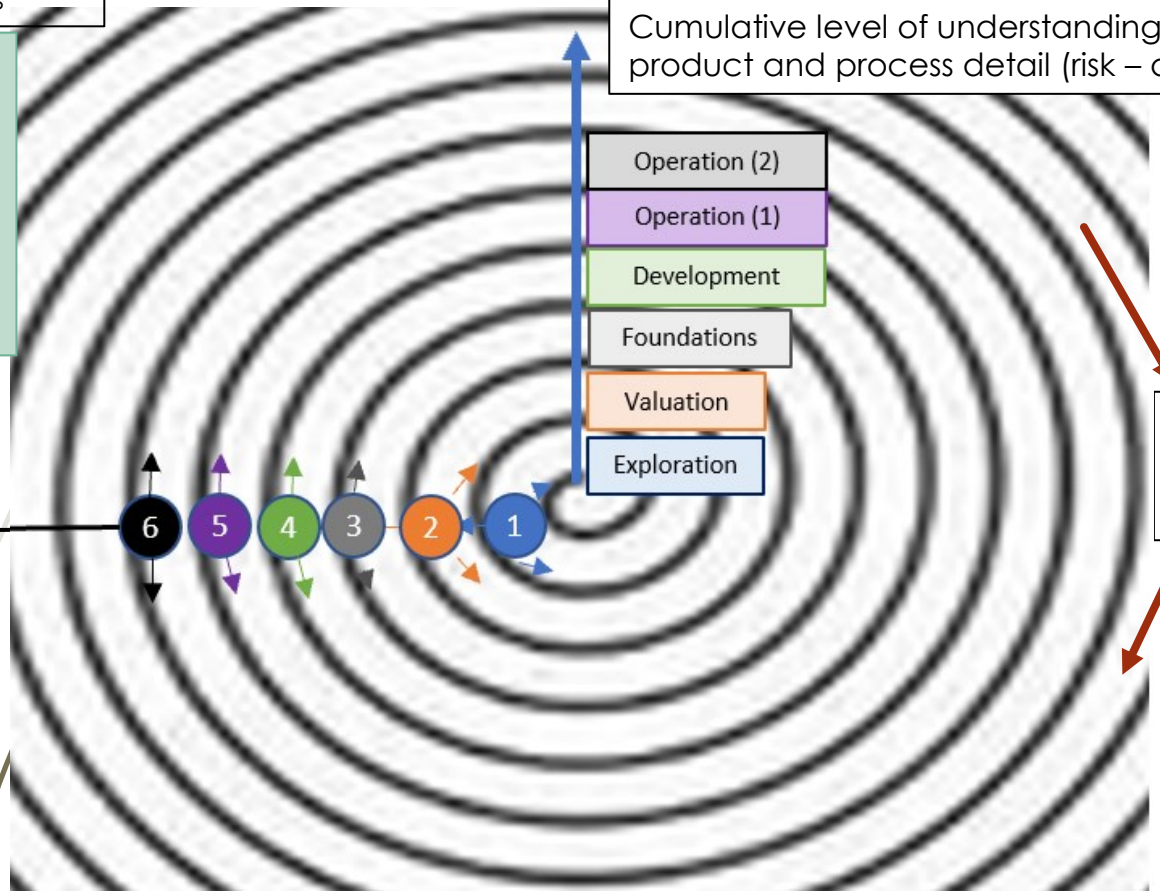


Adapted from INCOSE (2015). Systems Engineering Handbook: A Guide for System Life Cycle Process and Activities (4th ed.). Figure 3.5 on page 33.

Coupling with complexity: **Spiral Methods**

Risk – based stakeholder commitment review points

Spiral Model



Cumulative level of understanding, product and process detail (risk – driven)

1. Exploration commitment review
2. Valuation commitment review
3. Foundations commitment review
4. Development commitment review
5. Operation (1) and development (2) commitment review
6. Operation (2) and development (3) commitment review

Concurrent engineering of products and processes

Activities	
Concurrent risk – and – opportunity – driven growth of system understanding and definition	Initial scoping
Evaluation of evidence of feasibility to proceed	Feasibility evidence
Stakeholder review and commitment	…

Risk – based decisions

Evidence – based review content
- A first – class deliverable
- Independent expert review
- Shortfalls are uncertainties and risks

Adapted from INCOSE (2015). Systems Engineering Handbook: A Guide for System Life Cycle Process and Activities (4th ed.). Figure 3.10 on page 37.

Collaboration in text … in action

Ok this is how it should work:

The Device Manager sends a request to the Transaction Manager – that will put the Transaction Manager into a Checking State, from what it was before which was Idle.

The Transaction Manager then sends a message to the Account manager to get authorization and waits for a message to come back.

