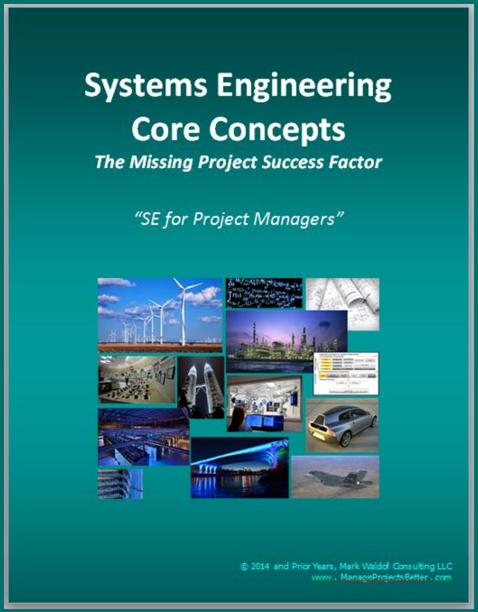
Systems Engineering Core Concepts

The Missing Project Success Factor

"SE for Project Managers"



This set of slides is a subset of an introductory course on systems engineering that provides a basic understanding of the "what" and "why" of SE.
A full course and a skills building workshop also exist.

The current version of this seminar can be found at the URL below.

Mark Waldof Consulting LLC
www.ManageProjectsBetter.com
owner@manageprojectsbetter.com

Systems Engineering Core Concepts

Condensed Seminar Outline

- SE Introduction
- What Is Systems Engineering
- Context and Role of SE
- System Successes and Failures
- What Goes Wrong with Projects
- What is a System?
- System Characteristics
- System Dimensions
- System Stakeholders
- System Variables
- Types of Projects
- System Development Models
- System Lifecycle
- System Development Process
- Critical Initial System Dev Steps
  - Introduction
  - Stating the Problem or Need
  - Root Causes
  - Operating Models
  - Stakeholder Needs
  - Alternative Solutions
  - Solution Boundaries
  - Alternative Trade-Off Criteria
  - Alternative Comparison and Selection
  - Describe Solution and Feasibility Check
  - System Requirements
  - System Design
- Capturing Systems Information
- The Role of the Systems Engineer
- Conclusion
- REFERENCE – System Documentation

Systems Engineering

Let’s Start By Looking at Situations Where Systems Engineering is Needed

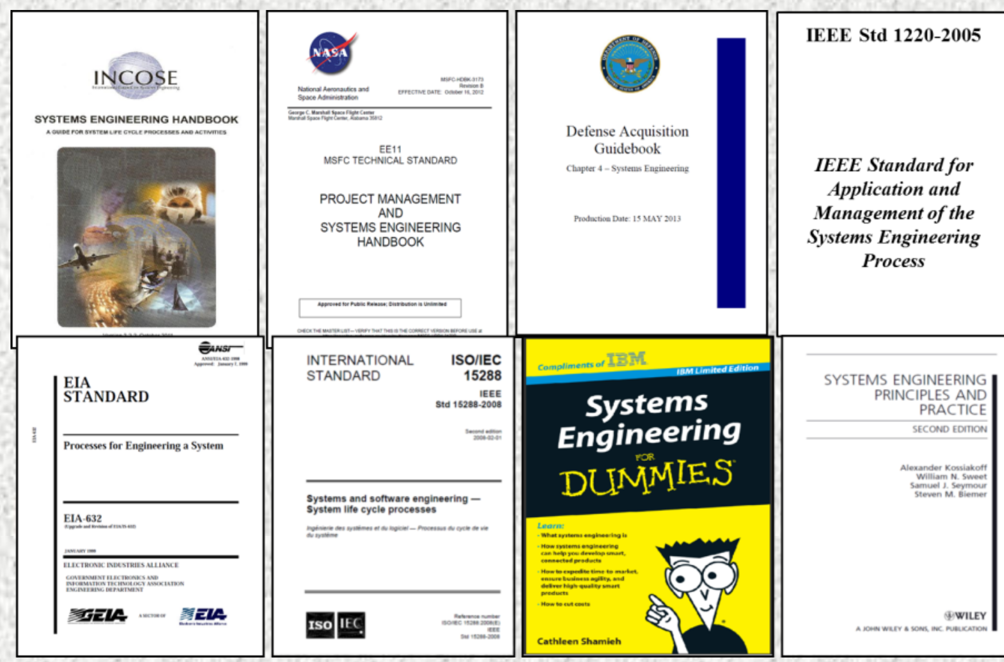
Systems Engineering is the Means to Resolve These Types of Situations and More

What is

Systems Engineering?

Systems Engineering References

- Multiple SE publications exist
- INCOSE is a recognized source of systems engineering information



What Is Systems Engineering?

Systems Engineering is an engineering discipline that focuses on understanding the problem or need, reviewing alternative solutions and selecting the solution that best meets all requirements.SE is designing the solution at a system level that meets the requirements and that forms the basis for lower level developments. SE addresses all stakeholder needs, all system lifecycle phases and all dimensions of a system. Following system level design, systems engineering technically manages the subsequent system component development, integration, system level testing, delivery and deployment.

What Is Systems Engineering?

One Example

Building Architect

- Collecting Requirements
- Resolving Conflicting Constraints
- Overall Building Design
- Component Requirements
- User and Energy Considerations
- Codes, Rules and Regulations
- …(other “architectural” topics)….

Systems Engineer

Construction Site Manager

- Project Plan
- Schedule Management
- Resources Management
- Team Communication
- Periodic Status Reviews
- Issues and Corrective Actions
- …(other managing topics)….

Project Manager

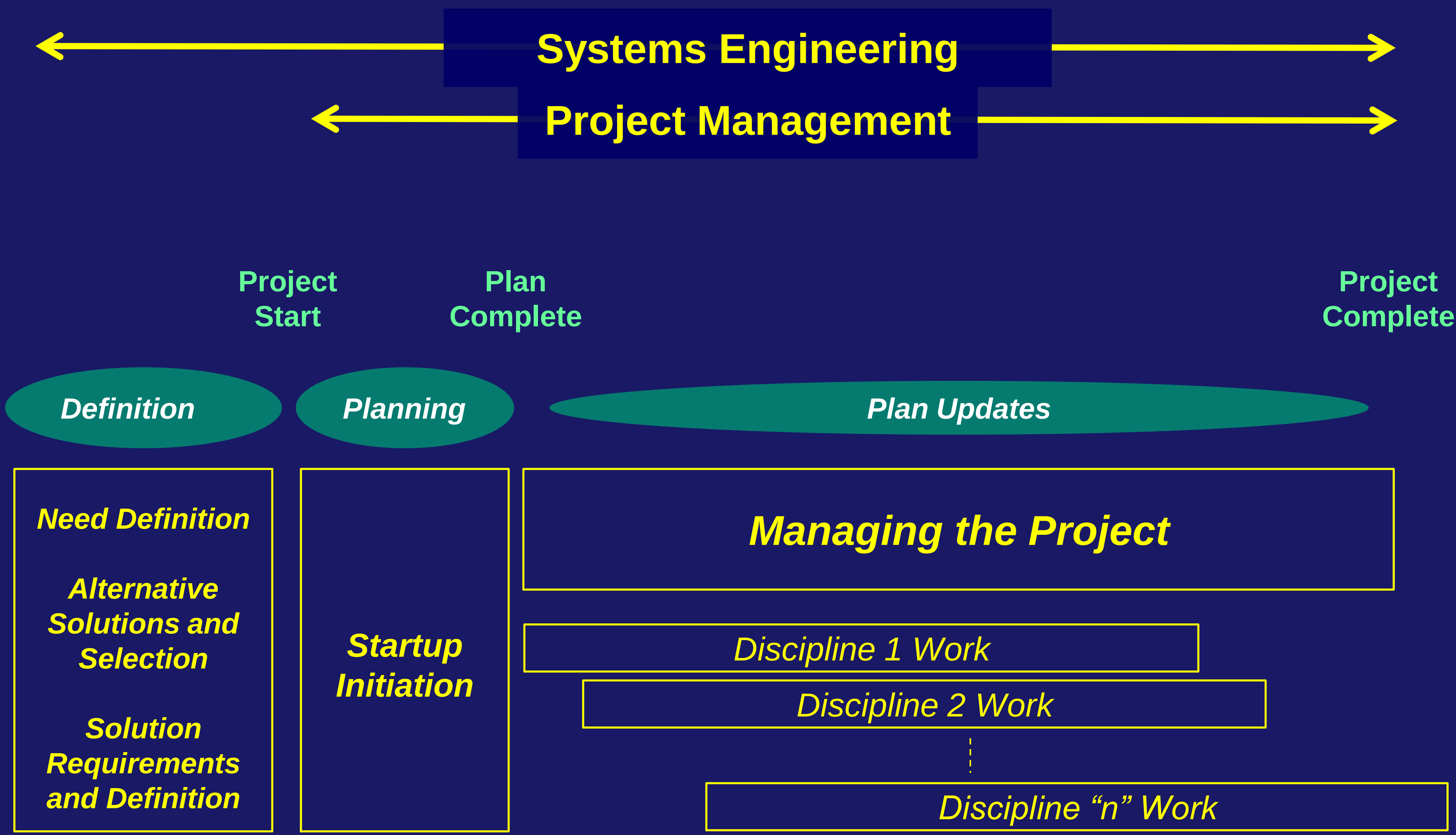
“Systems Engineering”

is the

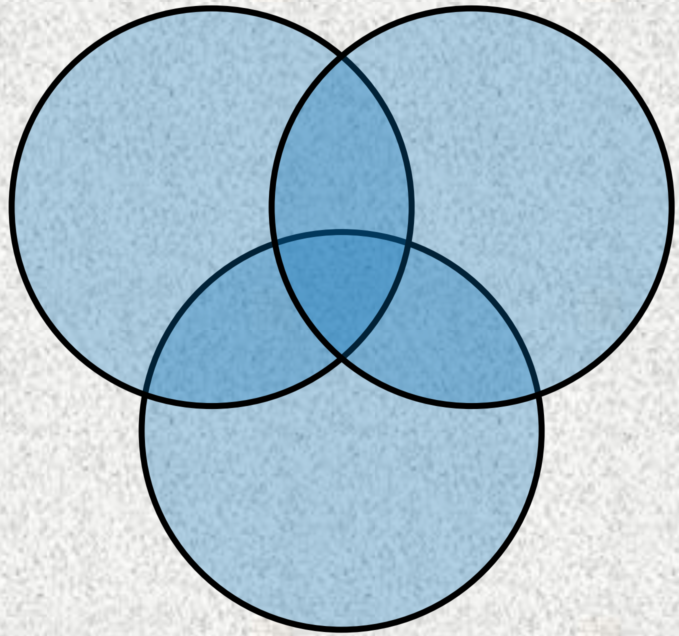
“Engineering of Systems”

The Context and Role of Systems Engineering

Notional SE and PM Role Timeframe



Notional Roles



Systems Engineering

- Need Definition & Validation
- Alternative Solutions
- Solution Selection
- System Requirements
- System Level Design
- Component Requirements
- Development Oversight
- Change Management
- Technical Risk Management
- Component Testing
- System Integration
- System Testing, Validation and Verification
- System Follow-On Support
- System Enhancements

Development Disciplines

- Support for Requirements
- Component Design
- Component Build
- Component Test
- Integration Support

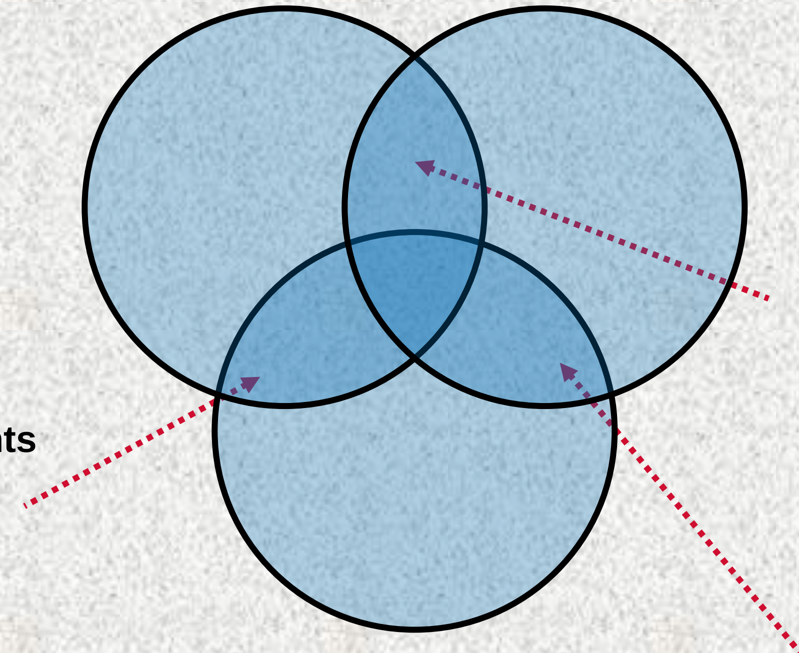
Project Management

- Client Management
- Proposal Generation
- Project Definition
- Plans
- Team Building
- Stakeholder Communications
- Project Controls
- Resource Management
- Risk Management
- Team Management
- Supplier Management

<u>Notional</u> Overlapping Roles

Systems Engineering

Project Management



- System Level Requirements
- Risk Management
- Managing Changes
- Oversight, Monitoring

- Requirements Management
- Budget Management
- Schedule Management
- Risk Management
- Resource Management
- Supplier Management

Development Disciplines

- Requirements Management
- Tracking Technical Performance
- System Integration & Test

Note that the SE role may be performed by a dedicated individual, by an individual that is also performing other roles or by a team

Systems Engineer

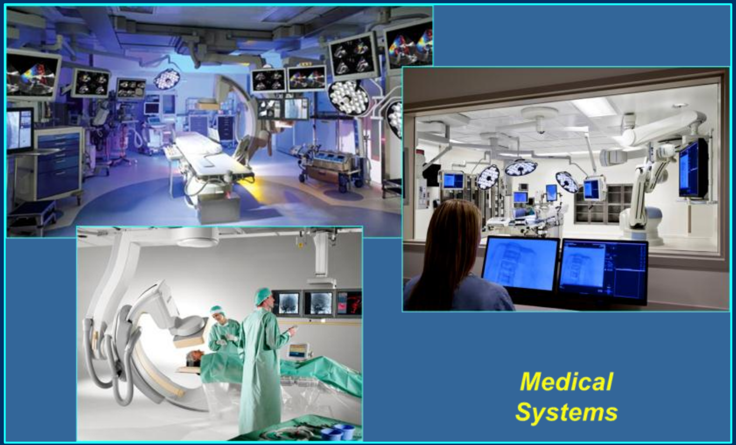
<u>Notional</u> Knowledge and Skill Areas

System Successes

Things That Work Well

These Systems Are…

- Valuable to Society
- Reliable
- Efficient
- *….(hold many other advantages)….*



System Failures

Things Fail

These Systems Failed for Multiple And Different Reasons…

- Poor Operator to System Interfaces
- Poor System Procedures
- Poor Backup Plans
- Bad Component Designs
- Poor System Integration Testing
- Inadequate System Design
- Poor Safety Procedures
- *…(many, many more)……*

What Goes Wrong With Projects?

Knowing How Development Projects Fail Provides an Awareness of the Need for Systems Engineering

What Can Go Wrong In Developments?

Partial List

1) The Problem or Need is Misunderstood
2) Alternative Solutions Not Identified and Evaluated
3) Wrong Technical Solution Selected
4) All System Level Requirements Not Defined
5) Requirements Not Mutually Compatible or Misunderstood
6) All Appropriate System Dimensions Not Addressed
7) All System Lifecycle Phases Not Addressed
8) Poor Overall System Level / Architectural Design
9) Detail Requirements Before System Level Design Baseline
10) Users Insufficiently Involved in Development
11) Maintenance Approach Not Considered in Requirements or Design
12) Upgradeability Not Considered in Requirements or Design
13) Stakeholders Not on the Same Page for Requirements or Design
14) Inaccurate Estimates of Time and Resources
15) Inadequate Lower Level Requirements

What Can Go Wrong In Developments?

Partial List Continued

16) Inadequate Interface Definitions
17) Inadequate Testing at Lower Levels Prior to System Integration
18) Lacking or Poor Change Management
19) Poor Work Plan
20) Inadequate Developer Technical Skills
21) Inadequate Development Processes or Tools
22) Development Methods Not Aligned with Complexity
23) Poor Management of Project Work and Issues
24) Inadequate Stakeholder Communications
25) Inadequate Internal Team Communications
26) Inadequate Accountabilities Defined
27) Inadequate Technical Skills in Decision Makers
28) No Risk Management
29) Inadequate Sponsor Management
30) Over Designed

Shortcoming	Report A	Report B	Report C	Report D	Report E
1. The Problem or Need is Misunderstood				X	
2. Alternative Solutions Not Identified					
3. Wrong Technical Solution Selected				X	
4. Requirements Not Well Defined	X	X	X	X	X
5. Requirements Not Mutually Compatible					
6. All System Dimensions Not Addressed					
7. All System Lifecycle Phases Not Addressed					
8. Poor Overall Architectural Design					
9. Detail Requirements Before Top Level Design					
10. Users Insufficiently Involved in Development	X		X		
11. Maintenance Approach Not Considered					
12. Upgradeability Not Considered					
13. Stakeholders Not on the Same Page	X		X	X	X
14. Inaccurate Estimates of Time and Resources			X	X	
15. Inadequate Lower Level Requirements					
16. Inadequate Interface Definitions		X			
17. Inadequate Testing Prior to Integration					
18. Lacking or Poor Change Management			X		
19. Poor Work Plan			X		X
20. Inadequate Developer Technical Skills		X	X		
21. Inadequate Development Processes or Tools	X		X	X	X
22. Methods Not Aligned with Complexity					X
23. Poor Management of Project Work and Issues	X		X	X	X
24. Inadequate Stakeholder Communications			X	X	X
25. Inadequate Internal Team Communications					X
26. Inadequate Accountabilities Defined					
27. Poor Technical Skills In Decision Makers					
28. No Risk Management	X			X	
29. Inadequate Sponsor Involvement	X				X
30. Over Designed	X				

Expert Views of Project Shortcomings Vary

("X" = A Project Issue)

Shortcoming	Report A	Report B	Report C	Report D	Report E
1. The Problem or Need is Misunderstood	M	M	M		M
2. Alternative Solutions Not Identified	M	M	M	M	M
3. Wrong Technical Solution Selected	M	M	M	M	
4. Requirements Not Well Defined					
5. Requirements Not Mutually Compatible	M	M	M	M	M
6. All System Dimensions Not Addressed	M	M	M	M	M
7. All System Lifecycle Phases Not Addressed	M	M	M	M	M
8. Poor Overall Architectural Design	M	M	M	M	M
9. Detail Requirements Before Top Level Design	M	M	M	M	M
10. Users Insufficiently Involved in Development		M		M	M
11. Maintenance Approach Not Considered	M	M	M	M	M
12. Upgradeability Not Considered	M	M	M	M	M
13. Stakeholders Not on the Same Page		M			
14. Inaccurate Estimates of Time and Resources	M	M			M
15. Inadequate Lower Level Requirements	M	M	M	M	M
16. Inadequate Interface Definitions	M		M	M	M
17. Inadequate Testing Prior to Integration	M	M	M	M	M
18. Lacking or Poor Change Management	M		M	M	M
19. Poor Work Plan	M	M		M	
20. Inadequate Developer Technical Skills	M			M	M
21. Inadequate Development Processes or Tools		M			
22. Methods Not Aligned with Complexity	M	M	M		M
23. Poor Management of Project Work and Issues		M			
24. Inadequate Stakeholder Communications	M	M			
25. Inadequate Internal Team Communications	M	M	M	M	
26. Inadequate Accountabilities Defined	M	M	M	M	M
27. Poor Technical Skills In Decision Makers	M	M	M	M	M
28. No Risk Management		M	M		M
29. Inadequate Sponsor Involvement		M	M	M	
30. Over Designed		M	M	M	M

Many Views of Project Shortcomings Miss Key Root Causes

(“M” = Missing in Report)

What Does this Mean?