If the authorization doesn't come back within 2 seconds the Transaction Manager sends a denied message back to the Device Manager. The Device manager will have started in an Idle state but after it gets the confirmation it should move to a state where it waits until it gets the authorization. If it instead gets a denied message then it should move back to being idle.

All this should happen in less than 5 seconds.

Engineer 1

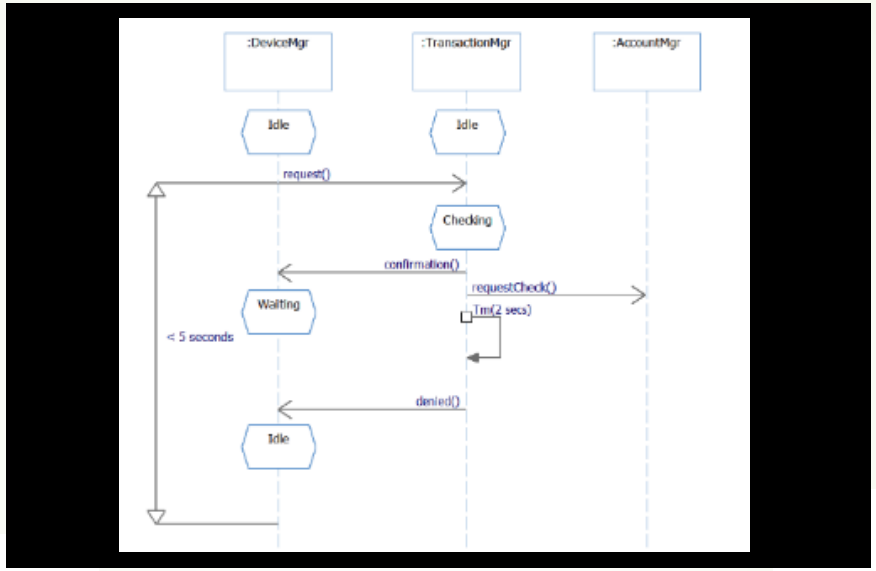
Communicating through only **text can be difficult** when trying to impart knowledge and intent of a system.

Engineer 2

Source: "Essentials of IBM Rational Rhapsody for Systems Engineers" course from IBM Corporation 2012 & INCOSE Vision 2025

Collaboration using diagrams

Communicating through **visual representations** is a much **more natural** and intuitive method.



Here, look at this Sequence Diagram.

Engineer 1

The above model has:

- well defined **syntax** & **semantics**
- reduces ambiguity.

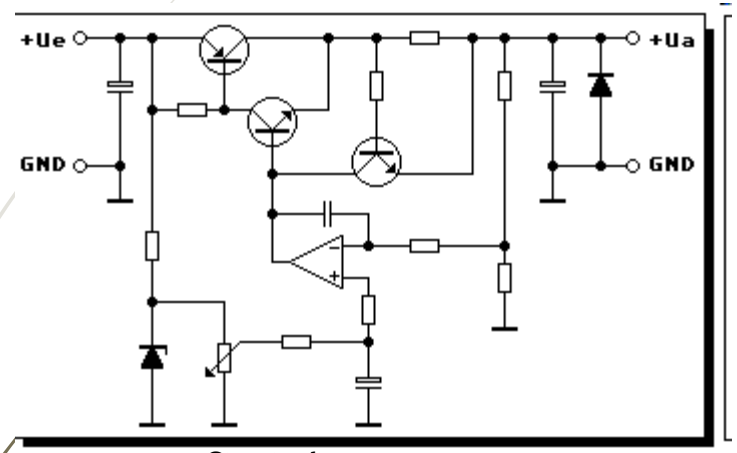
Ahhh, now I see!

Engineer 2

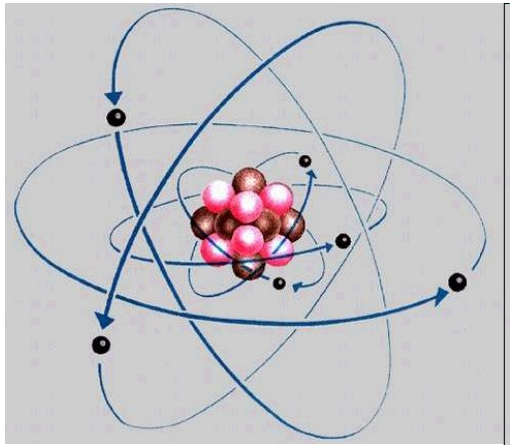
Source: "Essentials of IBM Rational Rhapsody for Systems Engineers" course from IBM Corporation 2012 & INCOSE Vision 2025

Graphical Abstraction

Graphical abstractions are used different fields,
They represent concepts.