Multiple views *are needed to fully understand challenges to projects.*

Systems Engineering is related to virtually all *technical issues projects face.*

With ***knowledge of "SE Core Concepts"****, project teams can* ***avert many causes*** *of project challenges and failure.*

Shortcoming	Report A	Report B	Report C	Report D	Report E	29
1. The Problem or Need Is Misunderstood				X		
2. Alternative Solutions Not Identified						
3. Wrong Technical Solution Selected				X		
4. Requirements Not Well Defined	X	X	X	X	X	
5. Requirements Not Mutually Compatible						
6. All System Dimensions Not Addressed						
7. All System Lifecycle Phases Not Addressed						
8. Poor Overall Architectural Design						
9. Detail Requirements Before Top Level Design						
10. Users Insufficiently Involved In Development	X		X			
11. Maintenance Approach Not Considered						
12. Upgradeability Not Considered						
13. Stakeholders Not on the Same Page	X		X	X	X	
14. Inaccurate Estimates of Time and Resources			X	X		
15. Inadequate Lower Level Requirements						
16. Inadeq[illegible]		X				
17. Inadequate Testing Prior to Integration						
18. Lacking or Poor Change Management			X			
19. Poor Work Plan			X		X	
20. Inadequate Developer Technical Skills		X	X			
21. Inadequate Development Processes or Tools	X		X	X	X	
22. Methods Not Aligned with Complexity					X	
23. Poor Management of Project Work and Issues	X		X	X	X	
24. Inadequate Stakeholder Communications			X	X	X	
25. Inadequate Internal Team Communications					X	
26. Inadequate Accountabilities Defined						
27. Poor Technical Skills In Decision Makers						
28. No Risk Management	X			X		
29. Inadequate Sponsor Involvement	X				X	
30. Over Designed	X					

Expert Views of Project Shortcomings Vary
("X" = A Project Issue)

Shortcoming	Report A	Report B	Report C	Report D	Report E	30
1. The Problem or Need Is Misunderstood	M	M	M		M	
2. Alternative Solutions Not Identified	M	M	M	M	M	
3. Wrong Technical Solution Selected	M	M	M	M		
4. Requirements Not Well Defined						
5. Requirements Not Mutually Compatible	M	M	M	M	M	
6. All System Dimensions Not Addressed	M	M	M	M	M	
7. All System Lifecycle Phases Not Addressed	M	M	M	M	M	
8. Poor Overall Architectural Design	M	M	M	M	M	
9. Detail Requirements Before Top Level Design	M	M	M	M	M	
10. Users Insufficiently Involved In Development		M		M	M	
11. Maintenance Approach Not Considered	M	M	M	M	M	
12. Upgradeability Not Considered	M	M	M	M	M	
13. Stakeholders Not on the Same Page		M				
14. Inaccurate Estimates of Time and Resources	M	M			M	
15. Inadequate Lower Level Requirements	M	M	M	M	M	
16. Inadeq[illegible]	M		M	M	M	
17. Inadequate Testing Prior to Integration	M	M	M	M	M	
18. Lacking or Poor Change Management	M		M	M	M	
19. Poor Work Plan	M	M		M		
20. Inadequate Developer Technical Skills	M			M	M	
21. Inadequate Development Processes or Tools		M				
22. Methods Not Aligned with Complexity	M	M	M		M	
23. Poor Management of Project Work and Issues		M				
24. Inadequate Stakeholder Communications	M	M				
25. Inadequate Internal Team Communications	M	M	M	M		
26. Inadequate Accountabilities Defined	M	M	M	M	M	
27. Poor Technical Skills In Decision Makers	M	M	M	M	M	
28. No Risk Management		M	M		M	
29. Inadequate Sponsor Involvement		M	M	M		
30. Over Designed		M	M	M	M	

Many Views of Project Shortcomings Miss Key Root Causes
("M" = Missing in Report)

A Famous Quote

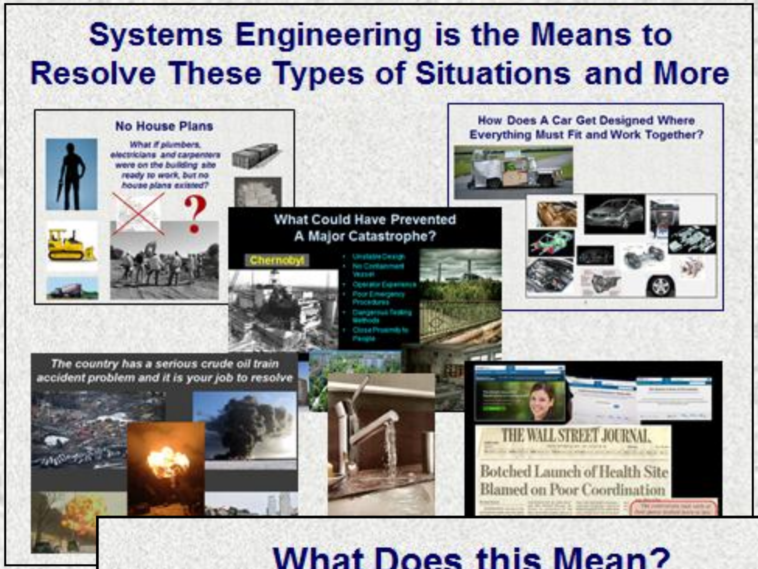
"Most projects in trouble today, started out that way"

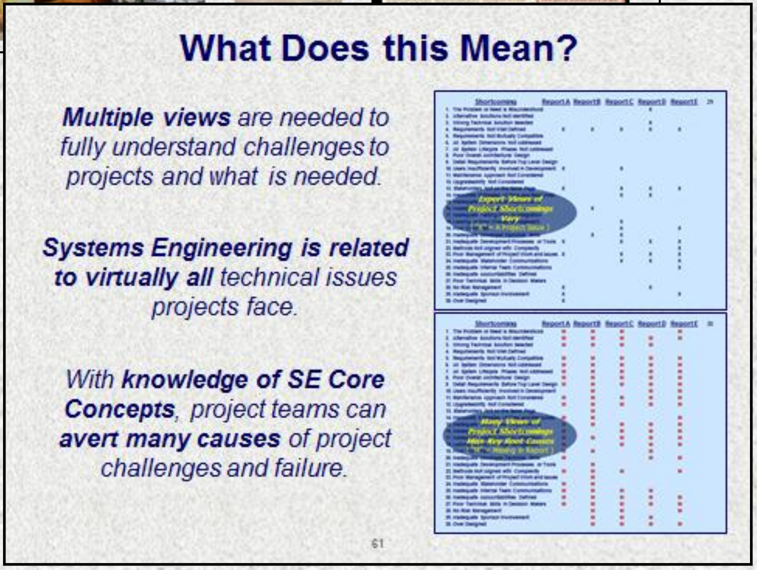
To keep any project out of trouble requires many things including a sound understanding of systems engineering core concepts

Systems Engineering Basics are the Answer to Many Project Problems

Basic SE Functions

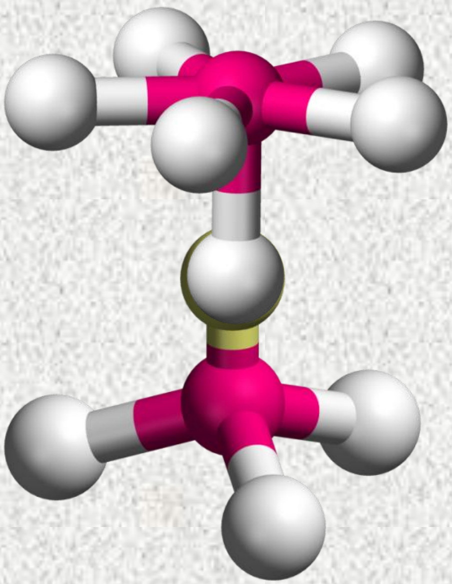
- Problem / Improvement Definition
- Solution Requirements
- Alternative Solution Identification
- Solution Selection
- System Level Solution Design
- System Component Requirements
- Design Integration
- System Integration and Validation
- System Operation Support
- System Upgrade Design
- System Disposal



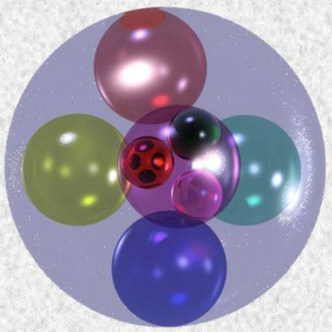


What is a System?

“System”



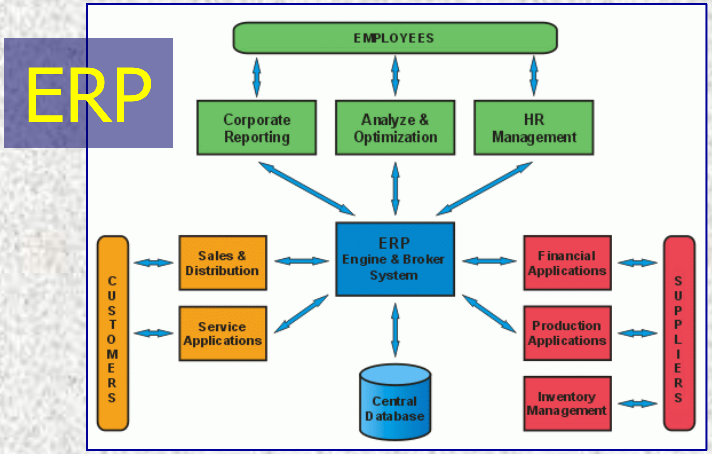
Sys-tem: *An assembly of different things that work in concert to form an entity that performs a desired function, serves a desired purpose or solves a given problem.*

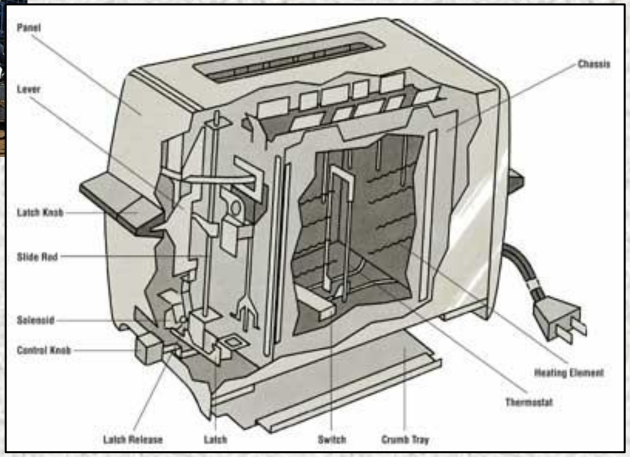


System

System……

The combination of things, often hardware, software, procedures and humans, that taken together perform a desired set of functions meeting a defined need.



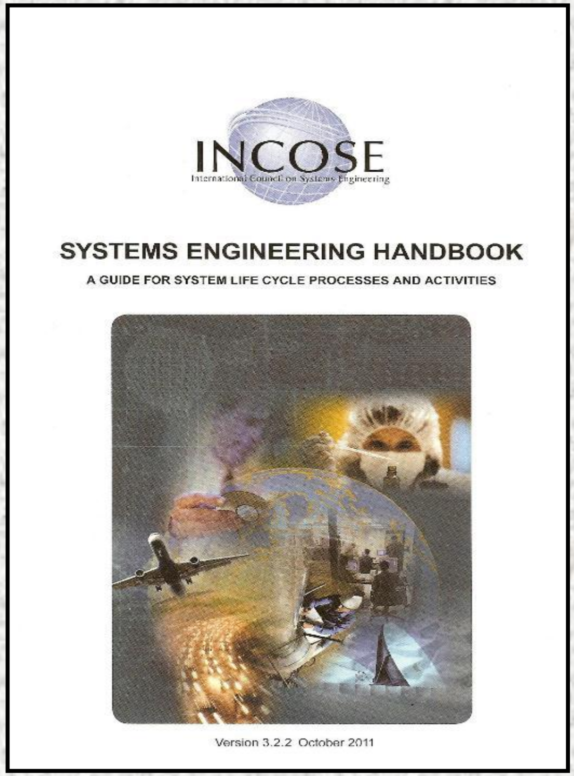


"System"

INCOSE - An interacting combination of elements viewed in relation to function.

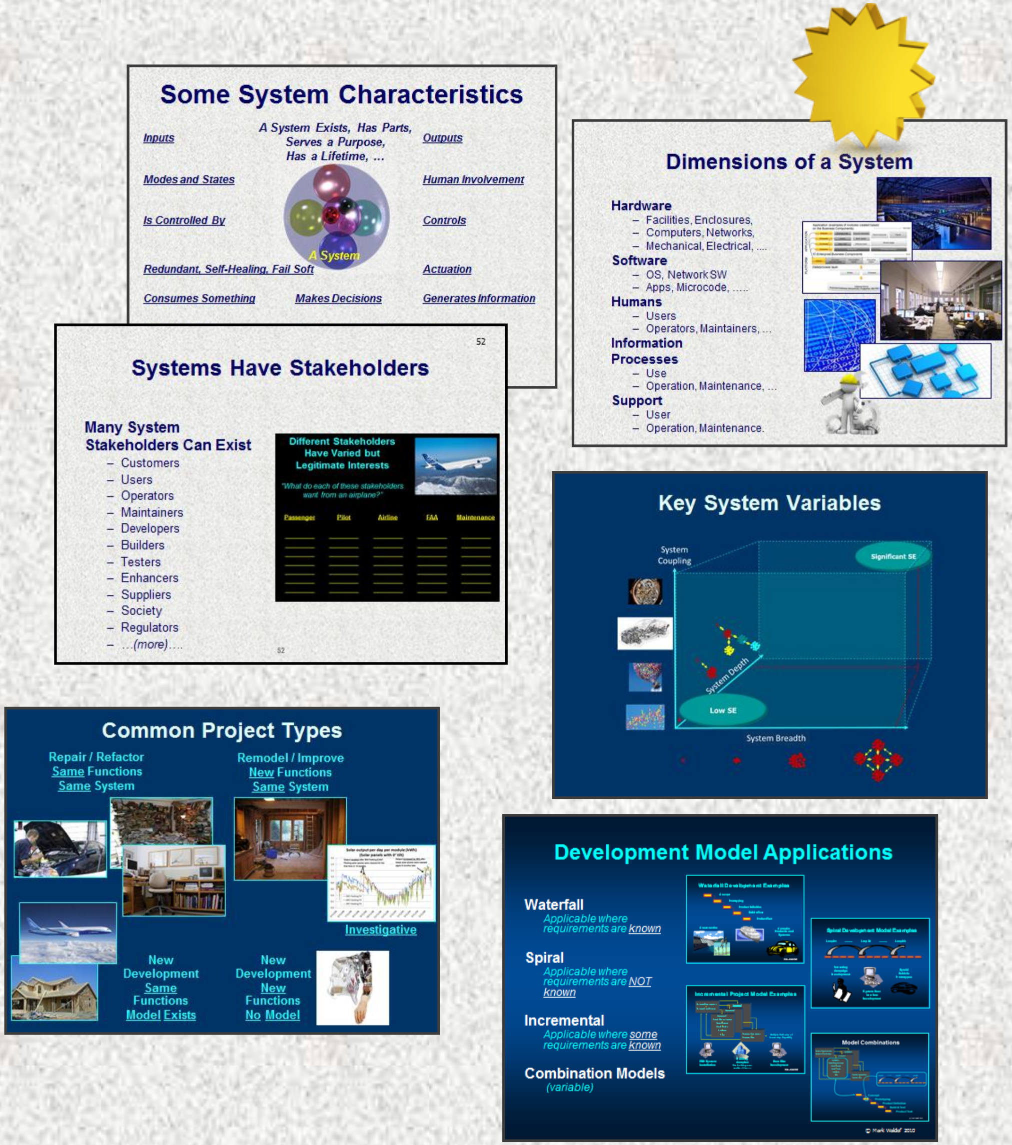
NASA Systems Engineering Handbook - A set of interrelated components which interact with one another in an organized fashion toward a common purpose.

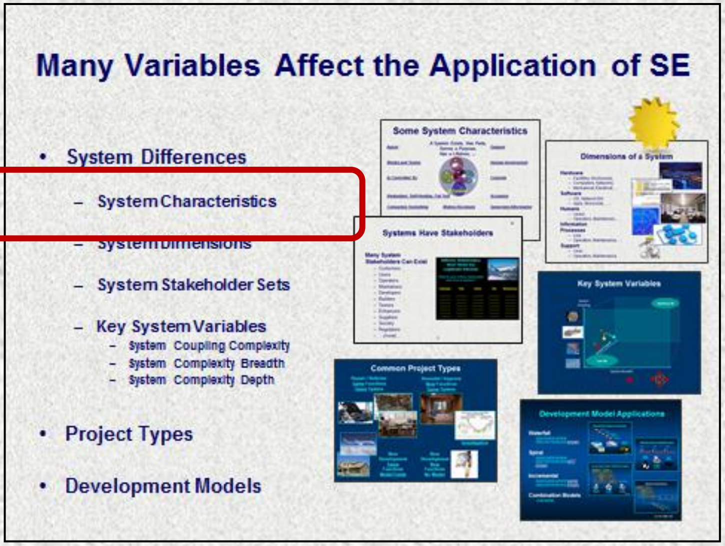
US DoD - An integrated composite of people, products, and processes that provide a capability to satisfy a stated need or objective.



Many Variables Affect the Application of SE

- **System Differences**
  - System Characteristics
  - System Dimensions
  - System Stakeholder Sets
  - Key System Variables
    - System Coupling Complexity
    - System Complexity Breadth
    - System Complexity Depth
- **Project Types**
- **Development Models**





Systems Have Different

System Characteristics

Variable System Characteristics

Inputs
- *Has inputs, or*
- *May not have inputs*

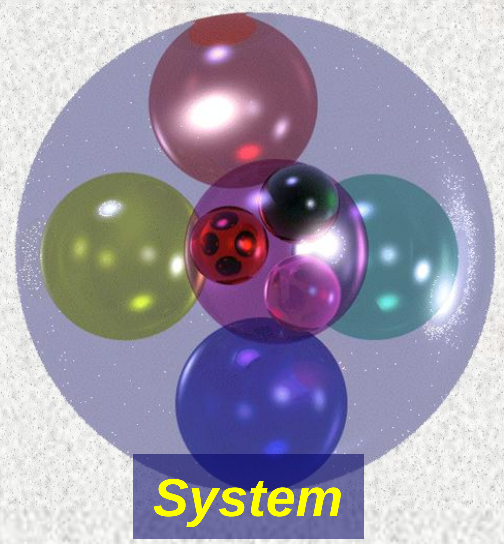
Modes and States
- *Modes – Few or Many*
- *States – Few or Many*

Is Controlled By
- *Controlled by humans*
- *Controlled by another system*
- *Not controlled by anything*

Redundant, Self-Healing, Fail Soft
- *No or Yes*

Consumes Something
- *No or Yes*

Components
- *Physical, and/or*
- *Human, and/or*
- *Information, and/or*
- *Combinations*



Makes Decisions
- *No or Yes*

Energy Relationship
- *Uses energy*
- *Produces energy*
- *Transforms energy*
- *Not related to energy*

Outputs
- *Has outputs, or*
- *May not have outputs*

Human Involvement
- *No Human Involvement*
- *Humans For Operation*
- *Humans For Maintenance*

Controls
- *Controls humans*
- *Controls another system*
- *Does not control anything*

Upgradeable
- *No or Yes*

Generates Information
- *No or Yes*

Context
- *Is part of a larger system*
- *Is not a part of a larger system*

Example System Characteristics

Inputs
- No Inputs

Outputs
- No Outputs

Modes and States
- One

Human Involvement
- None for Operation
- Humans Move on its Exterior

Controlled By
- Nothing

Controls
- Nothing

Redundant, Self-Healing, Fail Soft
- Redundant Design

A Fixed Bridge

Upgradeable
- No

Consumes Something
- No

Makes Decisions
- No

Generates New Information
- No

Components
- Multiple Physical

Energy Relationship
- None Used
- None Produced

Context
- Is Part of a Larger System

Example System Characteristics

Inputs
- Sensors
- Power
- Maintenance

Outputs
- Heart Stimulus
- Status

Modes and States
- Multiple

Human Involvement
- None for Operation

Controlled By
- Software

Controls
- A Heart

Redundant, Self-Healing, Fail Soft
- Redundant Design

A Pacemaker

Upgradeable
- Yes for Software
- No for Hardware

Consumes Something
- Power

Makes Decisions
- Yes

Generates New Information
- Yes

Components
- Multiple Physical
- Multiple Information

Energy Relationship
- Uses Energy

Context
- Is Part of a Larger System

Example System Characteristics

Inputs
- Wind

Outputs
- Electricity

Modes and States
- Off
- Standby
- Operational

Human Involvement
- None

Controlled By
- Another System

Controls
- Nothing

Redundant, Self-Healing, Fail Soft
- No

Upgradeable
- Yes

Consumes Something
- Nothing

Makes Decisions
- No

Generates New Information
- No

Components
- Multiple Physical and Informational

Energy Relationship
- Converts Energy

Context
- Is Part of a Larger System

Wind Energy System

Example System Characteristics

Inputs
- Sensors
- Power
- Maintenance

Outputs
- RF Radiation
- Display
- Status

Modes and States
- Many

Human Involvement
- Significant

Controlled By
- Software
- External Systems
- User

Controls
- Multiple

Redundant, Self-Healing, Fail Soft
- Not Really

A Cell Phone

Upgradeable
- Yes for Software
- No for Hardware

Consumes Something
- Power

Makes Decisions
- Yes

Generates New Information
- Yes

Components
- Physical
- Information
- Software

Energy Relationship
- Uses Energy

Context
- Is Part of a Larger System

What Can Go Wrong?
What Do You Do?

POTENTIAL PROJECT FAILURE

- If all key system characteristics are not defined, the system may fail to meet all needs

WHAT TO DO

- Know what system characteristics apply to your developments
- Make sure all appropriate characteristics are defined
- Ensure the team and stakeholders understand and agree on system characteristics

Example System Characteristics

Inputs
- *Has inputs, or*
- *May not have inputs*

Modes and States
- *Modes – Few or Many*
- *States – Few or Many*

Is Controlled By
- *Controlled by humans*
- *Controlled by another system*
- *Not controlled by anything*

Outputs
- *Has outputs, or*
- *May not have outputs*

Human Involvement
- *No Human Involvement*
- *Humans For Operation*
- *Humans For Maintenance*

Controls
- *Controls humans*
- *Controls another system*
- *Does not control anything*

Redundant, Self-Healing, Fail Soft
- *No or Yes*

Upgradeable
- *No or Yes*

Consumes Something
- *No or Yes*

Makes Decisions
- *No or Yes*

Generates Information
- *No or Yes*

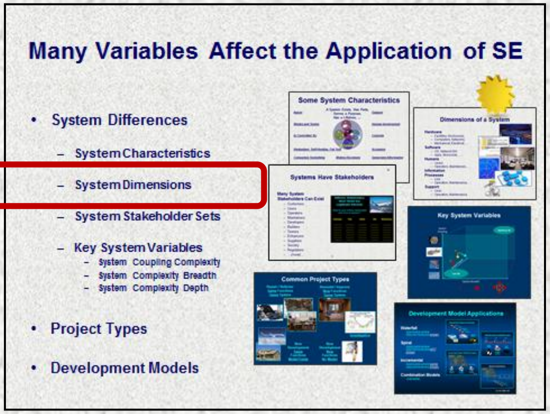
Components
- *Physical, and/or*
- *Human, and/or*
- *Information, and/or*
- *Combinations*

Energy Relationship
- *Uses energy*
- *Produces energy*
- *Transforms energy*
- *Not related to energy*

Context
- *Is part of a larger system*
- *Is not a part of a larger system*

Project Variable	*In your team, identify what characteristics exist for an example system you work with*
Inputs	
Outputs	
Modes and States	
Human Involvement	
Controlled By	
Controls.......	
Redundant, Self-Healing?	
Upgradeable	
Consumes Something	
Makes Decisions	
Generates New Information	
Components	
Energy Relationship	
Context Relative to Other System	

Decide, Write Them Down!



Systems Engineering Addresses All

System Dimensions

Systems Have Dimensions

A system is typically composed of multiple dimensions

These dimensions can include all or some of the following……

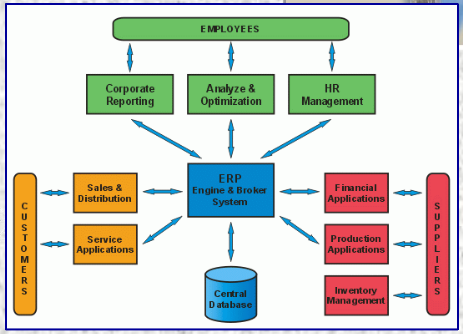
- Hardware
- Software
- Humans
- Information
- Processes
- Support

<u>All needed system dimensions must be addressed</u> during development

Examples of System Dimensions

Common System Dimensions

- Hardware
  - **Operational**
  - **Test and Maintenance**
- Software
  - **OS, Apps, Microcode, …..**
  - **Test and Maintenance**
- Humans
  - **Operators**
  - **Users**
  - **Testers and Maintainers**
- Information
- Processes
  - **User**
  - **Operational**
  - **Test and Maintenance**
- Support
  - **Training, Experts**
  - **Guides, Call Centers, ……**



Example System Dimensions
Automobile

System Dimensions

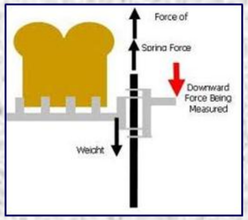
- Hardware
  - **Chassis, Body, Interior**
  - **Engine, Computers, Transmission, ..**
- Software
  - **ENGINE TRANS SOFTWARE**
- Information
  - **Base Operation Information**
- People
  - **Owner, Driver**
  - **Passengers**
  - **Maintainers**
- Processes
  - **User – OWNERS MANUAL**
  - **Operational – OWNERS MANUAL**
  - **Maintenance – SHOP MANUAL**
- Support
  - **Training – DRIVER ED**
  - **Call Center or Experts - DEALER**

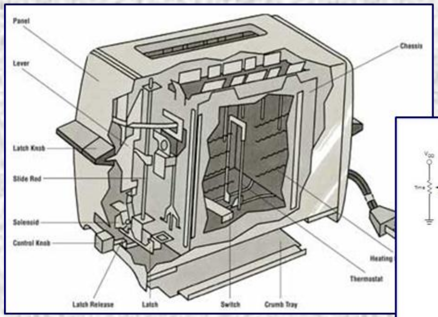
System Dimensions

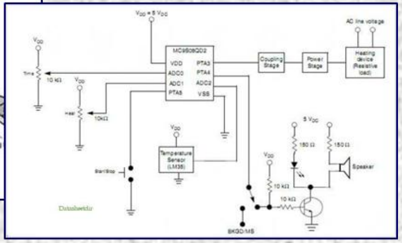
Toaster

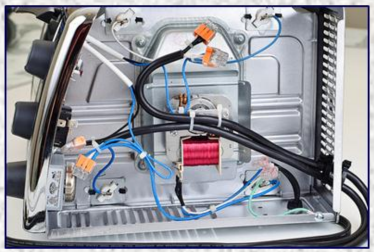
System Dimensions

- Hardware
  - **Enclosure**
  - **Elements**
  - **Controls**
- Software
  - **Logic Control**
- People
  - **User**
- Information
- Processes
  - **Operation**
  - **Emergency**
- Support
  - **Manual**
  - **Help Line**





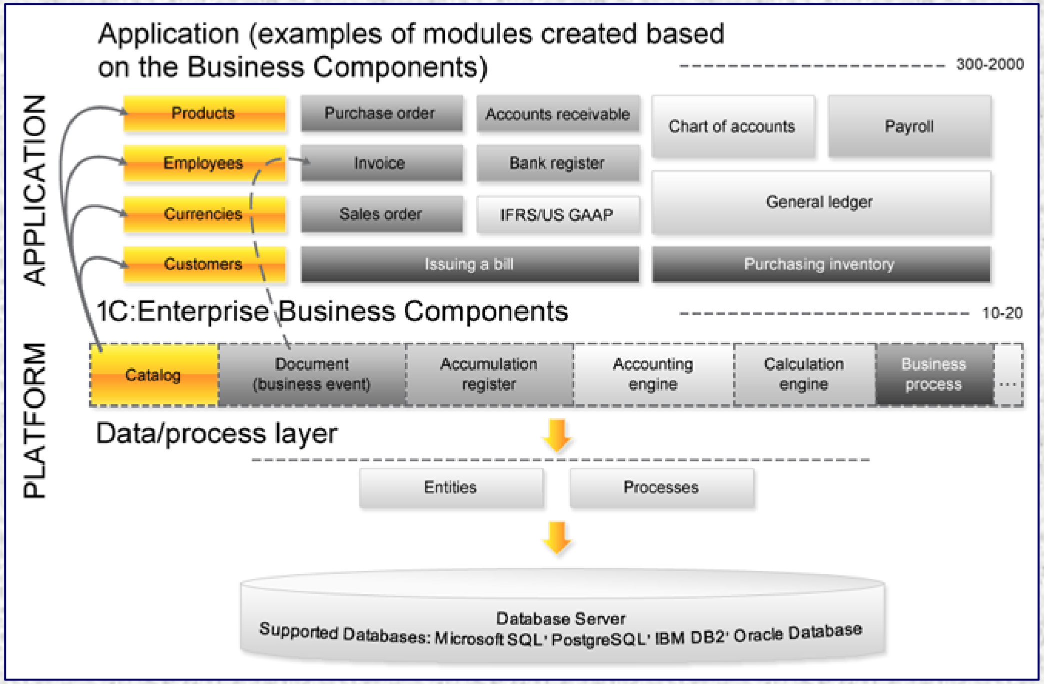




System Dimensions
Business Application

System Dimensions

- Hardware
  - **Servers**
  - **Networks**
- Software
  - **Control, DBM**
  - **App 1**
  - **App n**
- People
  - **Users**
  - **Maintainers**
- Information
  - **Data Base 1**
  - **Data Base n**
- Processes
  - **User A**
  - **User N**
- Support
  - **Call Center**
  - **On-Line Help**



Example System Dimensions
Cell Phone

System Dimensions.

- Hardware
  - **Frame, Screen, Circuit Board, Battery, Xcvr, Camera, ……**
- Software
  - **OS, Phone SW, Apps, …..**
- People
  - **User / Operator**
- Information
  - **User Information**
  - **……….**
- Procedures
  - **OWNERS MANUAL**
- Support
  - **Training – Built In, Manual, Web, ..**
  - **Customer Service**

System Dimensions Building

System Dimensions

- Hardware
  - **Structure**
  - **HVAC**
  - **……**
- Software
  - **Environment Control**
  - **Security**
  - **Communication**
  - **……**
- People
  - **Operators**
  - **Users**
  - **Maintainers**
- Information
- Procedures
  - **Operational**
  - **Emergency**
  - **Maintenance**
- Support
  - **(Multiple)**

Common System Dimension Shortfalls

- Developers focus on one or two dimensions only
  - Hardware suite defined does not meet all software needs
  - System designed without consideration of current processes
  - Software designed without considerations for maintenance and update
- Dimensions addressed are not developed to be……
  - Mutually compatible
  - Efficiently compatible
- Human interfaces are not well considered
  - User needs not sufficiently reflected in the design
  - Assumed user skill levels not consistent with real users
  - User support not adequately addressed
- System not designed for ease in fault isolation or repair
- System not designed for ease of upgrade
- Inadequate focus on complete and understandable procedures
- Procedures that miss system operational scenarios or modes
- *……(more)……..*

What Can Go Wrong?
What Do You Do?

POTENTIAL PROJECT FAILURE

- Developments that do not address all system dimensions may not work, might fail later or might be inefficient

WHAT TO DO

- Identify all needed dimensions and make sure all needed system dimensions are addressed and integrated

- Educate your team and ensure stakeholders understand the need for addressing all system dimensions

System Dimensions

What To Do

Reference

You Are A Client

- Be aware of the need for all system dimensions to be addressed
- Include requirements for all dimensions in your work direction to the contractor

You Are Management

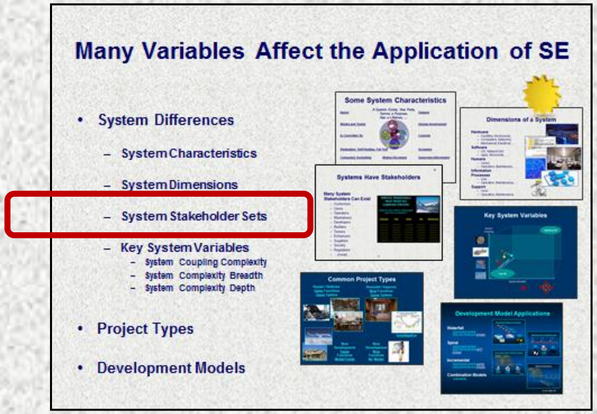
- Educate your PMs on SE and system dimensions
- Make sure all needed dimensions are addressed
- Ask about dimension considerations at project startup and during the project

You Are An SE, Project Manager or Technical Lead

- Make sure your work responsibilities include or specifically do not include system dimension efforts
- Educate your team
- Ensure all needed dimensions are addressed and integrated

You Are A Team Member

- Ask about system dimension considerations during project startup
- Bring up issues that might result if all system dimensions are not addressed



Systems Engineers Address the Needs of All

System Stakeholders

Full Group Discussion

Requirements Come From Different Stakeholders

"what do each of these stakeholders want from an airplane?"

Passenger

- ________________
- ________________
- ________________
- ________________
- ________________
- ________________
- ________________

Pilot

- ________________
- ________________
- ________________
- ________________
- ________________
- ________________
- ________________

Airline

- ________________
- ________________
- ________________
- ________________
- ________________
- ________________
- ________________

Maintenance

- ________________
- ________________
- ________________
- ________________
- ________________
- ________________
- ________________

System Stakeholders

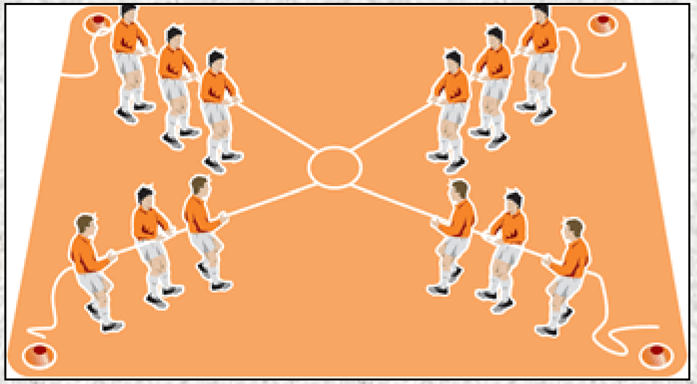
Common System Stakeholders

- Customers
- Users
- Operators
- Maintainers
- Developers
- Builders
- Testers
- Enhancers
- Suppliers
- Society As A Whole
- ………..

Example Stakeholder Involvement

Customers

- Needed outcomes, requirements, system scope and boundaries
- Consistent involvement in all phases of development



Users

- Needed outcomes, user requirements, beta/pilot involvement & feedback
- Consistent involvement in all development phases affecting users

Different Agendas

Maintainers

- Involved in system design to impact design modularity, self-tests, built-in diagnostics and related maintenance actions
- Consistent development phase involvement, authority to impact design

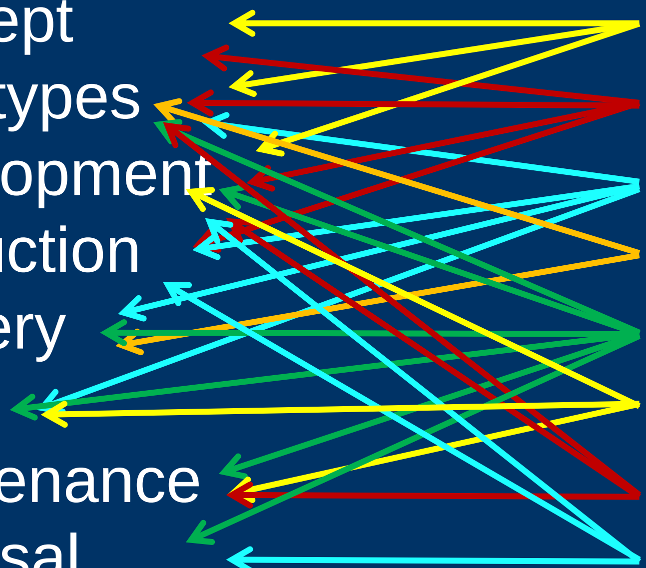
Developers

- Understand higher level system design, understand needed outcomes, supports integrated developments
- Continuously monitors developments to ensure system compatibility

Each Stakeholder Can Be Involved in Multiple System Lifecycle Phases

Lifecycle Phase

- Concept
- Prototypes
- Development
- Production
- Delivery
- Use
- Maintenance
- Disposal



Stakeholder

- Marketing
- Engineering
- Production
- Dealer
- Buyer
- Support
- Maintainer
- Disposer

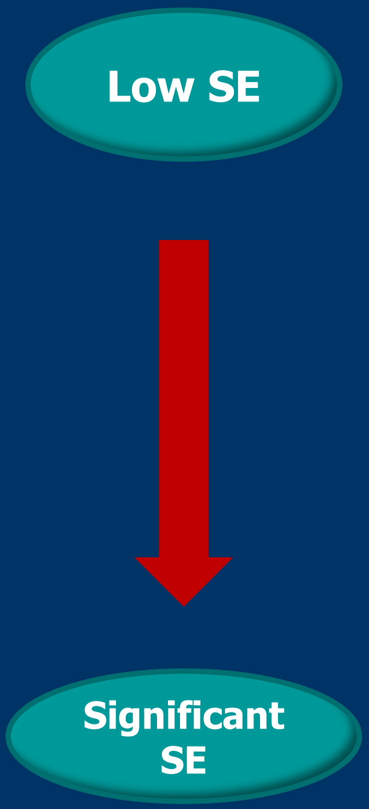
Notional Involvement Only

Key System Variables

How Stakeholders Affects Design

Few Stakeholders

- Developer is the User



- Little Communication Needed for Requirements, Design, Build, Test and Validation
- Little Coordination Required
- Easy Baseline Controls
- No Opposing Stakeholder Interests

Many Stakeholders

- Owners
- Buyers
- Users
- Developers
- Suppliers
- Maintainers
- Regulators
- Legal
- …….

- High Level of Communication of Needed for Requirements, Design, Build, Test and Validation
- Coordination Required
- Baseline Controls Required
- Opposing Stakeholder Interests Need Facilitation

Common Stakeholder Shortfalls

When stakeholders are not adequately involved, many of the common project problems can result.

What Can Go Wrong In Developments?

Partial List

1) The Problem or Need is Misunderstood
2) Alternative Solutions Not Identified and Evaluated
3) Wrong Technical Solution Selected
4) All System Level Requirements Not Defined
5) Requirements Not Mutually Compatible or Misunderstood
6) All Appropriate System Dimensions Not Addressed
7) All System Lifecycle Phases Not Addre
8) Poor Overall System Level / Architectu
9) Detail Requirements Before System Le
10) Users Insufficiently Involved in Develop
11) Maintenance Approach Not Considere
12) Upgradeability Not Considered in Requ
13) Stakeholders Not on the Same Page fo
14) Inaccurate Estimates of Time and Res
15) Inadequate Lower Level Requirements

What Can Go Wrong In Developments?

Partial List Continued

16) Inadequate Interface Definitions
17) Inadequate Testing at Lower Levels Prior to System Integration
18) Lacking or Poor Change Management
19) Poor Work Plan
20) Inadequate Developer Technical Skills
21) Inadequate Development Processes or Tools
22) Development Methods Not Aligned with Complexity
23) Poor Management of Project Work and Issues
24) Inadequate Stakeholder Communications
25) Inadequate Internal Team Communications
26) Inadequate Accountabilities Defined
27) Inadequate Technical Skills in Decision Makers
28) No Risk Management
29) Inadequate Sponsor Management
30) Over Designed

System Stakeholders
What Should Be Done?

All Stakeholders Must….

- **<u>Be</u> <u>Identified</u>**
- **<u>Be</u> <u>Known</u> to Other Stakeholders**
- **Have a <u>Defined</u> <u>Role</u> that is Accepted by All Stakeholders**
- **Have a <u>Defined</u> <u>Decision</u> <u>Authority</u> that is Accepted by All Stakeholders**
- **Have a <u>Defined</u> <u>Communication</u> <u>Scheme</u> that enables on-going stakeholder to stakeholder and stakeholder to team communications**

What Can Go Wrong?
What Do You Do?

POTENTIAL PROJECT FAILURE

- Missing stakeholder needs can mean system failure or an ineffective design

WHAT TO DO

- Ensure all stakeholder roles, accountabilities and decision authorities are defined
- Ensure stakeholder needs drive requirements and design
- Ensure adequate stakeholder involvement exists during development

System Stakeholders

What To Do

Reference

You Are A Client

- Know all stakeholders
- Work with the contractor to define all stakeholder roles
- Support defined roles during the project

You Are Management

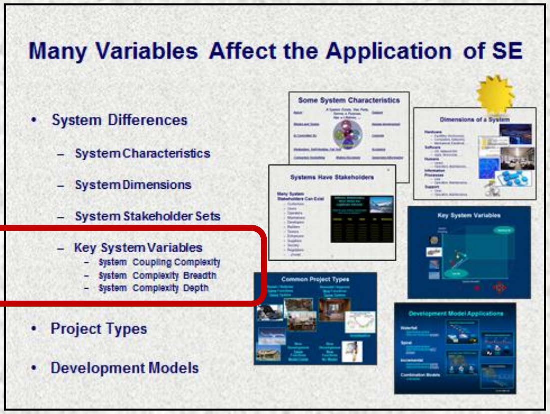
- Know all stakeholders
- Define roles and authorities to team
- Validate team understanding
- Direct development teams on stakeholder interaction
- Check on needed stakeholder involvement during development

You Are An SE, Project Manager or Technical Lead

- Know all stakeholders
- Educate the team on stakeholder roles
- Ensure stakeholder interaction during the project
- Establish baselines to support sound stakeholder expectations

You Are A Team Member

- Ask about stakeholders, their roles and authorities
- Raise issues on stakeholder roles and authorities that you feel threaten project success



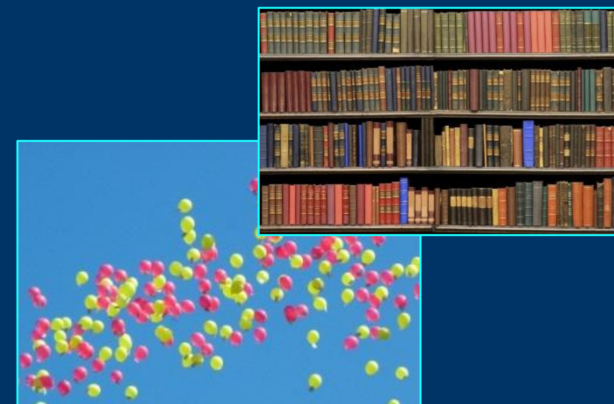
Systems Engineering Deals With

System Variables

Coupling, Depth and Breadth

Key System Variables
System Components Coupling

Little or No Coupling



Each system component acts independently

Example: Stores in a mall, courses in a curriculum, books in a library, balloons in the air…..

Some Coupling

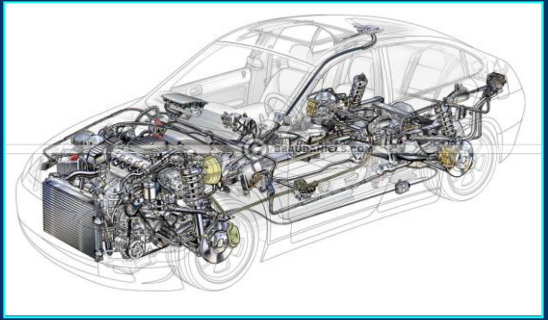
Simple system component interaction

Example: Independent business apps working with a common data base

Key System Variables

System Components Coupling

Strong Coupling



Many components must work together

Example: Car, entertainment, navigation, lighting, safety equip are integrated but are also independent from chassis, engine, …….