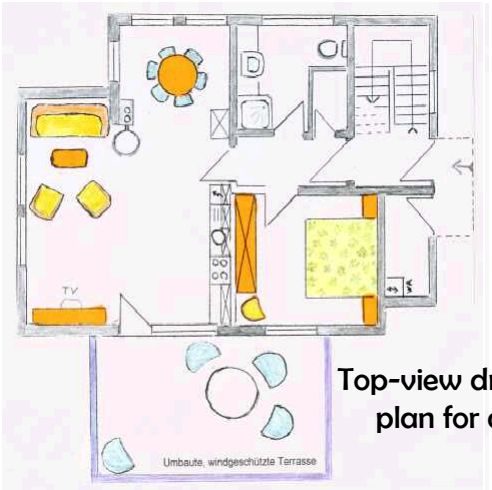


Circuit diagram

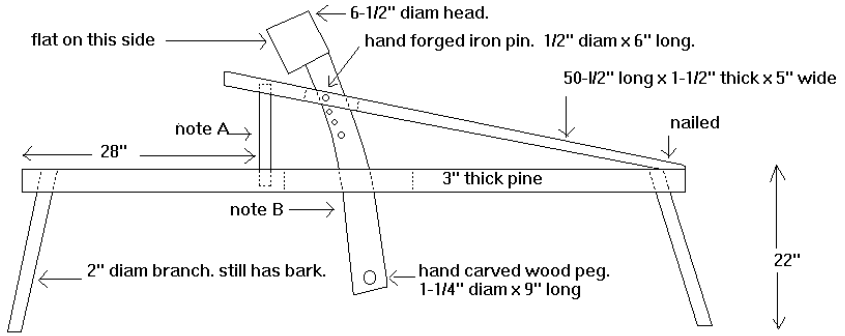


Abstract model of an atom

Computer-aided Manufacturing diagram for the face of a calculator

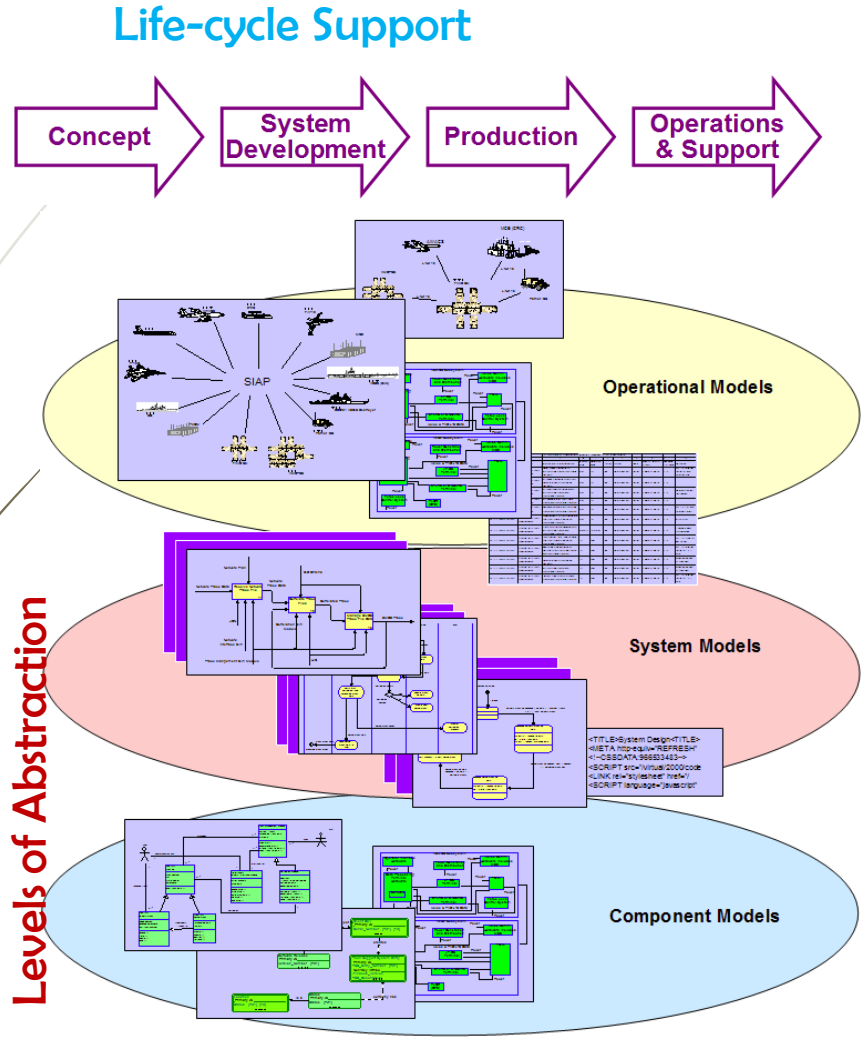


Top-view drafting of a floor plan for a living space



Side-view draft of a mechanical device

Source: "Essentials of IBM Rational Rhapsody for Systems Engineers" course from IBM Corporation 2012 & INCOSE Vision 2025



Model-based systems engineering

- formalized **application** of **modeling** to **support** system **requirements**, **design**, **analysis**, **verification**, and **validation** activities
- Done **throughout life cycle** phases

INCOSE Systems Engineering Vision 2020 (2007)

More simply…

MBSE is the formalization of the practice of systems development through the use of models

Source: INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc

Model-based systems engineering

What is its scope?