Very Strong Coupling

Components dependent on each other

Example: Watch - All components necessary to function

Key System Variables

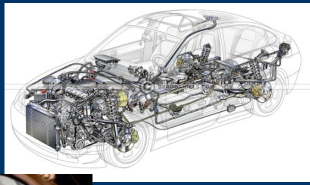
How Coupling Affects SE

- Low Component Coupling
- Low Component Interaction
- Low Component Interface Design
- Low Component Effects on System Function

Low SE

- High Component Coupling
- High Component Interaction
- High Component Interaction Design
- High Component Effects on System Function

Significant SE

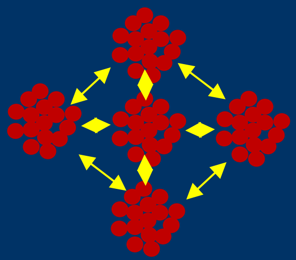


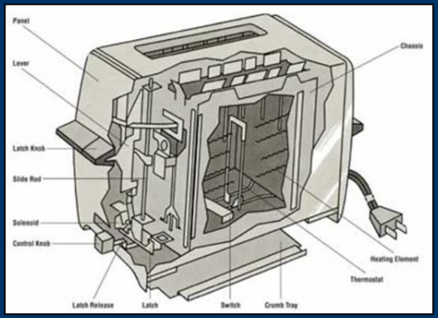
Key System Variables

System Breadth Complexity

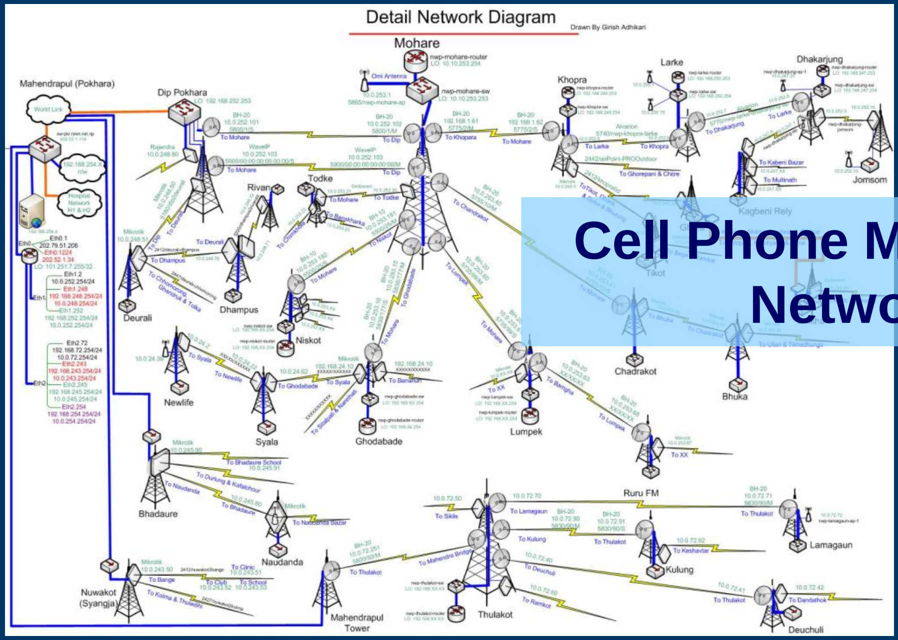
Not Broad

Very Broad





A Toaster



Cell Phone Microwave Network

Key System Variables

How System Breadth Affects SE

- Single System
- Low Complexity Functions
- Low Complexity Interfaces

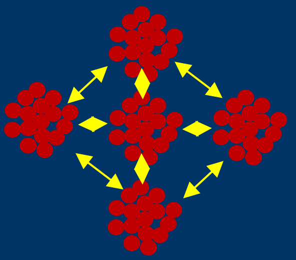
Low SE

Not Broad

Very Broad

- Multiple Peer System Components
- High Complexity Functions
- High Complexity Interfaces
- High Complexity Component Interactions

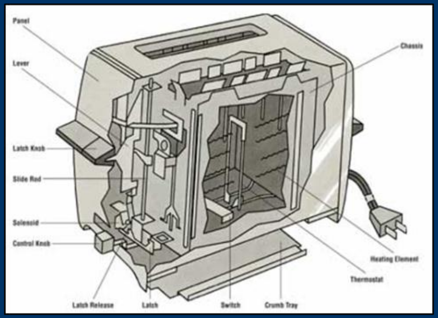
Significant SE



Key System Variables

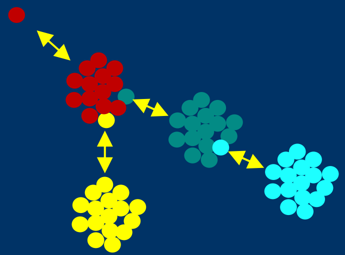
System Depth Complexity

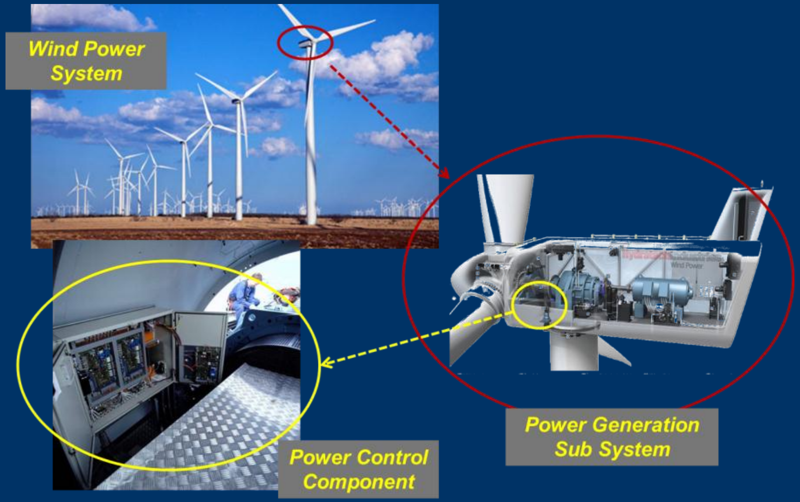
Not Deep



A Toaster

Very Deep





Power System

Key System Variables

How System Depth Affects SE

- Single Level Requirements
- Single Level Design
- Single Level Test and Validation
- Single Level Maintenance

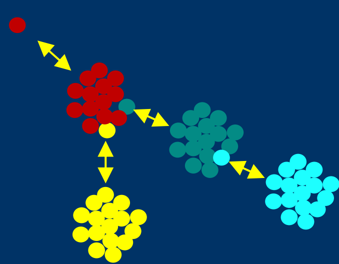
Low SE

Not Deep

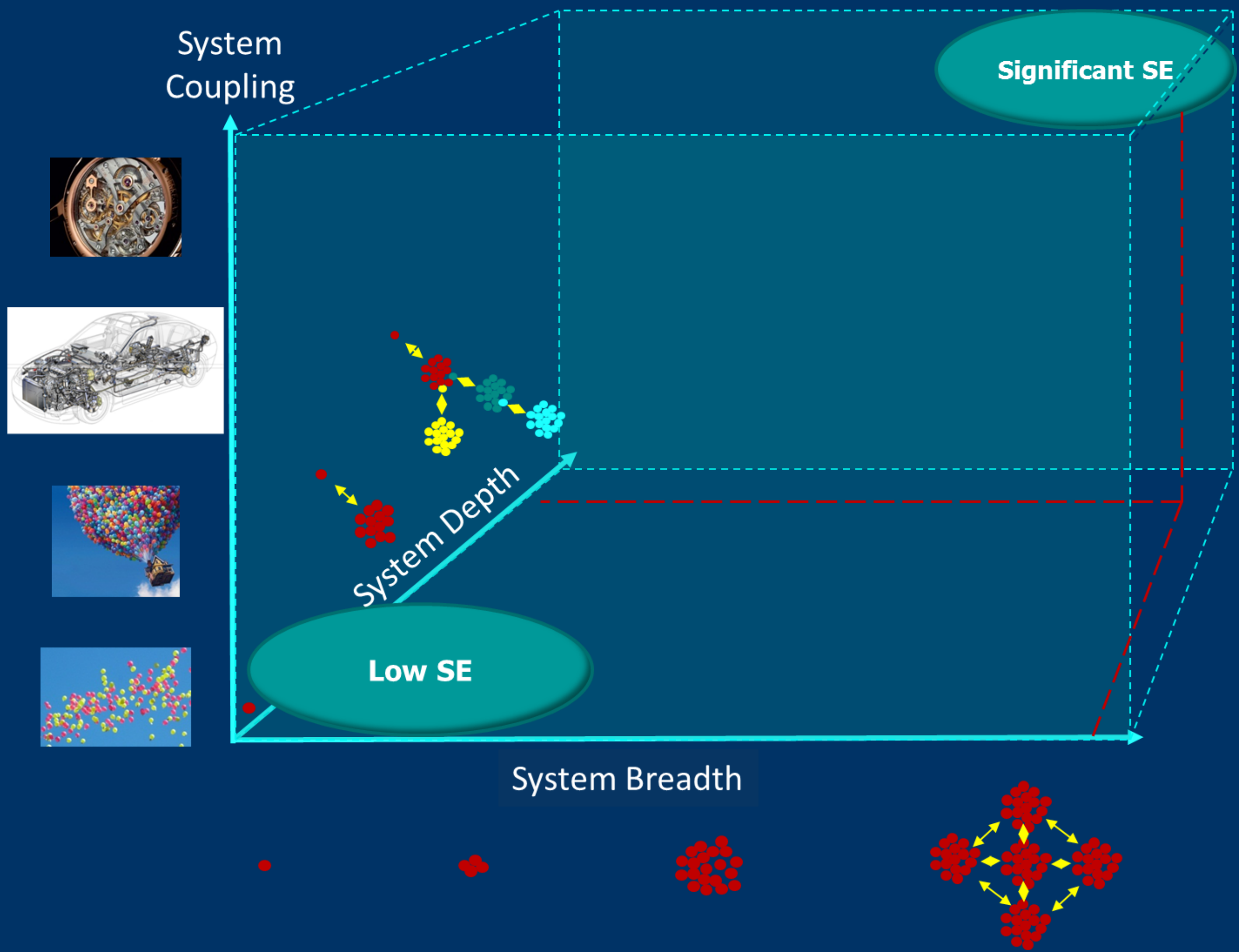
- Multi Level Requirements
- Multi Level Design
- Multi Level Integration
- Multi Level Test and Validation
- Multi Level Maintenance

Significant SE

Very Deep



Level of SE Needed



What Does a Low Level and a Significant Level of SE Mean?

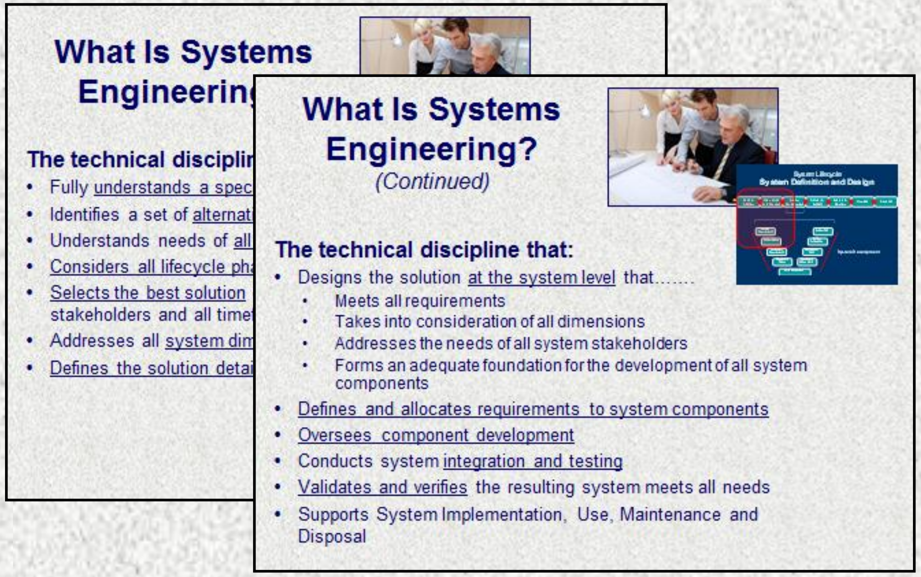
Low Level of SE =

Limited SE practices may be adequate for a non-complex development

- Need Definition & Validation
- Alternative Solutions Considered
- Criteria Based Solution Selection
- Solution Requirements
- System (Solution) Level Design
- Solution Component Requirements

High Level of SE =

Performing all SE concepts included in this course with rigor



What Can Go Wrong?
What Do You Do?

POTENTIAL PROJECT FAILURE

- An inadequate level of SE will invariably inject errors into any development

WHAT TO DO

- Understand SE basics
- Ensure the appropriate level of SE functions are applied
- Educate the team and stakeholders on why SE basics are needed

Systems Engineering Applies to Different

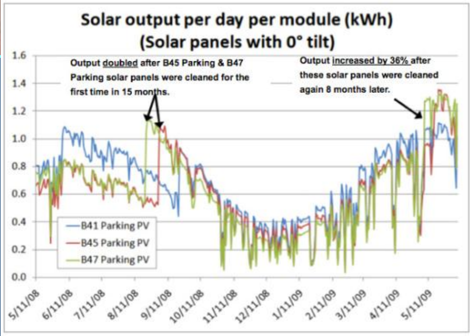
Types of Projects

Project Types

Repair / Refactor
<u>Same</u> Functions
<u>Same</u> System

Remodel / Improve
<u>New</u> Functions
<u>Same</u> System

<u>Investigative</u>



New Development
<u>Same</u> Functions
<u>New</u> System

New Development
<u>New</u> Functions
<u>New</u> System

Project Types

Repair / **Refactor**	Remodel / **Improve**	New **Model**	New **Type**	**Investigative**
Start = Something, Same Functions, Fix or Cleanup Existing Platform	Start = Something, New Functions, Existing Platform	Start = Previous Model, New Functions, Similar Platform	Start = Nothing, New Functions, New Platform	Start = Nothing Investigate & Report Results
○ Learn Existing ○ Identify Problem ○ Identify What Will Be Done ○ Repair, Re-Organize ○ Test	○ Understand New Needs ○ Tear Apart ○ Determine New Solution ○ Requirements ○ Design ○ Build ○ Integrate ○ Test	○ Understand Predecessor ○ Define New Needs ○ Requirements ○ Design ○ Build ○ Integration ○ Test	○ Define Need ○ Alternative Solutions ○ Concept ○ Requirements ○ Design ○ Build ○ Integration ○ Test	○ Understand New Information Need ○ Define Approach ○ Validate Method ○ Investigate ○ Validate Data ○ Report

How Systems Engineering Relates to Different Project Types

A) Repair / Refactor
- Understand the Current System & Model
- Root Cause Analysis
- Solution Definition and Action

B) Improve
- Understand the Operating Model
- Define New Improvement Needs
- Improvement (System) Design and Build
- Improvement Integration and Test

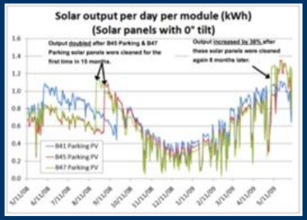
C) New Model
- New Need Definition
- Modify Existing Model of Operations
- System Design, Managed Developments
- Integration and Test

D) New Type
- New Need Definition
- Define New Model of Operations
- System Design, Managed Developments
- Integration and Test

E) Investigative
- (Various SE principles based on project)







Common Project Variables

Projects are different in many ways………

- **Start Conditions**
- Where **Deliverable Requirements** come from
- Who defines project **Constraints**
- What **Project Phases** are needed
- What **Development Model** is Needed
- **Who Funds** the project
- Is a **Contract** or **Client Agreement** involved
- **Who Manages** the project
- Who **Defines** the **Results** of a project
- Who **Accepts** the project results
- Who **Validates** that the project is complete
- Who **Uses the Project Deliverables**
- Who Will **Maintain and Enhance**

Why Is the Project Type Important?

- *The project type determines many project variables*

- *Knowledge of these variables is needed to reduce chaos and to support project success*

Common Project Variables

Projects are different in many ways………

- **Start Conditions**
- Where **Deliverable Requirements** come from
- Who defines project **Constraints**
- What **Project Phases** are needed
- **Who Funds** the project
- Is a **Contract** involved
- **Who Manages** the project
- Who **Defines** the **Results** of a project
- Who **Accepts** the project results and **Validates** that the project is complete
- Who **Uses the Project Deliverables**

Project Variable	*In your team, discuss which of the following project variables needs better definition and/or communication in your environment.*
Existing? Model? Nothing?	
Requirements Source	
Who Defines Constraints	
Defines Phases	
What Development Model	
Who Provides Funding?	
Contract Required?	
Managed By	
Who Defines Results	
Who Accepts Results	
Who Validates Results	
Who is the End User	
Who Maintains	
Who Defines Enhancements	

What Can Go Wrong? What Do You Do?

POTENTIAL PROJECT FAILURE

- If the team does not understand the project type, the wrong framework or foundation for the project may be adopted, causing chaos or failure

WHAT TO DO

- Understand different project types and why they are different

- Ensure all stakeholders understand what type is being performed and define all needed project variables

Project Types

What To Do

Reference

You Are A Client

- Understand what type of project you are asking for
- Ensure the contractor understands different project types and their ramifications

You Are Management

- Understand what type of project the client is asking for
- Make sure the project team understands the project type
- Validate your project teams have plans aligned with the project type
- Ensure variables are defined

You Are An SE, Project Manager or Technical Lead

- Understand what type of project is being requested
- Communicate the type of project to the team
- Ensure variables are defined
- Align your plan with the project type (and development model)
- Validate your understanding with stakeholders

You Are A Team Member

- Understand different project types
- Ensure your leadership also understands
- Ask questions if you feel there is any confusion

Systems Engineering Uses

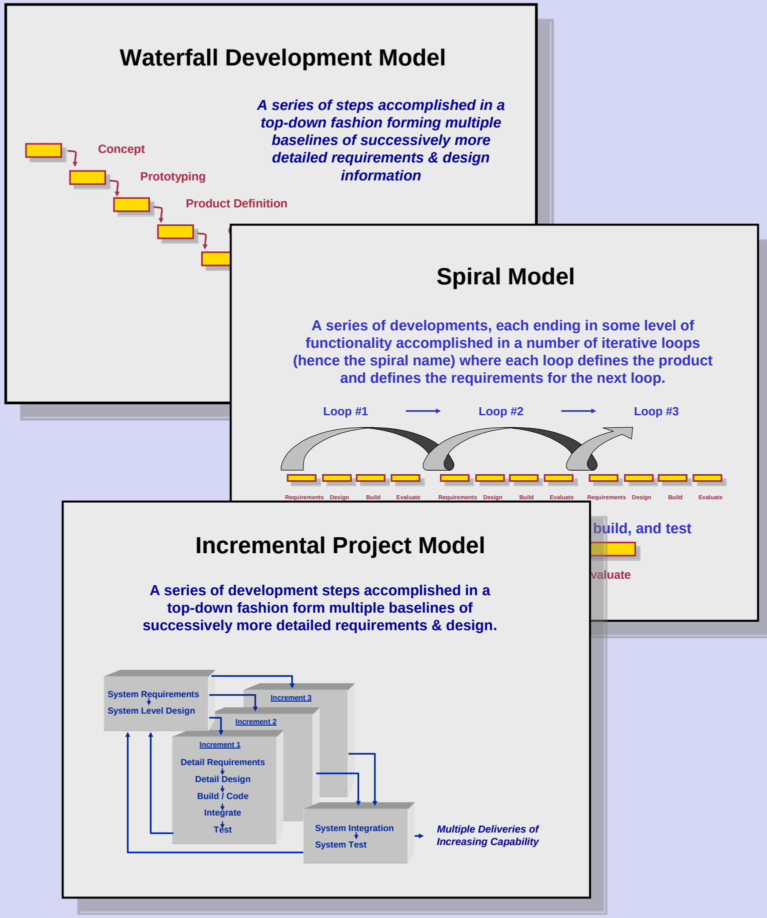
System Development Models

To Structure Development Projects

What Is a Development Model?

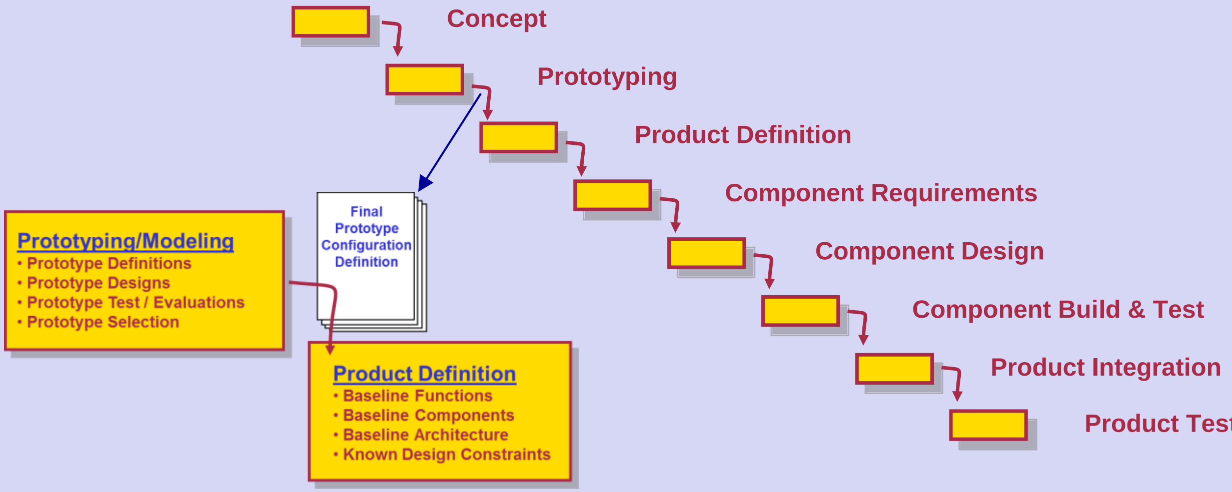
Development Models

- A pre-defined architecture of steps that define how project work is sequenced
- Common Development Model Types:
  - Waterfall
  - Spiral
  - Incremental
  - Combinations of the above

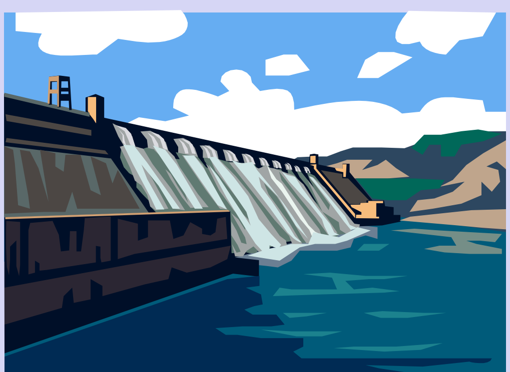


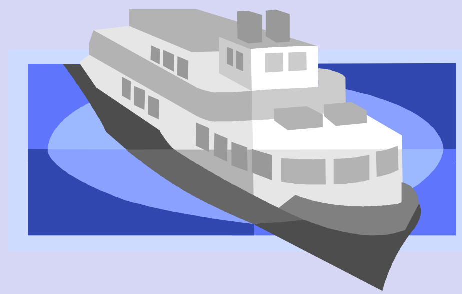
Different Project Types Do Require Different Development Models