To integrate with multiple modeling domains across the life cycle from system of systems to component.

What is its goal?

Facilitate results in quality / productivity improvements & lower risk

For the specification of a product, MBSE enhances the ability to:

- Capture information
- Analyze data
- Share knowledge
- Manage complexity

Source: INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc

Characteristics of MBSE

Model-centric, not Diagram-centric

- An underlying model of the system is required, not just several diagrams thrown together.
- A common repository is maintained for the model.
- All team members have access to the model.
- One version of the truth is maintained across all views

Views are generated from the model

- Consistency is maintained as changes occur
- Views are tailorable to the needs and understanding-level of the audience.

Unambiguous notation

- Syntax and Semantics for each model
- Omissions within the design are found

Complete, Query-able, Virtual System Prototype

- A prototype that can handle all aspects of Systems Engineering
- Capable of acting as a virtual prototype

Example of SE Domains

Source Requirements Domain

Behavior Domain

To use MBSE, all defined SE domains must:

- be integrated
- have connectivity
- be coherent

V & V Domain

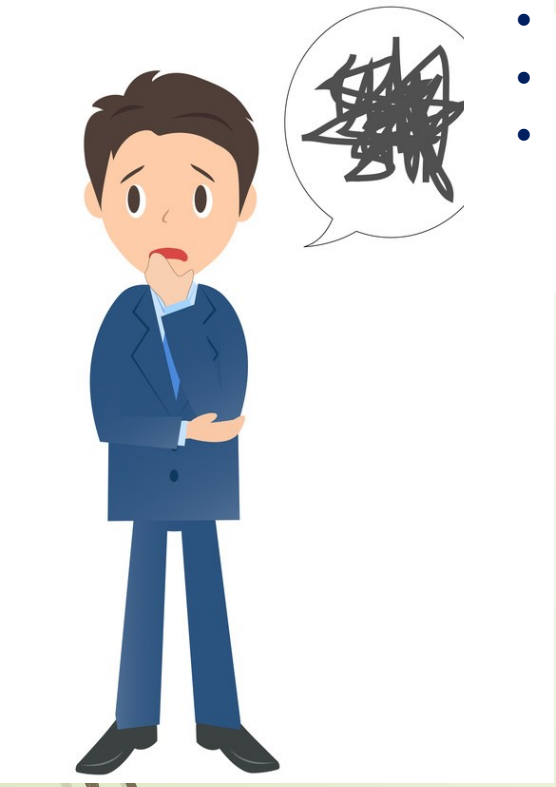
Architecture Domain

Adapted from Webinar by Vitech

Model-centric, not diagram-centric

"But don't we draw diagrams???"

- Model-centric approach develops a central model for the system-of-interest
- Have certain aspects represented by diagrams.



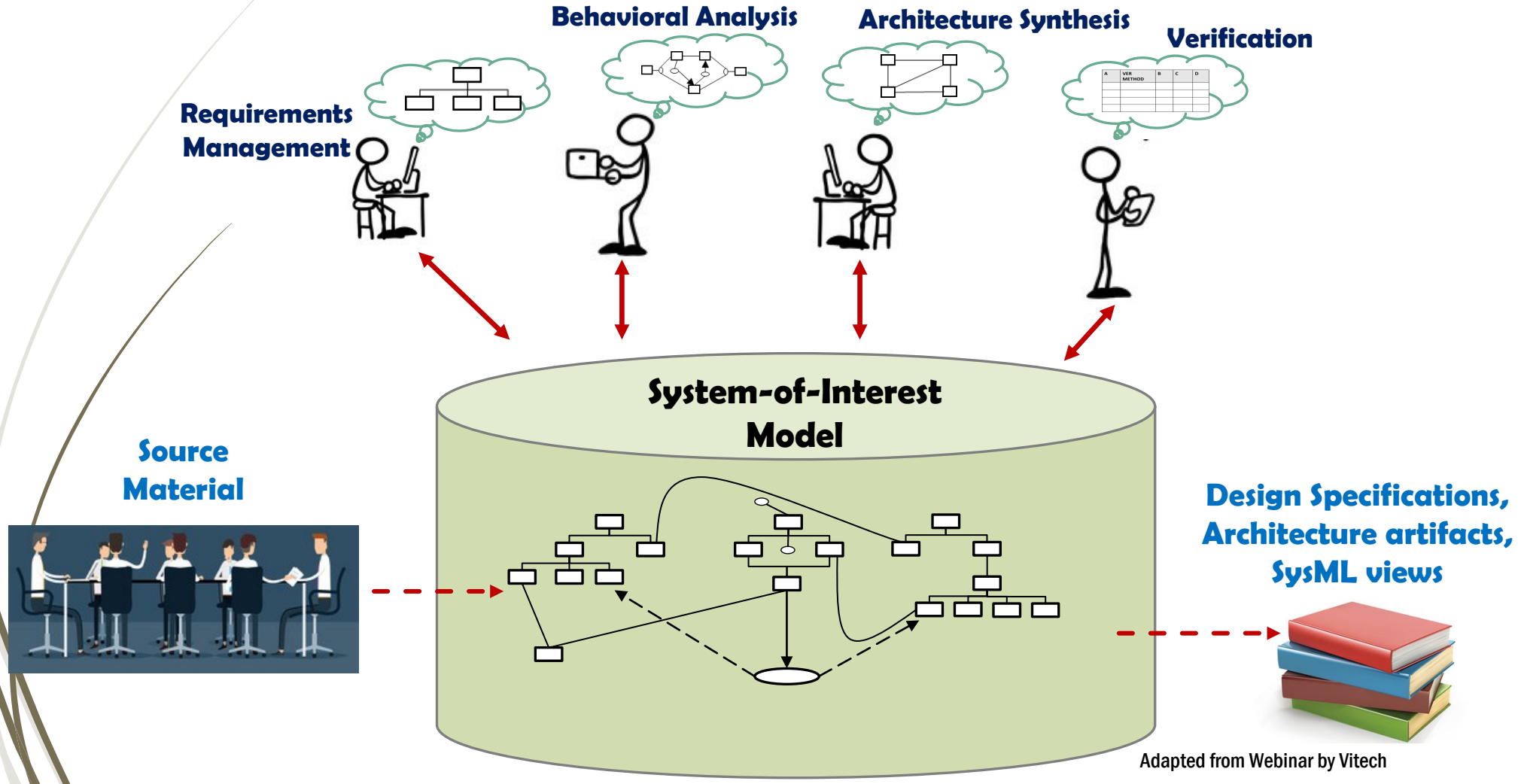
- **So**, by creating several diagrams of the system
- Throwing them together
- ➔ it **does not** constitute a model

It **takes** an **organizational effort** to **create** an **inclusive repository** for the **knowledge** contained in a **model**.

Adapted from Webinar by Vitech

Model-centric, not diagram-centric

Each member of a project must overcome the "silo" effect and contribute their experience in transforming and incorporating the source material into the model.



Adapted from Webinar by Vitech

Unambiguous Notation

… What, … A or B and C

Ambiguous Notation

- Anything that can lead to multiple interpretations
- wastes both time and money
- misunderstand leads to multiple versions of the "Truth"
- natural language are prone to ambiguity.

Examples of ambiguous notation that exists:

- "….part of…"
- "…kind of…"
- "….associated with…"
- "…depends on…"

Promote continuity, not Ambiguity

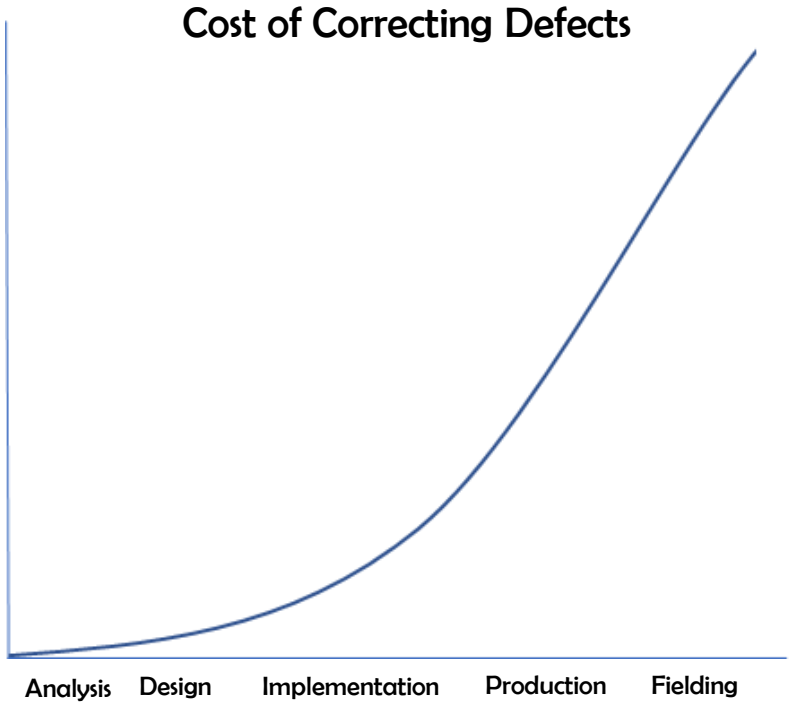
- Use models with specific syntax and semantics
- Ensure clarity among design teams