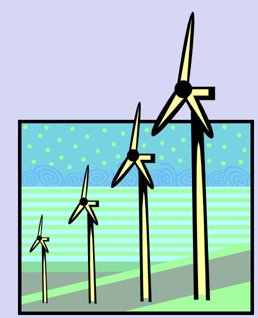
Waterfall Developments



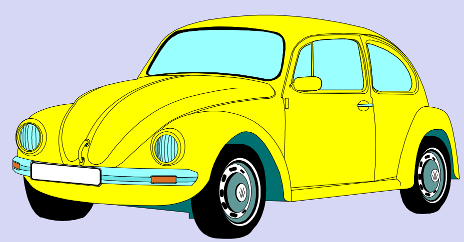
Construction



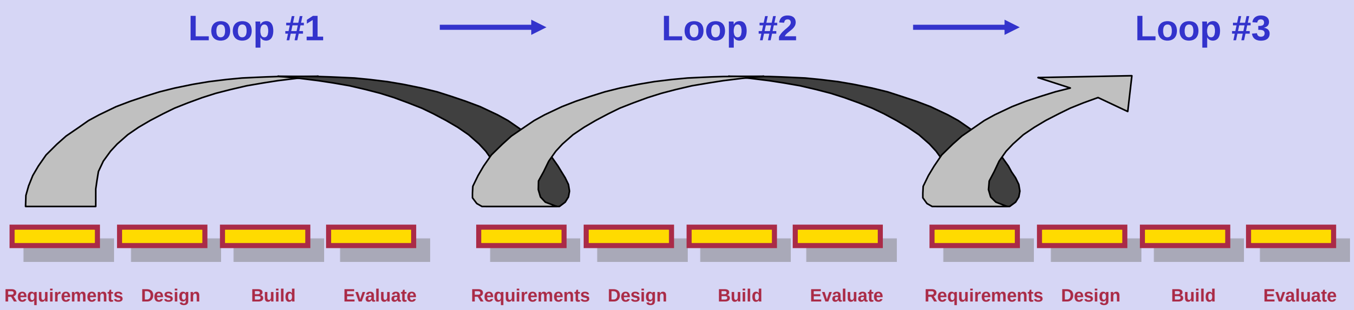


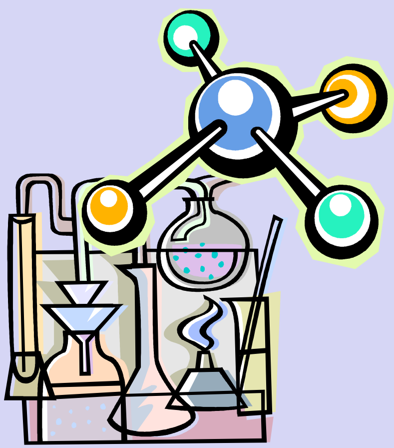


Complex Products and Systems



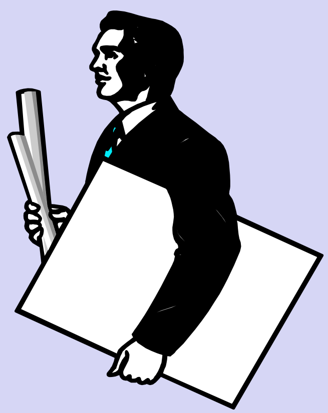
Spiral Developments





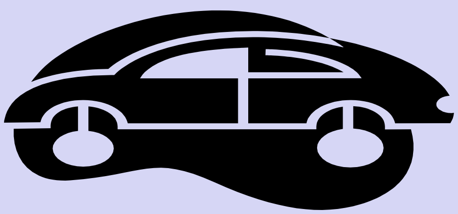
New Drug Development

Marketing Campaign Development

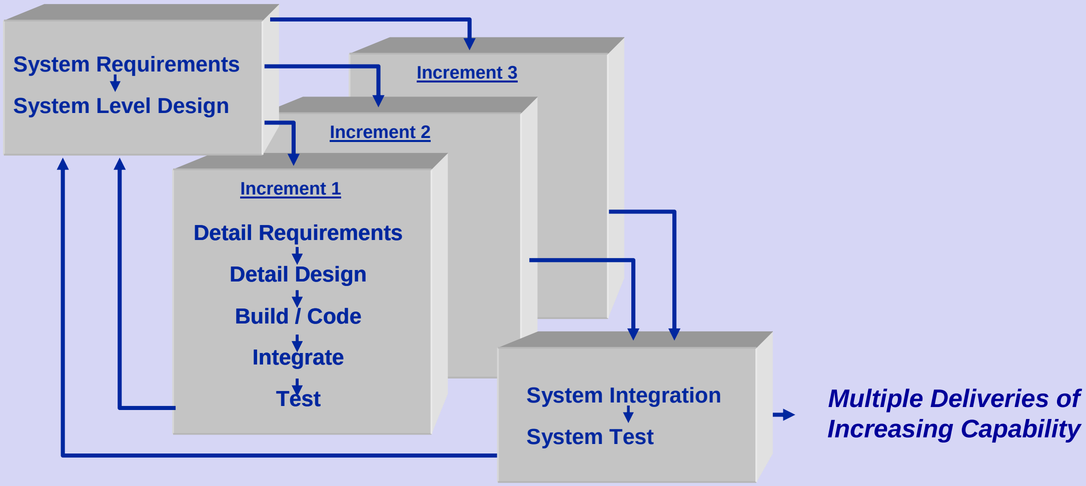


System User Interface Development

Hybrid Vehicle Prototypes



Incremental Developments



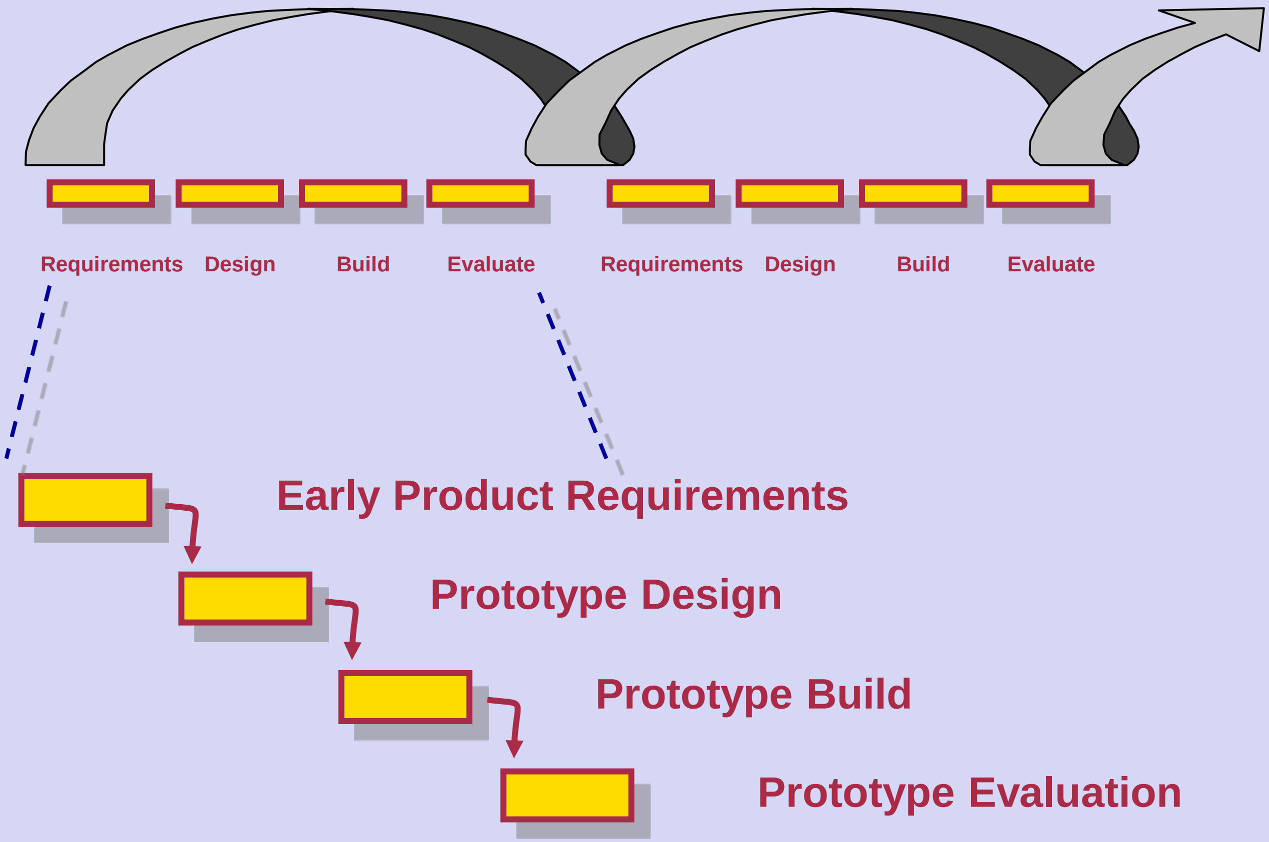
ERP System Installation -
Ledger module is followed by a material management module and so on ......

Building Complex
One building now, another later, ...

Web Site Development
Core site now, site additions as time and needs progress

Model Combinations

Example 1

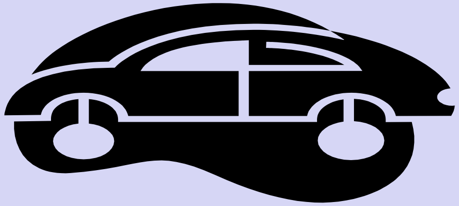


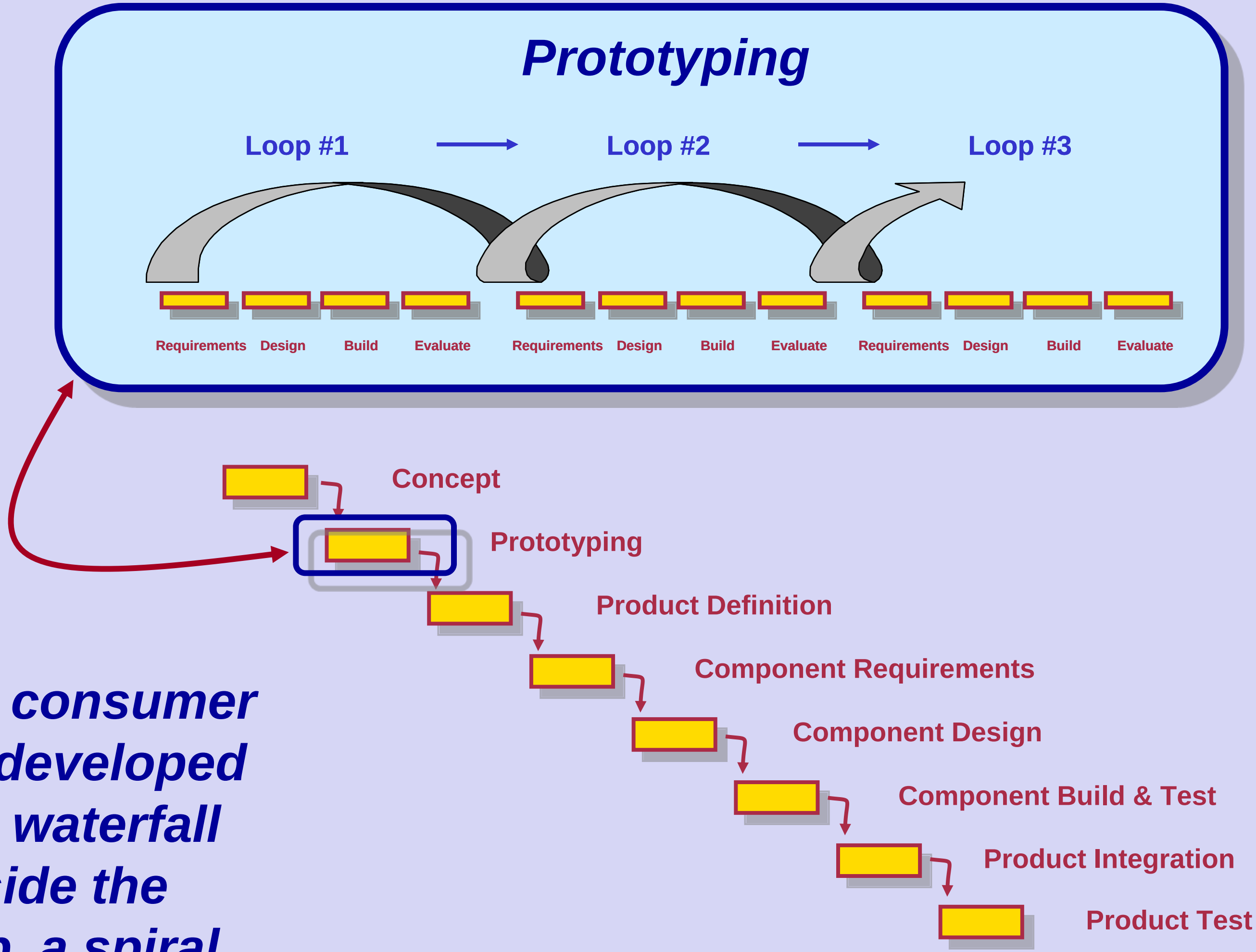
In this situation a simple waterfall model is used inside a spiral model. Each spiral starts with some requirements leading to a prototype build and evaluation which leads to the requirements for the next generation prototype.

Model Combinations

Example 2

Hybrid Vehicle Prototypes

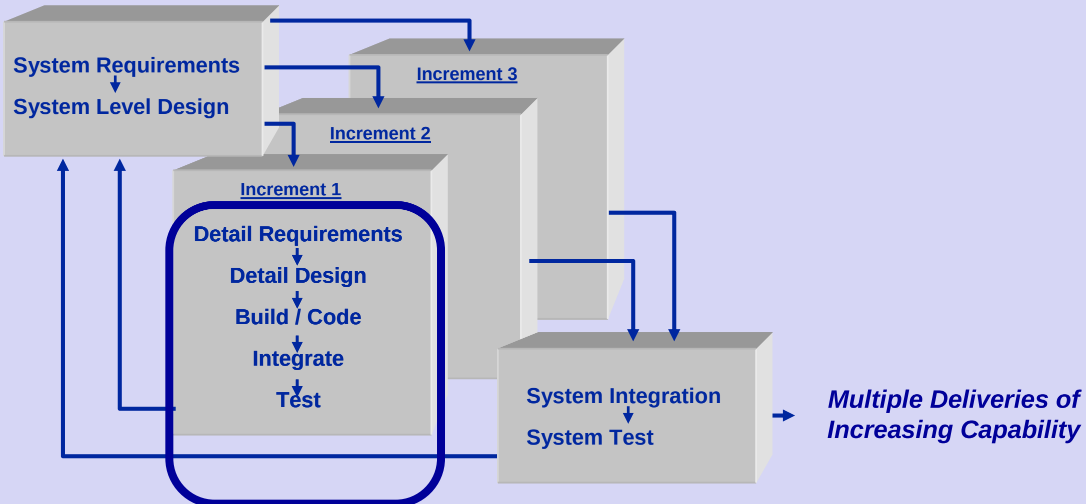




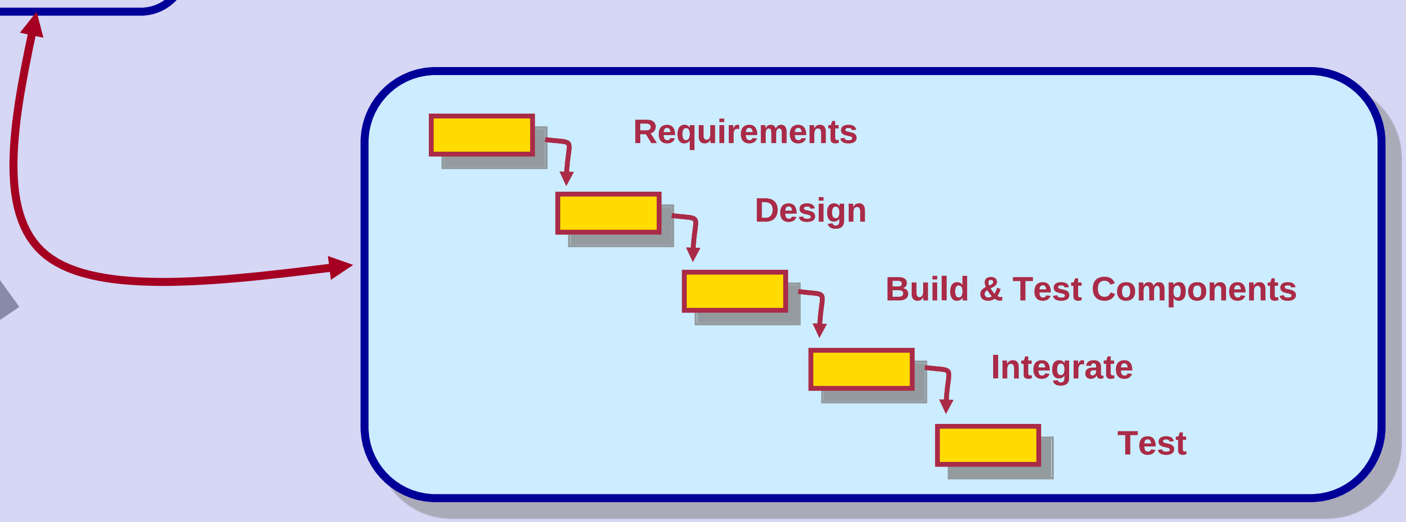
In this example, a consumer product is being developed with a long term waterfall model and inside the prototyping step, a spiral model is employed

Model Combinations

Example 3



In this example, one increment of an overall incremental development uses a waterfall sequence for development of that respective increment

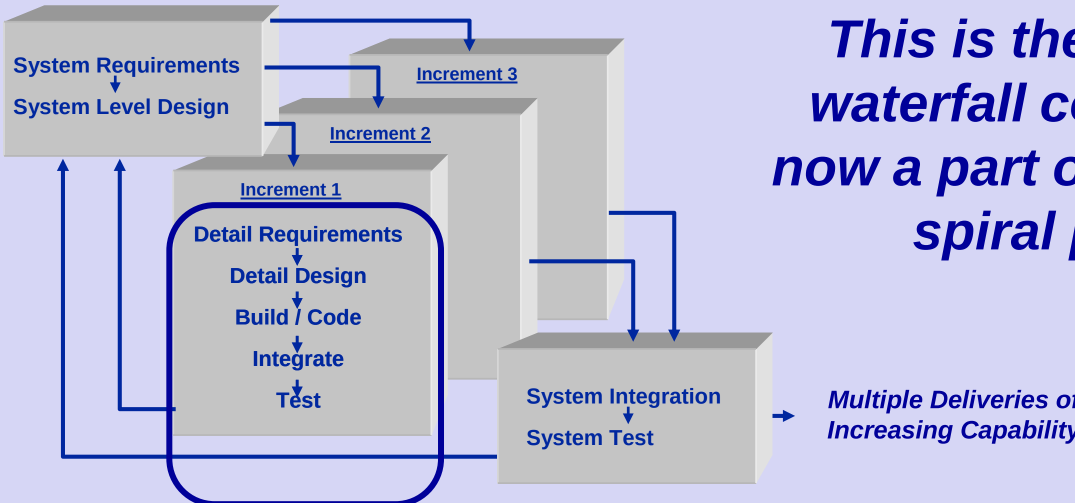


Web Site Development

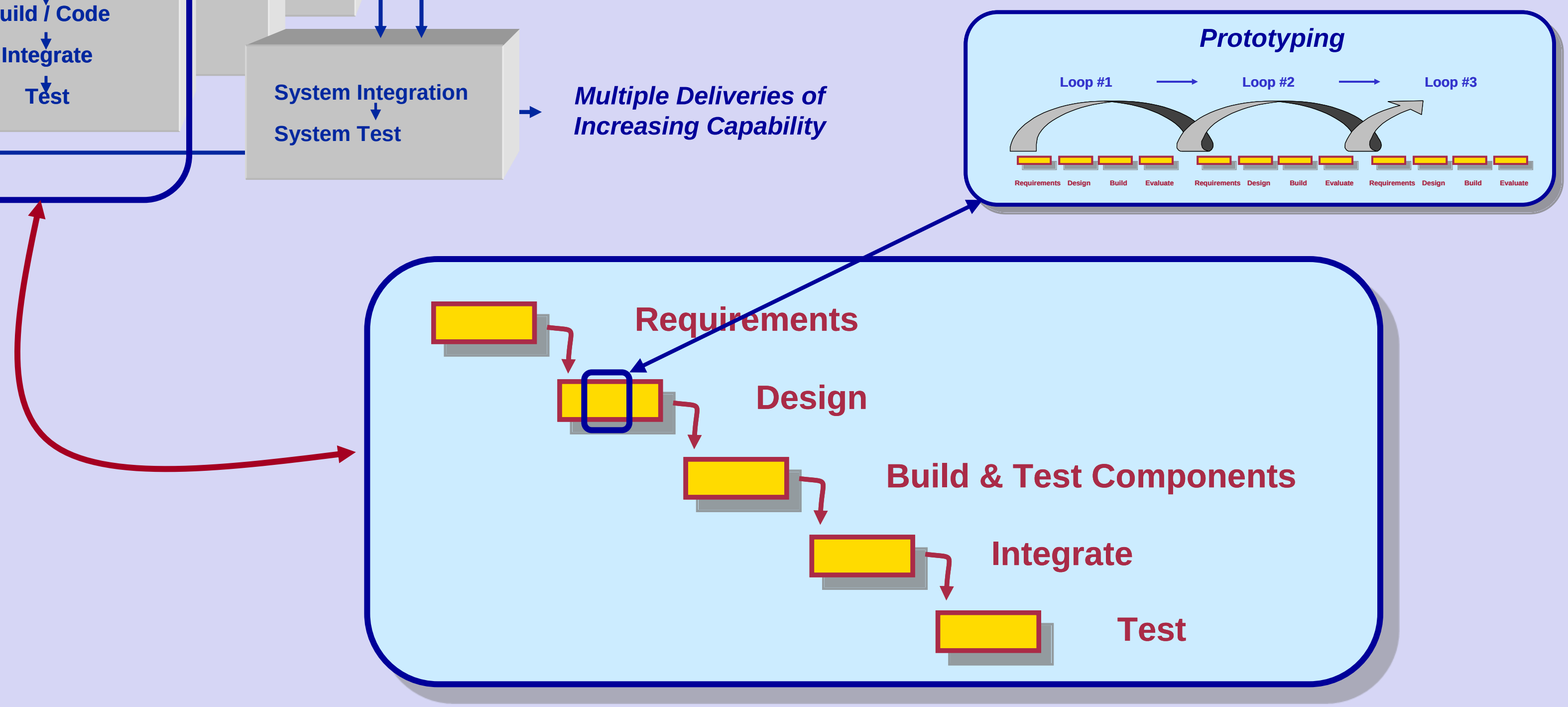
Core site now, site additions as time and needs progress

Model Combinations

Example 4



This is the same as the incremental / waterfall combination example #3, but now a part of the design effort includes a spiral prototyping component



Development Model Applications

Waterfall

Applicable where requirements are ***known***

Spiral

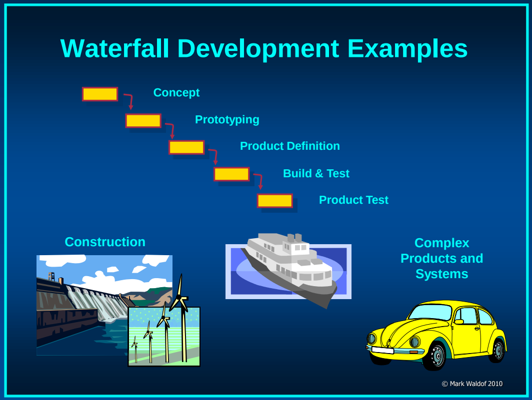
Applicable where requirements are ***NOT known***

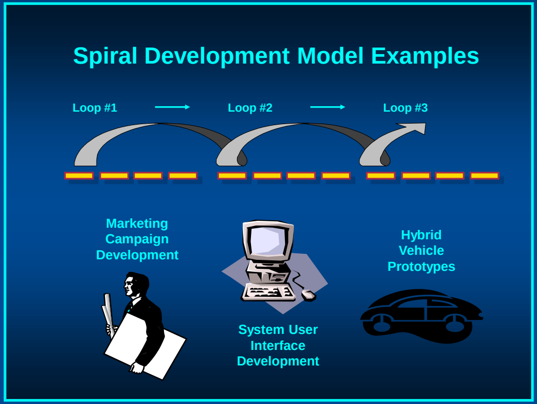
Incremental

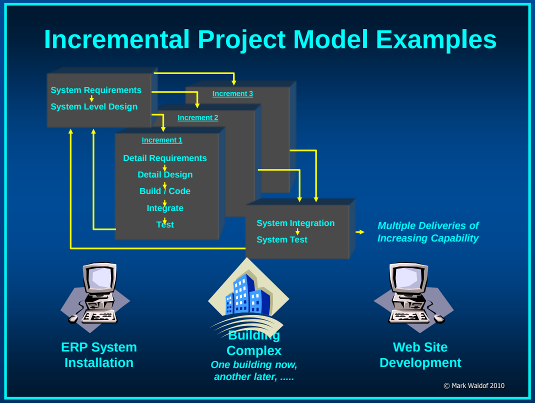
Applicable where ***some*** *requirements are* ***known***