Adapted from Webinar by Vitech

Impact of Unambiguous Notations

Defect Identification

- Incomplete interfaces
- Unallocated behavior
- Unimplemented requirements
- Unverified requirements
- Unaddressed risks
- Undocumented elements



EXAMPLE

Is the requirement correct? ➔

"The automated teller system shall respond to the user's query within a reasonable amount of time."

"reasonable amount of time" is ambiguous
If not caught early ➔ lot of rework later

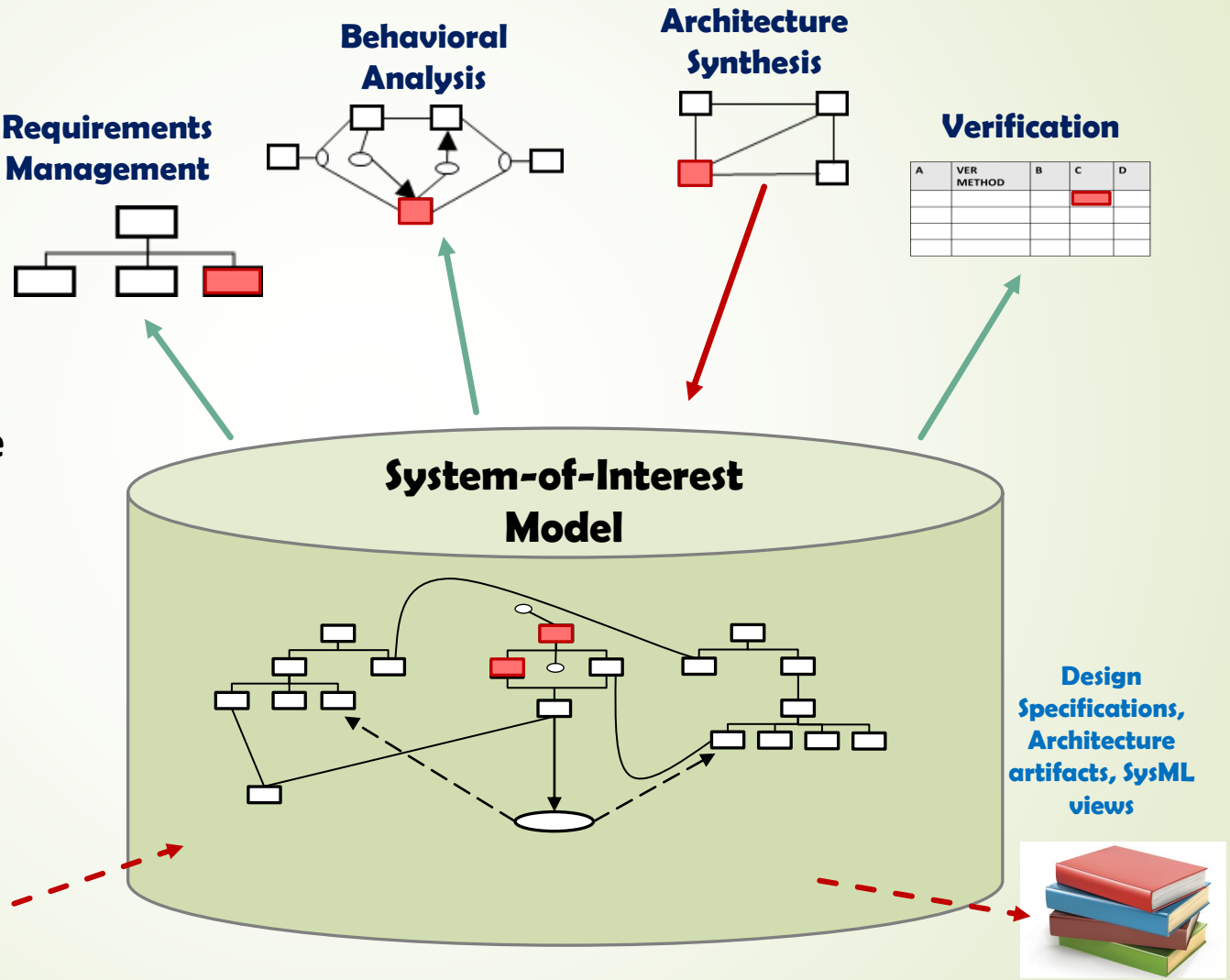
Adapted from Webinar by Vitech

Views Generated from the Model

Change control

Model-centric approach, viewpoints must be generated from the model

- **ensures** embedded **knowledge** is **consistent** and **correct**.
- **Changes** by developers **in one view** ➔ **reflected** across **all** the **model**



Adapted from Webinar by Vitech

Benefits of MBSE

Improved communications

Among all stakeholders

Customer, Program management, Systems engineers, Hardware and Software developers, Testers, and Specialty Engr

Ability to manage system complexity

- Enabling a system model to be viewed from multiple perspectives
- Analyze the impact of changes

Improved product quality

- Providing an **unambiguous** and **precise model** of the system
- Model can **be evaluated** for **consistency**, **correctness**, and **completeness**

Enhanced knowledge capture and reuse

- Capture **information** in **standardized** ways
- Leverages **abstraction** mechanisms **inherent** in **model-driven** approaches.
- **Lowers** maintenance **costs** to **modify** the **design**

Teach and learn SE

Provides a clear and unambiguous representation of the concepts

Source: INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc

MBSE methodologies

"Methodologies"

A methodology can be defined as the collection of related processes, methods, and tools used to support a specific discipline (Martin, 1996).

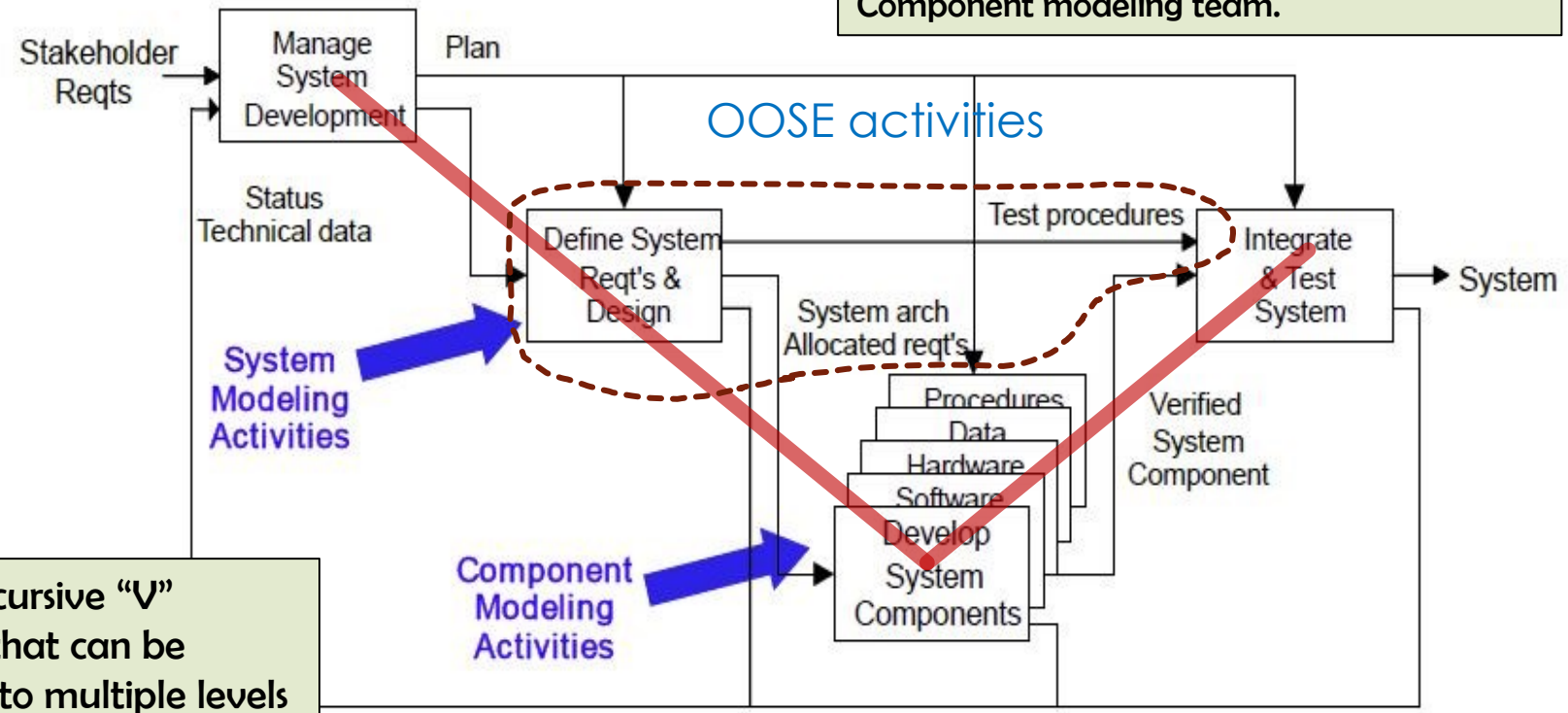
MBSE methodology

The collection of related processes, methods, and tools used to support the discipline of SE in a "model-based" or "model-driven" context (Estefan, 2008).