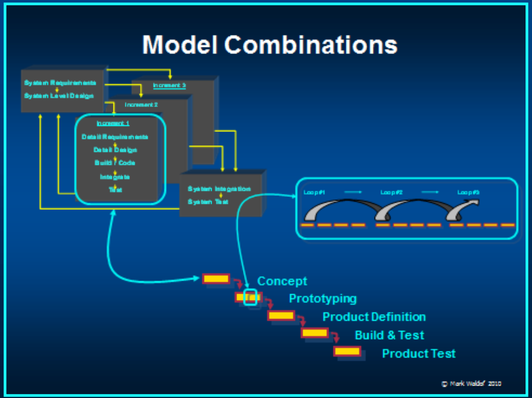
Combination Models

(variable)

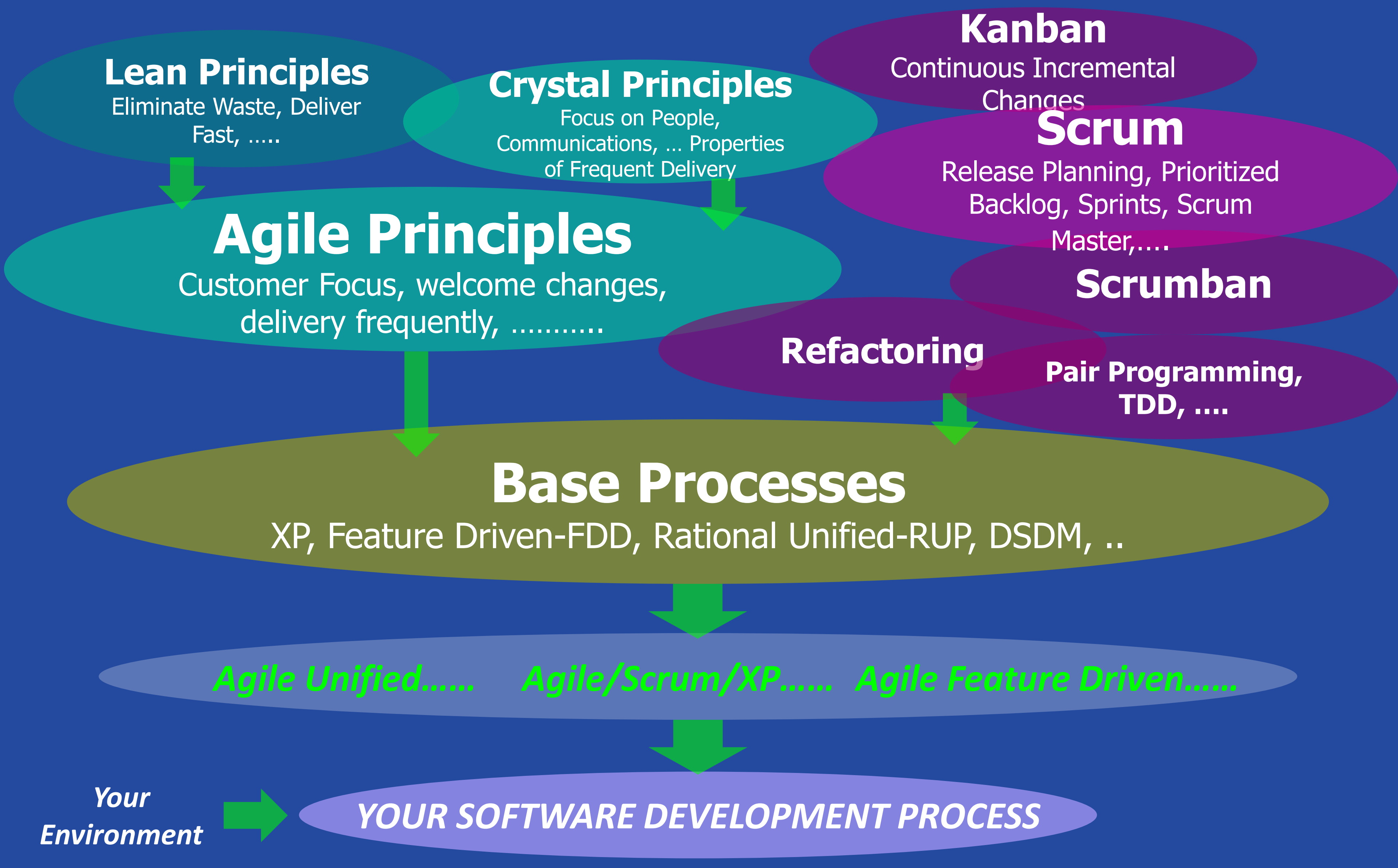




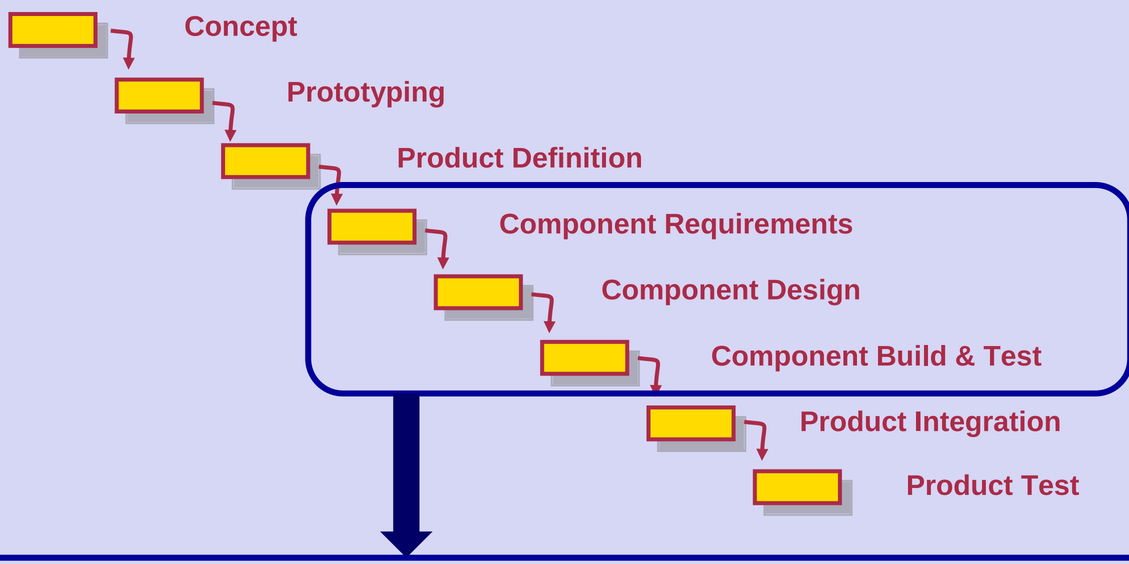




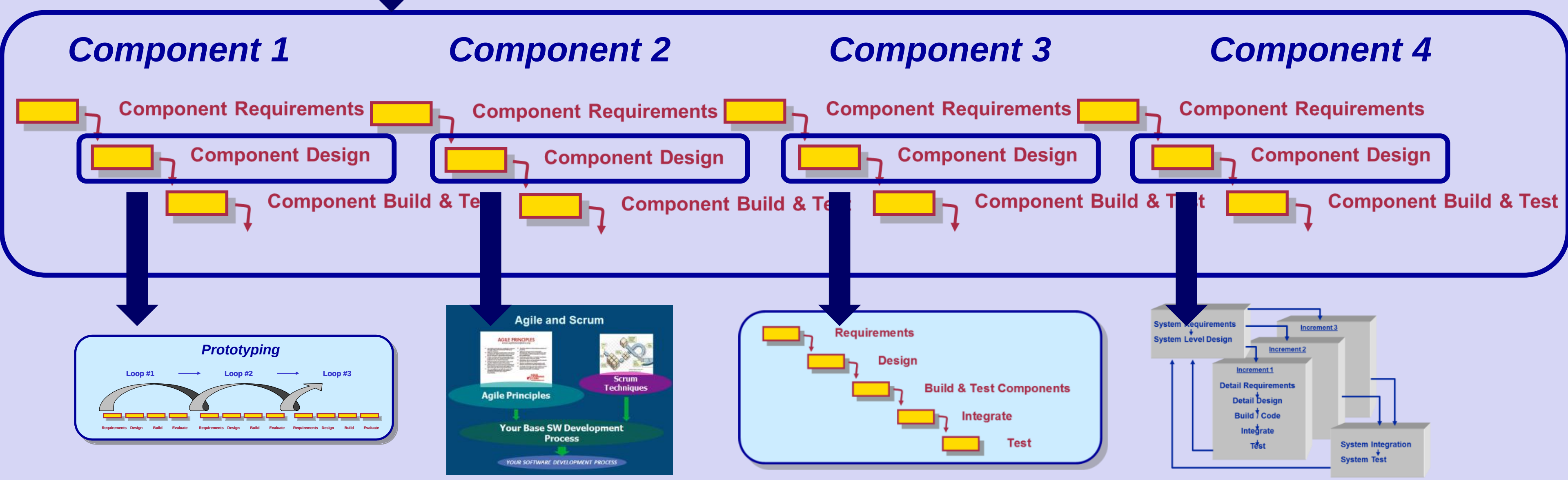
Sw Dev Method Combinations



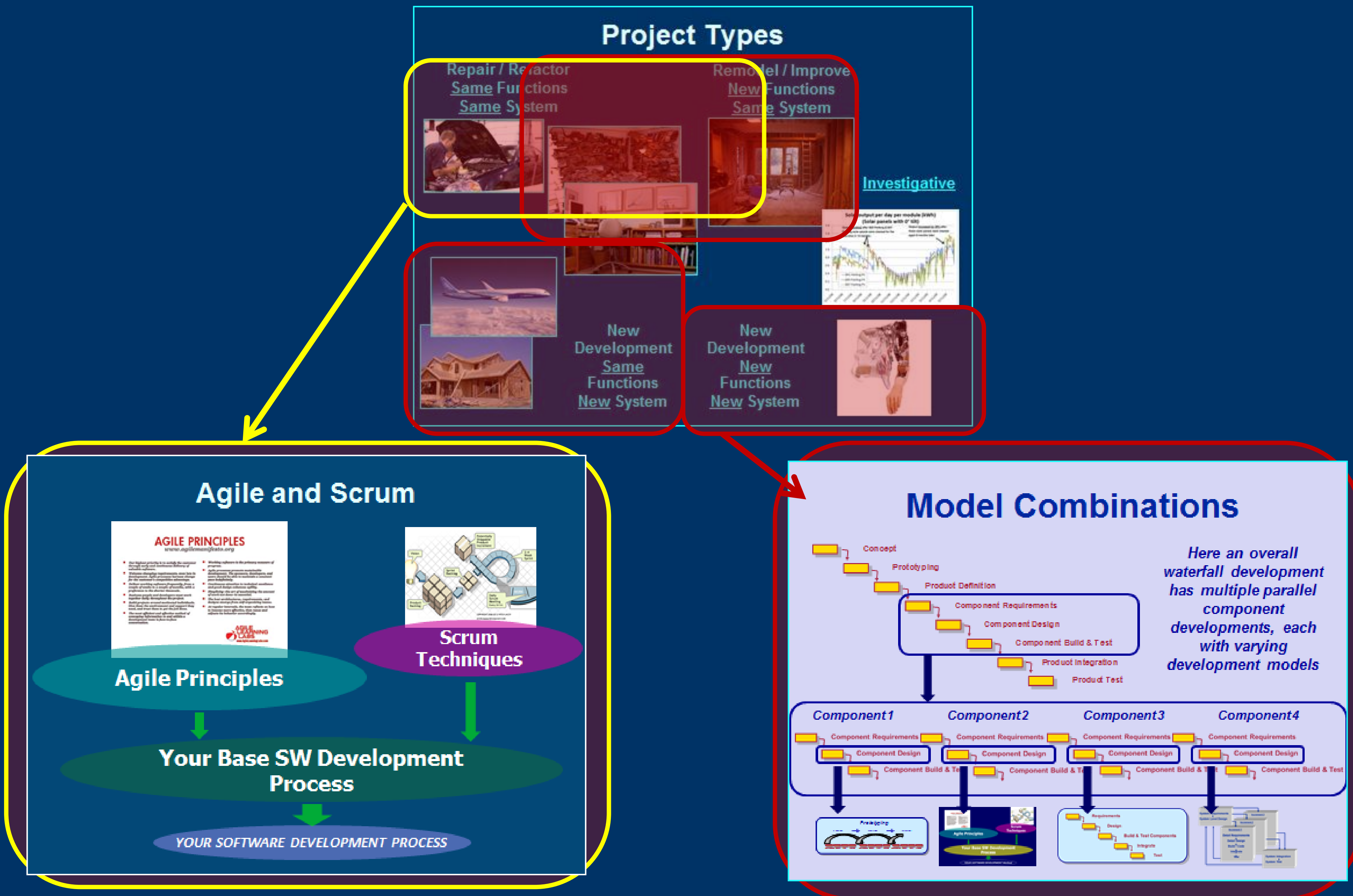
Model Combinations



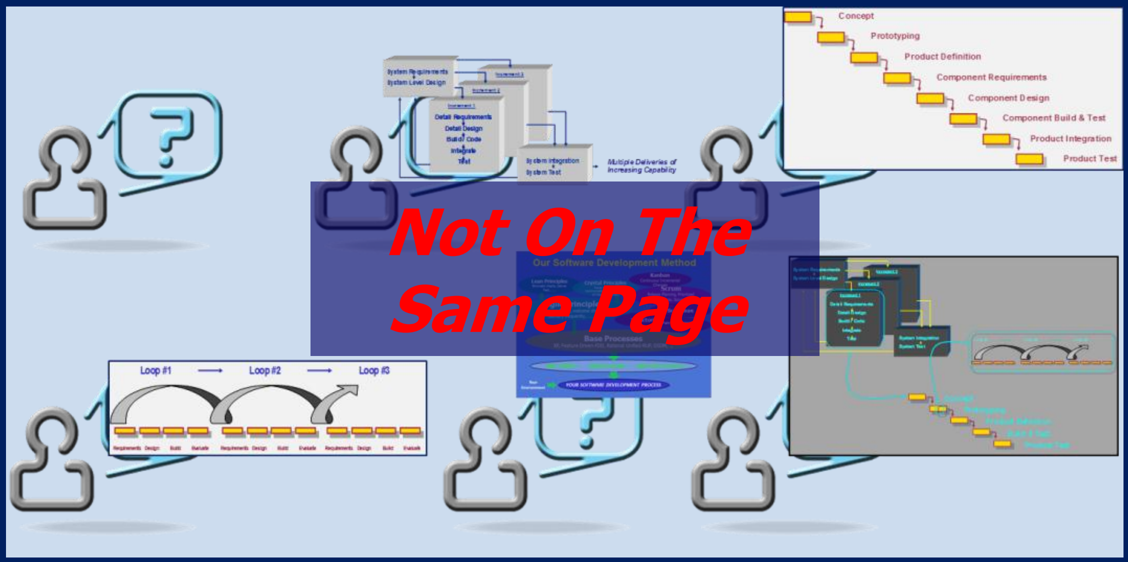
Here an overall waterfall development has multiple parallel component developments, each with varying development models



Project Types Related to Development Models



Why Define A Development Model?

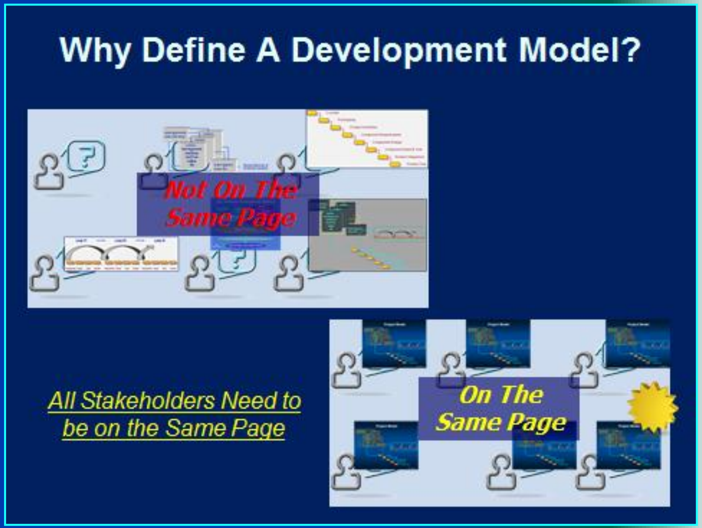


<u>*All Stakeholders Need to be on the Same Page*</u>

What Can Go Wrong? What Do You Do?

POTENTIAL PROJECT FAILURE

- Lack of common knowledge of a development model confuses the team
- No Model = No Real Plan
- The wrong development model may result in failure or cost and schedule overruns



WHAT TO DO

- Understand different development models and their applications
- Work with the team to select the best model
- Educate others on the importance of defining a model of development

Development Models

What To Do

Reference

You Are A Client

- Understand different models
- Ask your contractor to explain the development model they will use and why

You Are Management

- Understand different models
- Educate development teams
- Have teams present their intended model at project start for discussion and refinement
- Consider an independent review to validate model selection for large projects

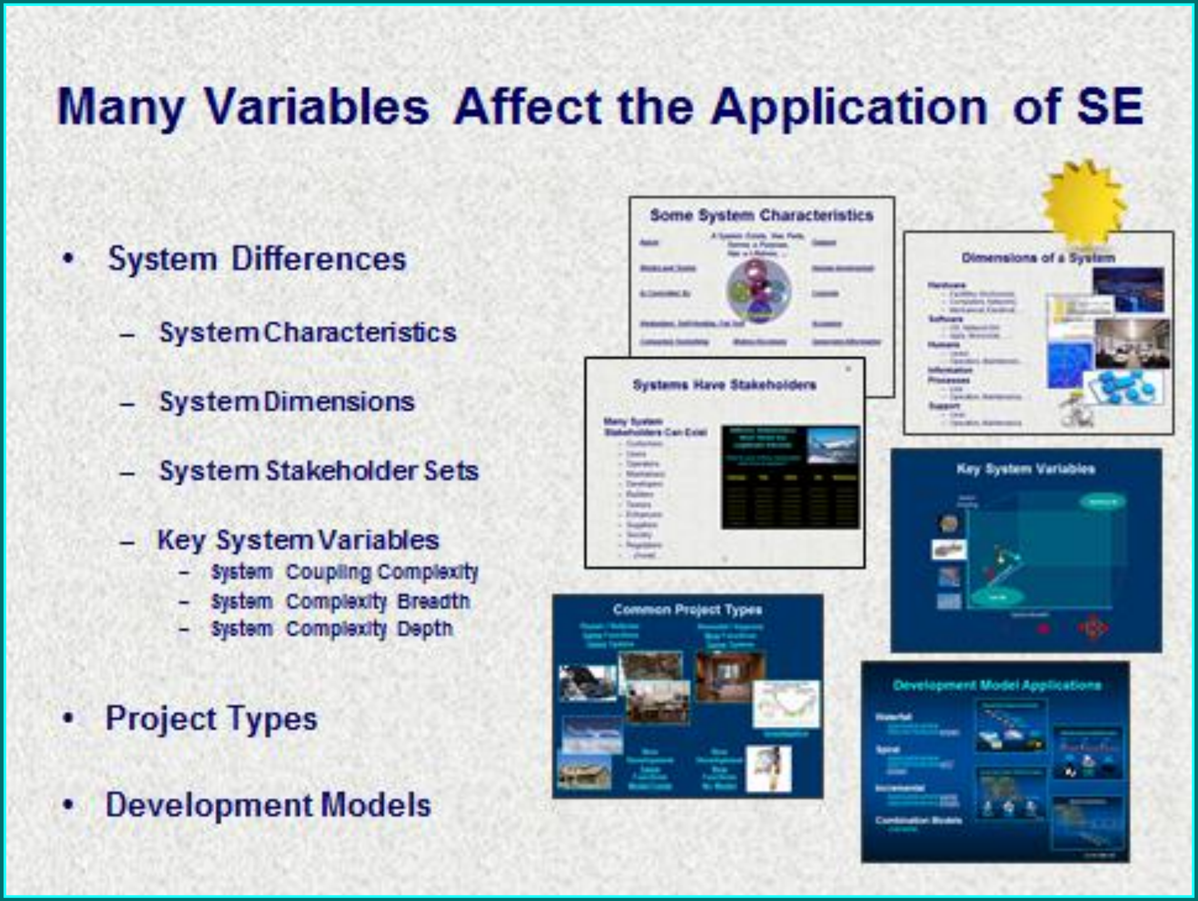
You Are A Project Manager or Technical Lead

- Understand different models
- Educate your team
- Work with team to define the right model
- Communicate to stakeholders
- Use the model as a basis for directing the development

You Are A Team Member

- Understand models
- Ask questions on what model the project will use
- Determine how your work fits within the model

The Application of SE Must be Matched to Specific System Variables, Project Type and Development Model