Source: INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc

OOSE - System Development Process

Integrated Product Development (IPD) is essential to improve communications between the Systems modeling team and the Component modeling team.

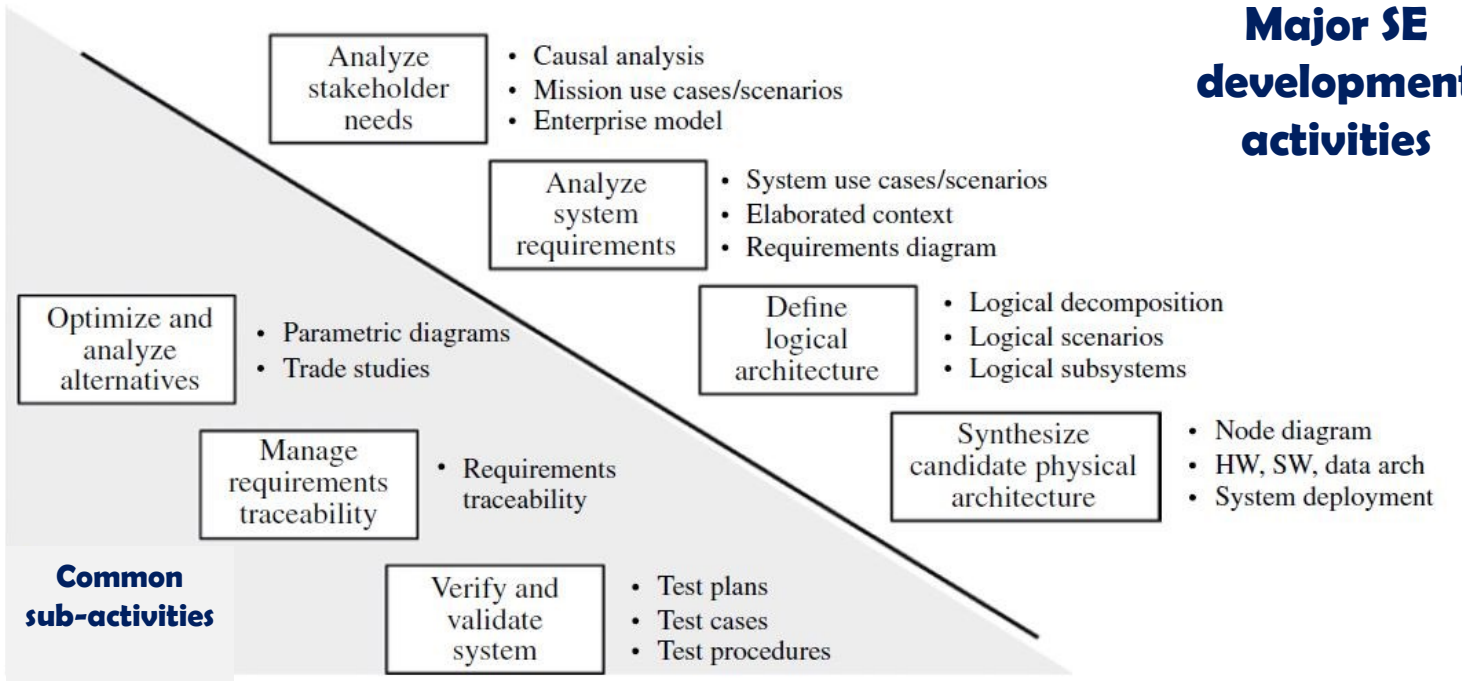


It is a recursive "V" process that can be applied to multiple levels of the system hierarchy.

Source: INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc

System Modeling Activities – OOSEM

Integrating MBSE into the SE Process



The system requirements and design process is decomposed into the OOSEM high-level activities depicted above.

Source: INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc

Summary

Model-centric vs Document-centric...
(Click here)

Model-centric systems engineering is to replace document-centric approaches

MBSE involves...
(Click here)

Formalization of practice of systems development through models

MBSE helps facilitate...
(Click here)

- **Communications among stakeholders**
- **Management of complexity**
- **Support for consistency & error checking**

A **model-centric** approach

- Removes ambiguity from a project
- Ensures one version of the "truth"
- One repository used by all members
- build multiple views from the model

References

- INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc, ISBN: 978-1-118-99940-0
- IBM Corporation 2012. *Essentials of IBM Rational Rhapsody for Systems Engineers v7.6.1* course
- Vitech Corporation. 2013. *Vitech Webinar 10/2/2013*

Dr. Oscar A Mondragon

Director MS in Software Engineering

Computer Science Department

The University of Texas at El Paso