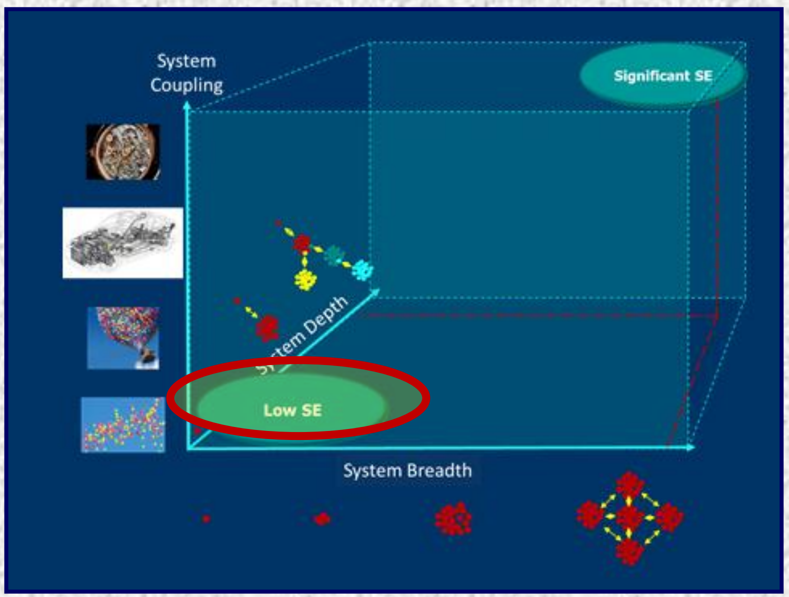


System Variables & Project Types

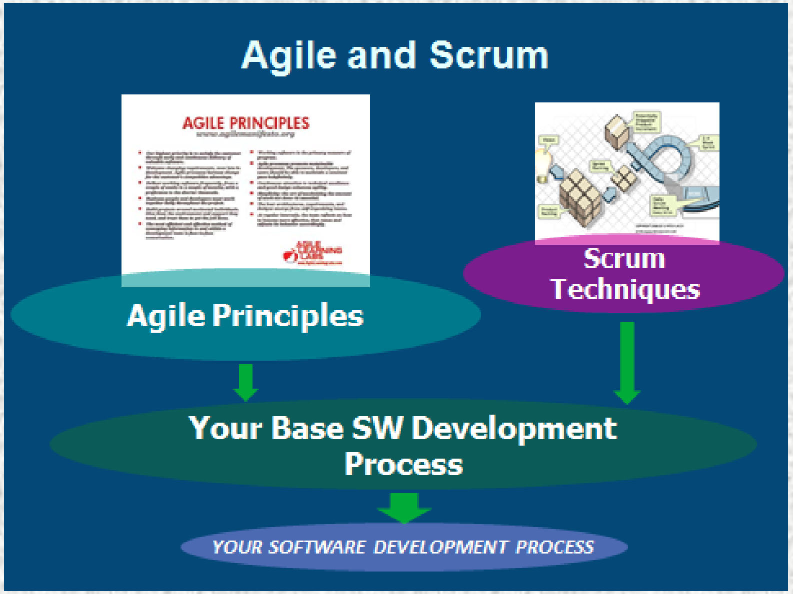
System Variable → Project Type ↓		System Dimensions Involved	Coupling	System Breadth and Depth, Complexity	Stakeholder Complexity
Repair / Refactor		*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
Remodel		*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
New Development *Model Exists*		*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
New Development *No Model*		*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
Investigative		*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*

LOW Need for Systems Engineering

System Variable Project Type	System Dimensions Involved	Coupling	System Breadth and Depth, Complexity	Stakeholder Complexity
Repair / Refactor	*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
Remodel	*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
New Development *Model Exists*	*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
New Development *No Model*	*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
Investigative	*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*

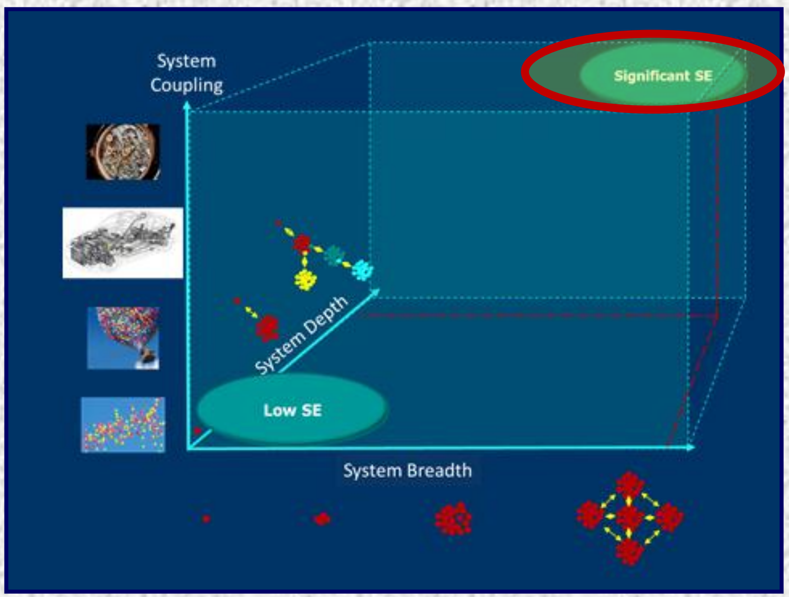


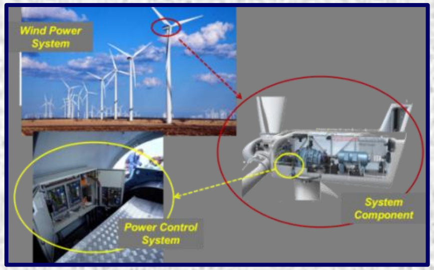
- Repair or Refactor
- Low Coupling
- Low Complexity
- Limited Stakeholder Set
- Fast Cycle Dev Model
- **Low SE Maybe Required**

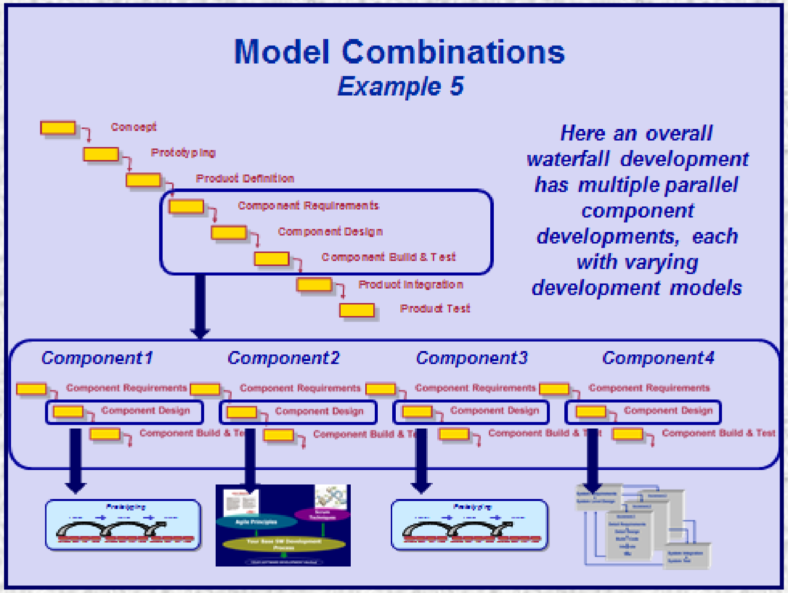


HIGH Need for Systems Engineering

System Variable → / Project Type ↓	System Dimensions Involved	Coupling	System Breadth and Depth, Complexity	Stakeholder Complexity
Repair / Refactor	*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
Remodel	*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
New Development *Model Exists*	*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
New Development *No Model*	*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*
Investigative	*One Several All*	*None or Low High Very High*	*Low Medium High*	*Low Medium High*







- "Blank Sheet of Paper"
- High Coupling
- High Complexity
- Complex Stakeholder Set
- Complex Development Model
- **High Level of SE Required**

What Can Go Wrong?
What Do You Do?

POTENTIAL PROJECT FAILURE

- An inadequate level of SE applied to a given project type will almost certainly end in a level of chaos

WHAT TO DO

- Understand project types, system differences and development models
- Match project type, system differences and level of SE for a given project
- Educate your team and ensure stakeholders understand what level of SE is needed and why

Systems Engineering Addresses All Phases in the

System Lifecycle

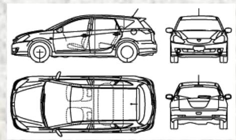
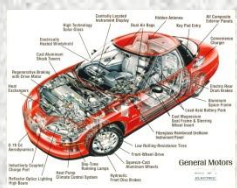
What is the System Lifecycle?

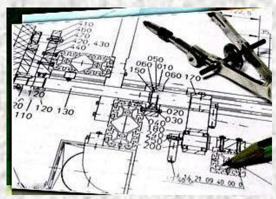
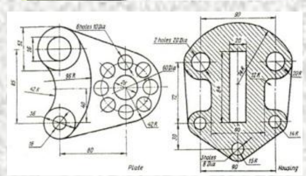
Answer: Cradle to grave lifetime of a system

System Lifecycle Phases

- New Need or a Problem to Solve Defined
- Alternative Solutions Identified Concepts
- Solution Selection (System Concept)
- System Development (System Level Design, Components, Integration)
- System Delivery, Install, Train, Support
- System Use, Operation and Maintenance
- System Upgrade
- System Disposal

Automobile "System" Lifecycle



Market Analysis



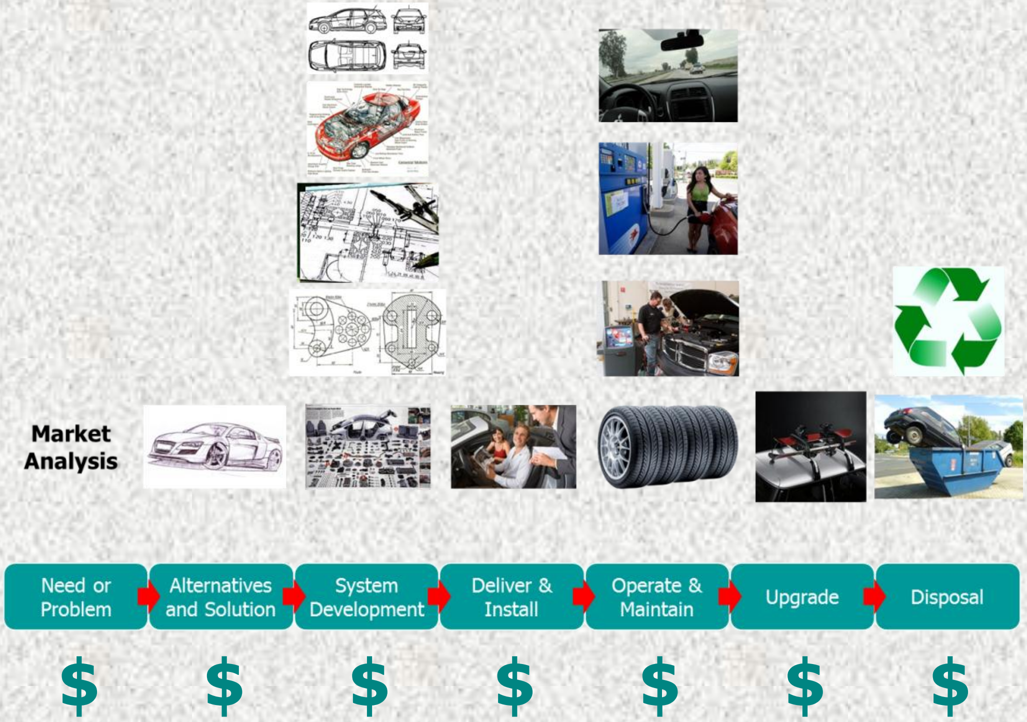



Need or Problem → Alternatives and Solution → System Development → Deliver & Install → Operate & Maintain → Upgrade → Disposal

How Systems Engineering Addresses the Lifecycle
Example: **Automobile Lifecycle Costs**

When a system is developed, a systems engineer must address costs for all phases of the lifecycle

- Cost to Develop
- Cost to Produce
- Cost to Deliver
- Cost to Operate
- Cost to Maintain
- Cost to Dispose

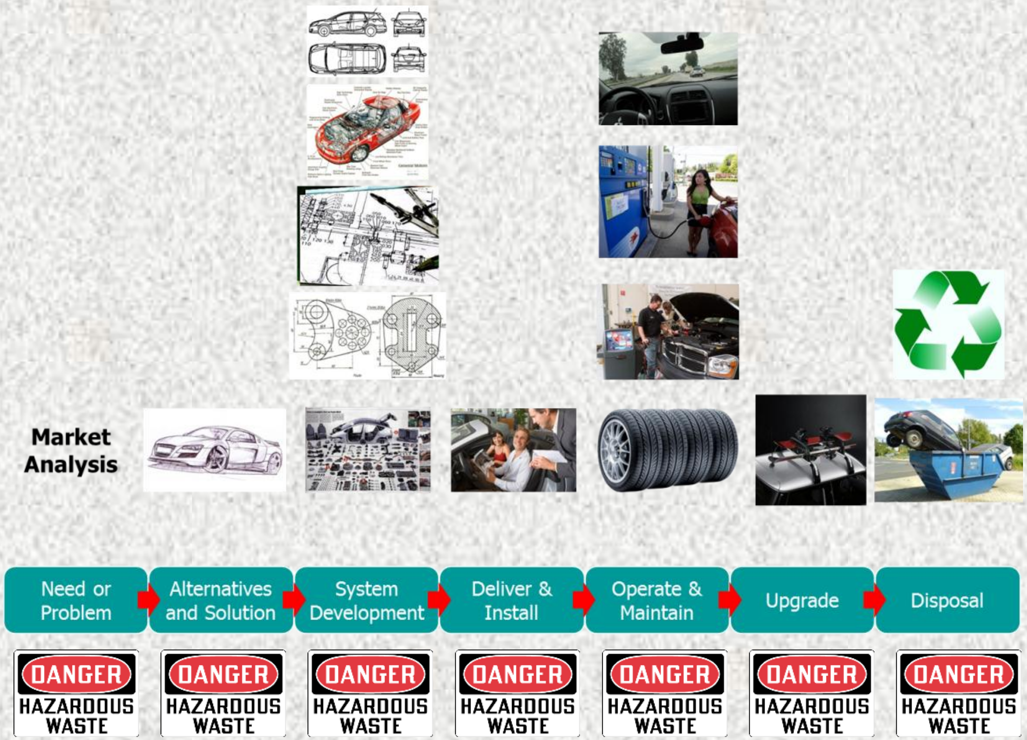


How Systems Engineering Addresses the Lifecycle

Example: **Automobile Lifecycle**
<u>Environmental Impacts</u>

When a system is developed, a systems engineer must address <u>environmental impacts</u> for all phases of system existence

- Production Impacts
- Storage Impacts
- Operation Impacts
- Maintenance Impacts
- Disposal Impacts



How Systems Engineering Addresses the Lifecycle

Example: **Design Must Consider all Future Phases**

When a system is developed, a systems engineer must address <u>all future system phases</u>

How the system will be….

- Used, Operating Modes
- Consumable Replenishment
- Stored, Transported
- Maintained
- Upgraded
- Disposed

How Systems Engineering Addresses the Lifecycle

Example: **Design Must Consider all Future Phases**

When a system is developed, a systems engineer must address <u>all future system phases</u>

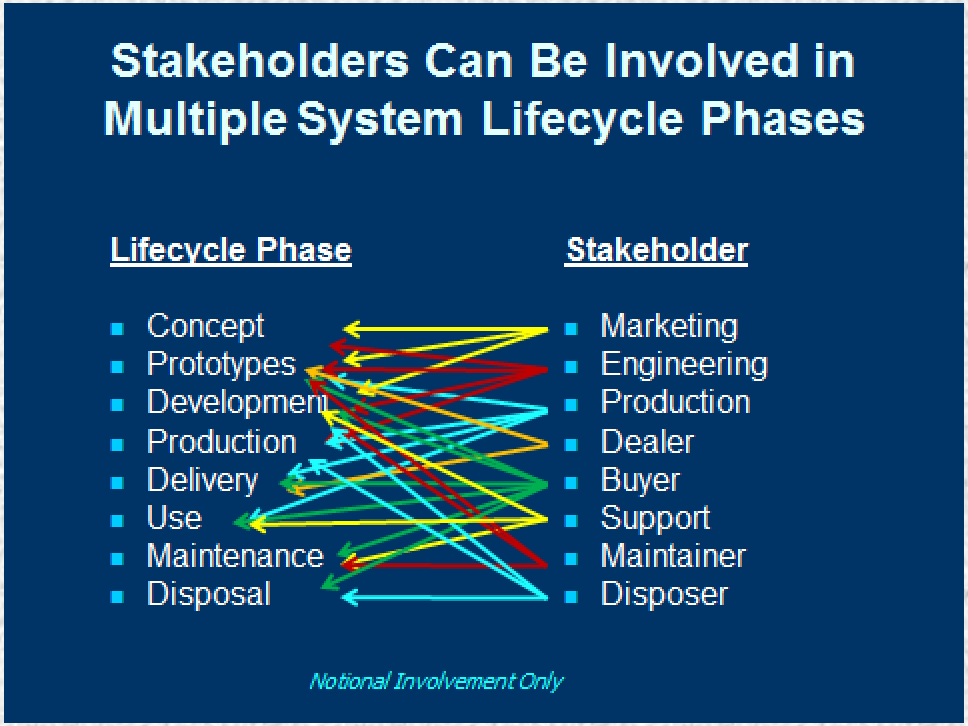
How the system will be….

- Used, Operating Modes
- Consumable Replenishment
- Stored, Transported
- Maintained
- Upgraded
- Disposed

Example: Development Must Consider all **Stakeholder Needs**

When we addressed system stakeholders earlier, we mentioned that any individual stakeholder can be involved in multiple lifecycle phases.

Systems Engineering is the key vehicle for ensuring the right stakeholders are involved in the appropriate lifecycle steps.



System Lifecycle
What To Do

Reference

You Are A Client

- Understand the system lifecycle process
- Understand where your project fits in the lifecycle
- Ensure your contractor and you develop a design meeting all lifecycle needs

You Are A Project Manager or Technical Lead

- Understand the system lifecycle
- Educate teams on the system lifecycle process
- Ensure your addresses all appropriate lifecycle phases

You Are Management

- Understand the system lifecycle
- Educate teams on the system lifecycle process
- Ensure each project understands what phases their development must address

You Are A Team Member

- Understand the system lifecycle
- Ask questions of leadership about lifecycle step influence on your project’s efforts

Systems Engineering Performs and Manages the….

System Development Process

What Is System Development?

Answer: The process that…..

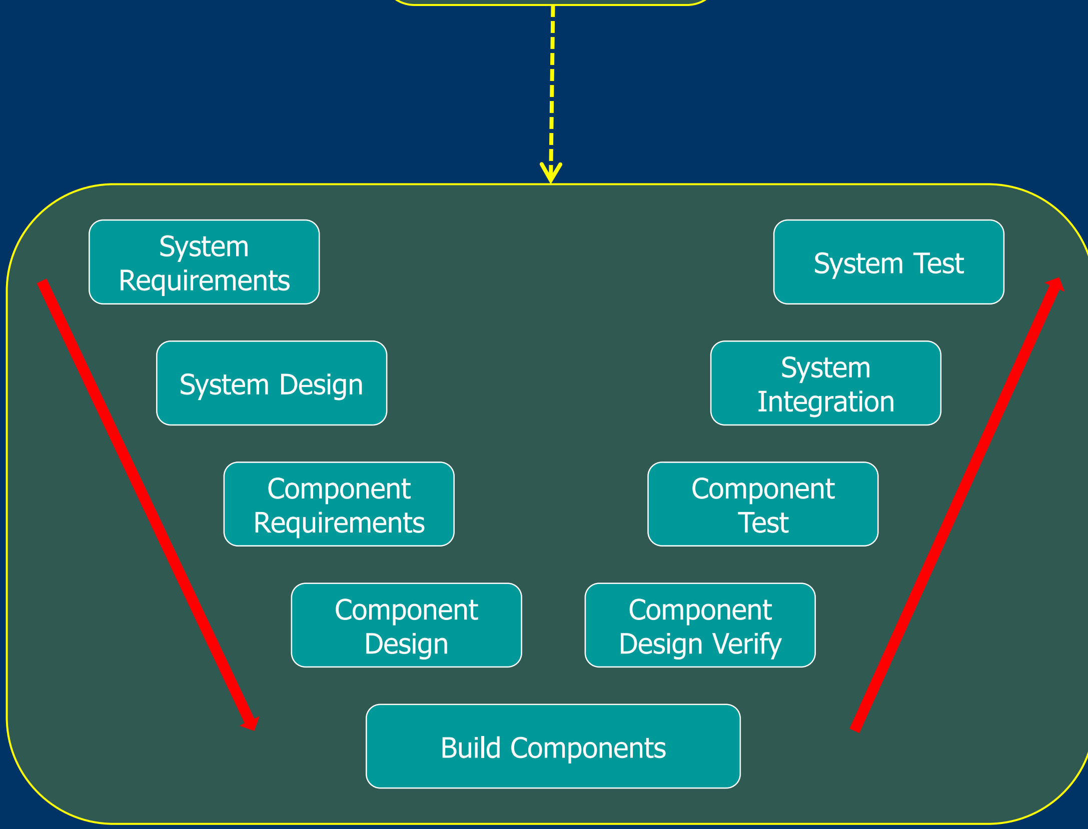
- Starts with a known solution and requirements, then
- Develops the given system

System Development Steps

- System Level Requirements
- System Level Design, Identification of System Components
- Allocation of Requirements to System Components
- System Component Design
- System Component Build and Test
- Integration of System Components
- Test of the Complete System
- Delivery and Support

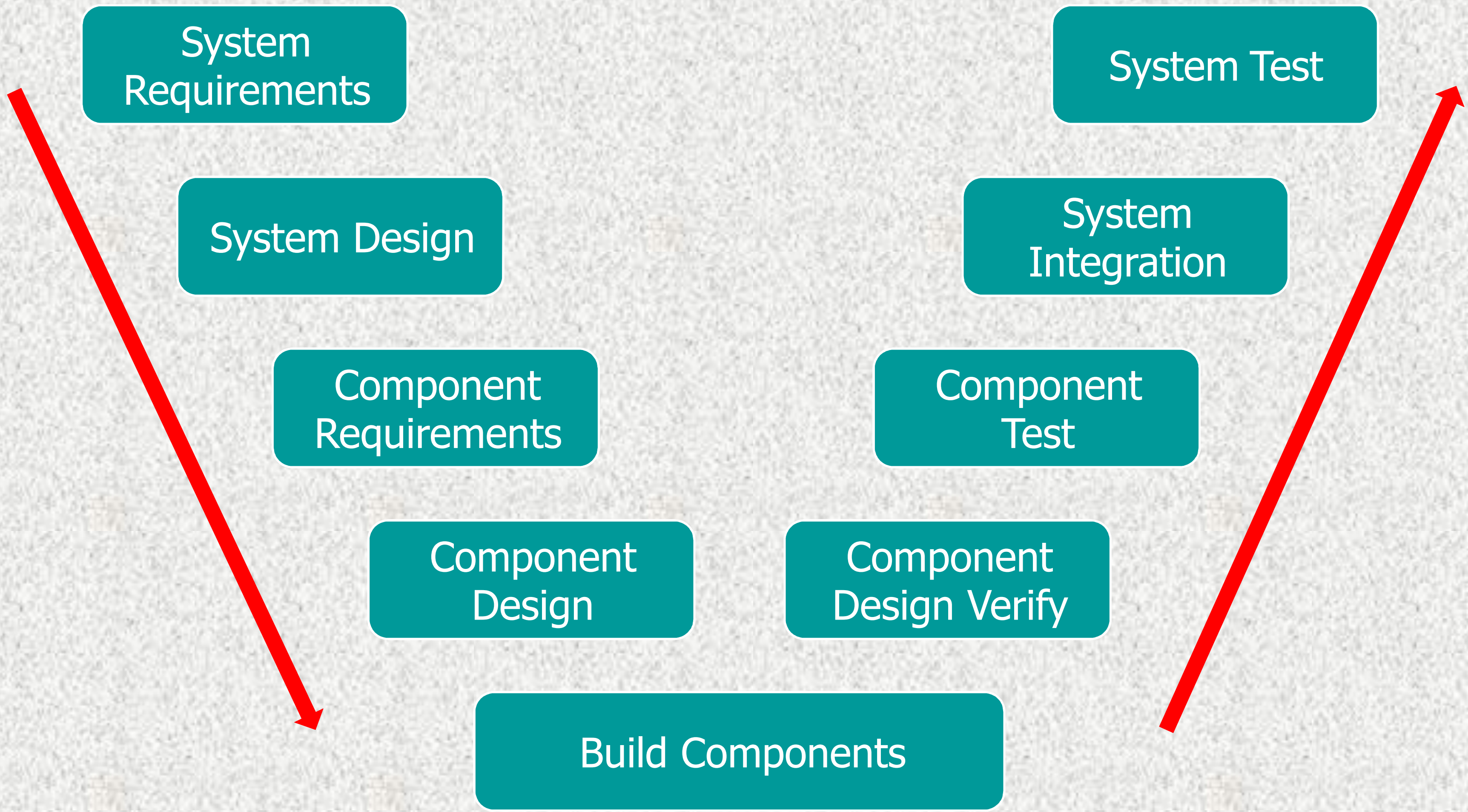
System Lifecycle

System Development

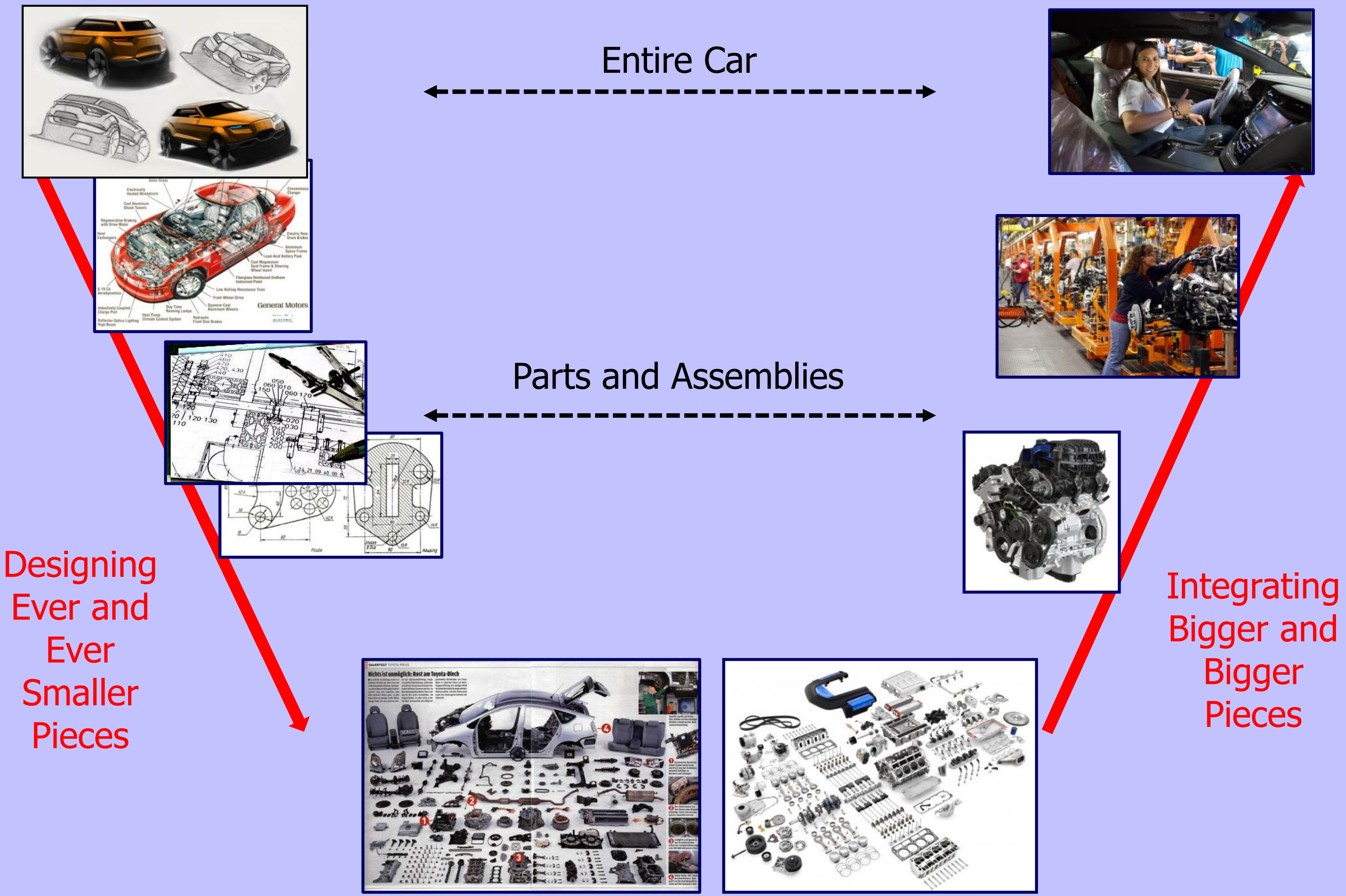


System Development Process

System Development Process

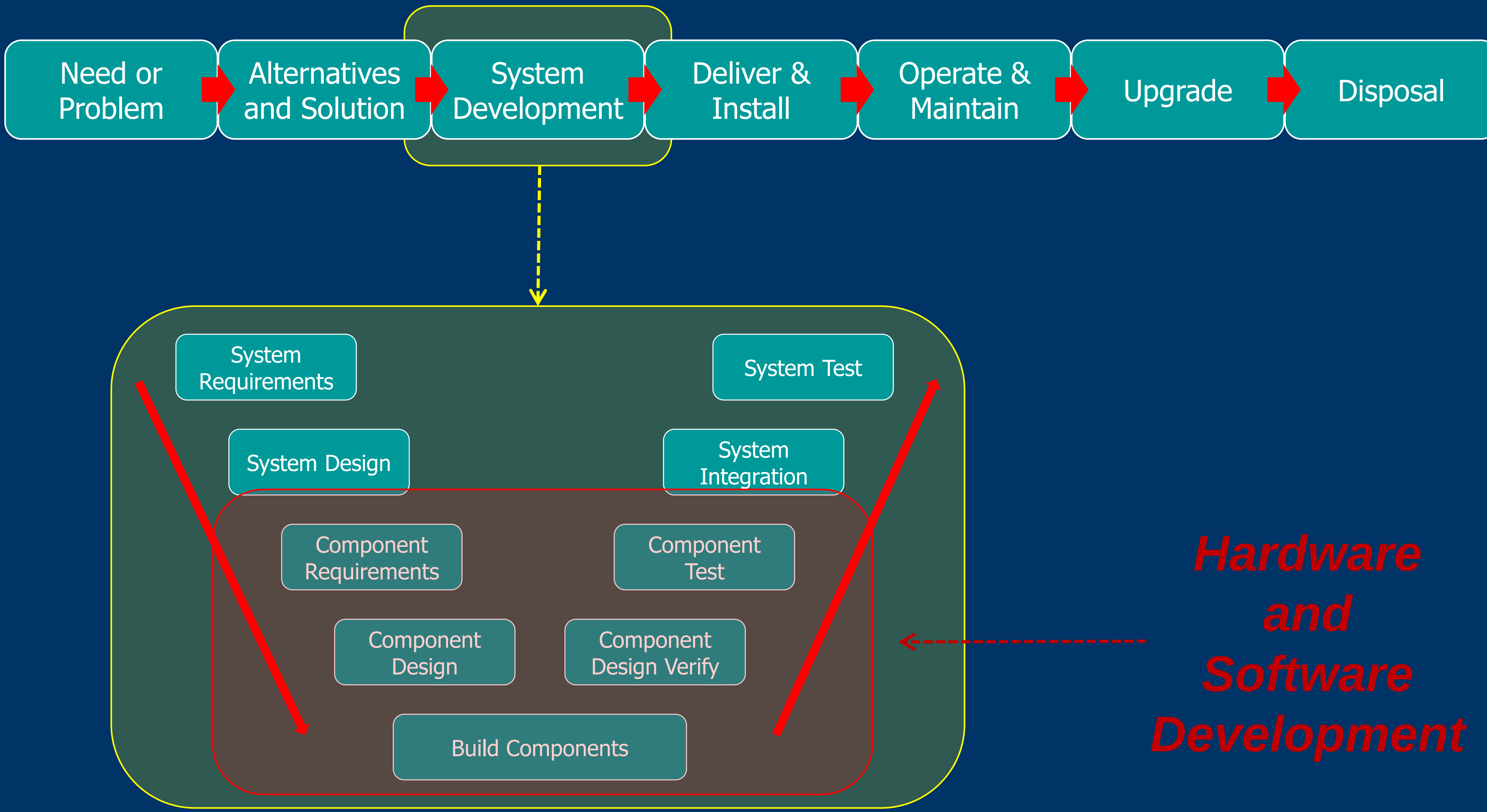


Automobile System Development

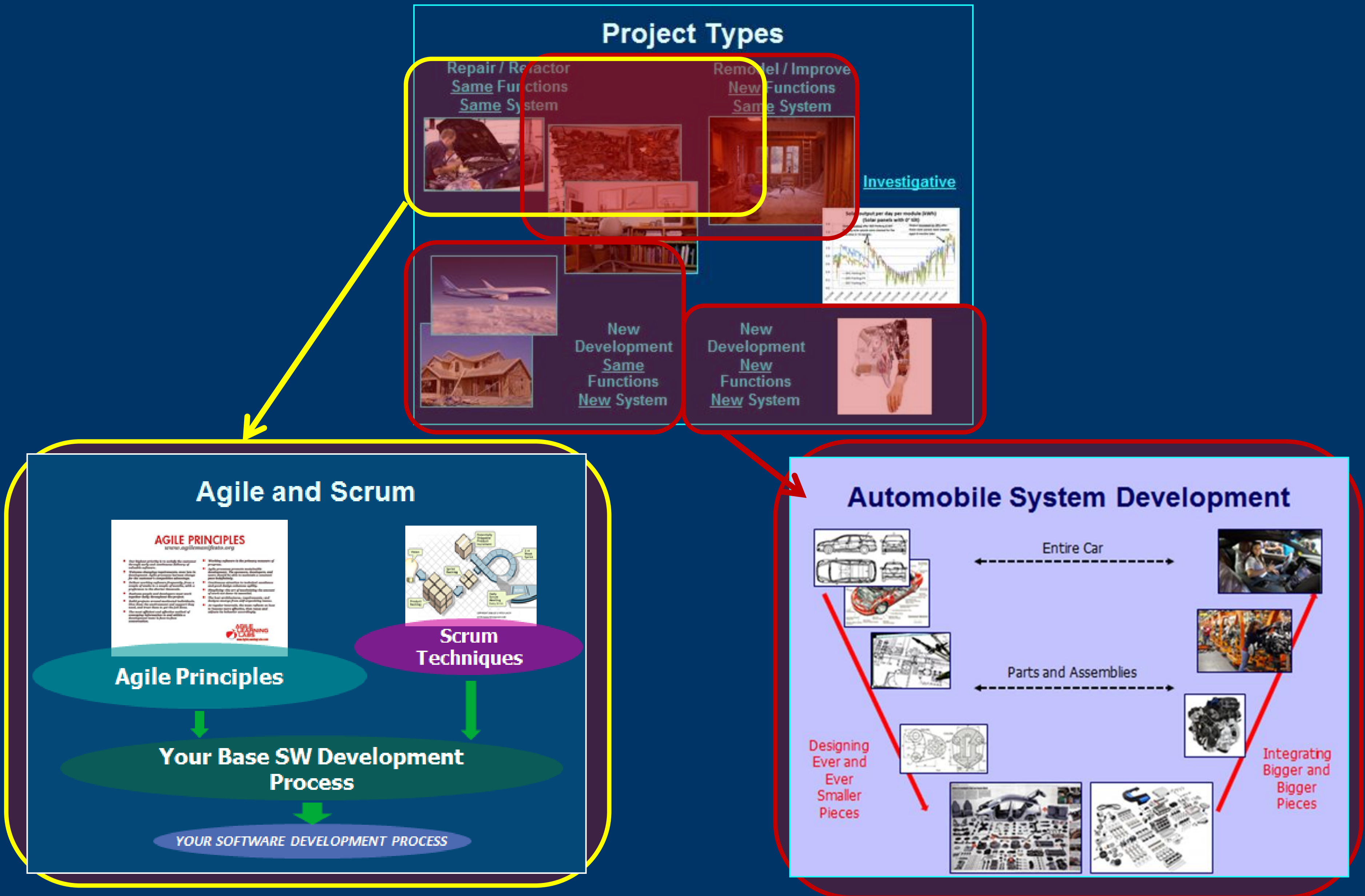


System Lifecycle

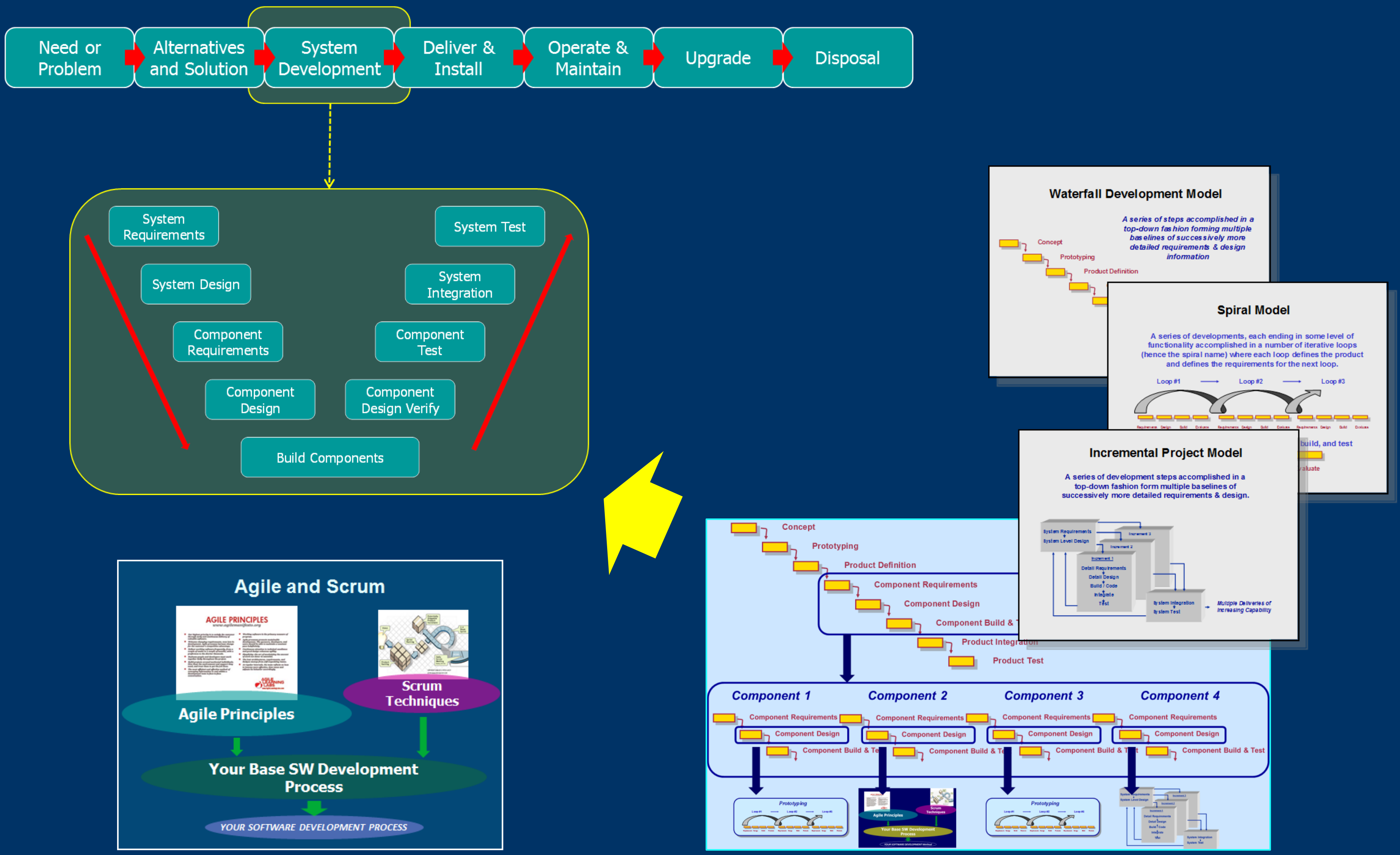
System Development



Project Types and System Developments



A System Development May Employ Many Internal Development Models



System Development Process

What To Do

Reference

You Are A Client

- Understand the system development process
- Understand it applies to some project types
- Ensure your contractor understands how it applies

You Are Management

- Understand the system development process
- Understand it applies to some project types
- Ensure your project teams understand project types and how the system development process applies

You Are A Project Manager or Technical Lead

- Understand the system development process
- Understand how it applies to some project types
- Ensure your project team understands how the system development process applies

You Are A Team Member

- Understand the system development process
- Ask questions to ensure the PM and team also understand

Systems Engineering Performs

Critical Initial System Lifecycle Steps

How A Project is Started Will Critically Affect Its Success

A Famous Quote

"Most projects in trouble today, <u>started out that way</u>"

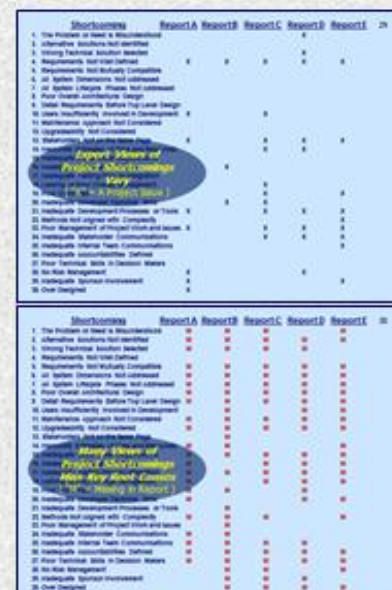
To keep any project out of trouble requires many things including a sound understanding of systems engineering core concepts

What Does this Mean?

Multiple views *are needed to fully understand challenges to projects and what is needed.*

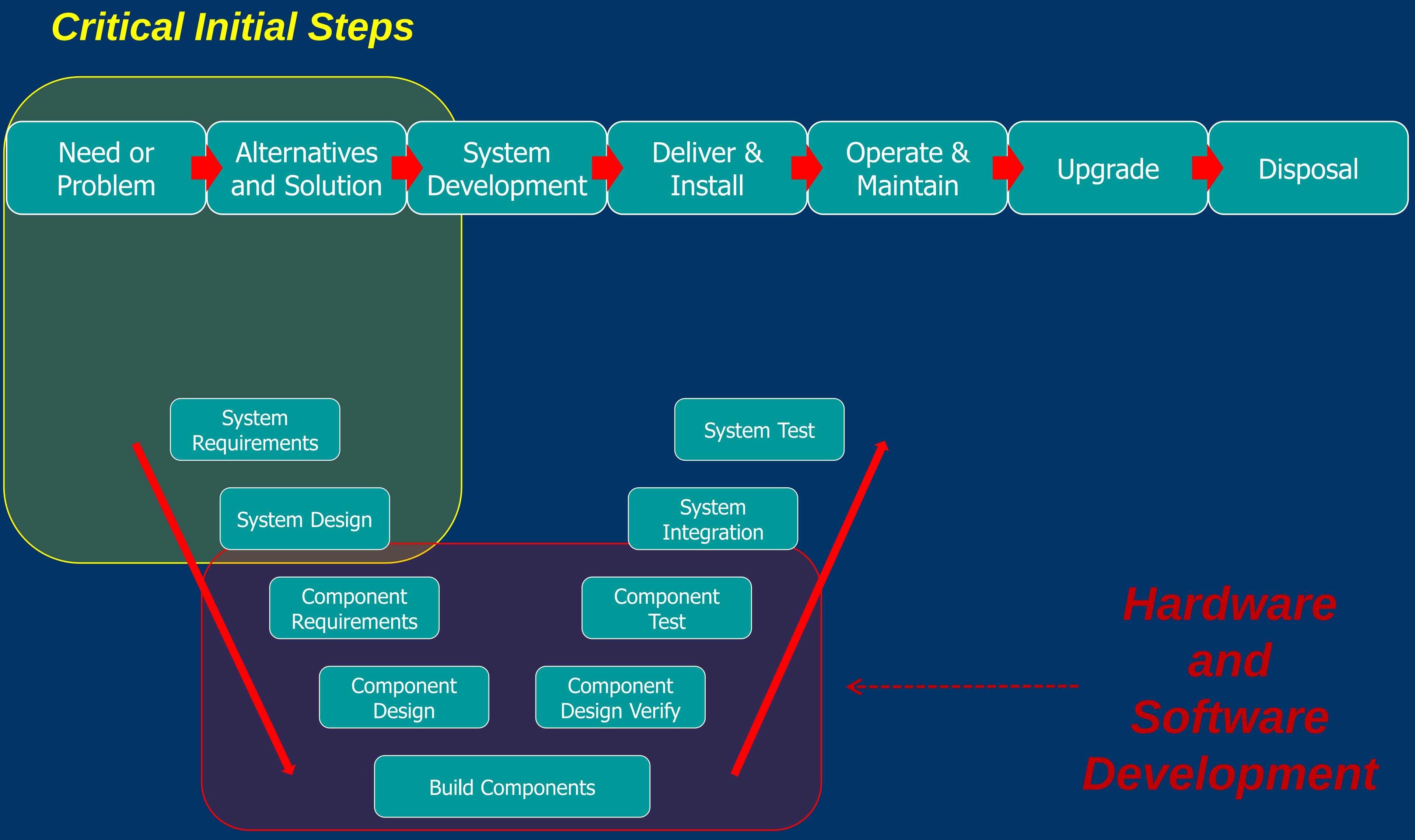
Systems Engineering is related to virtually all *technical issues projects face.*

With ***knowledge of SE Core Concepts****, project teams can* ***avert many causes*** *of project challenges and failure.*

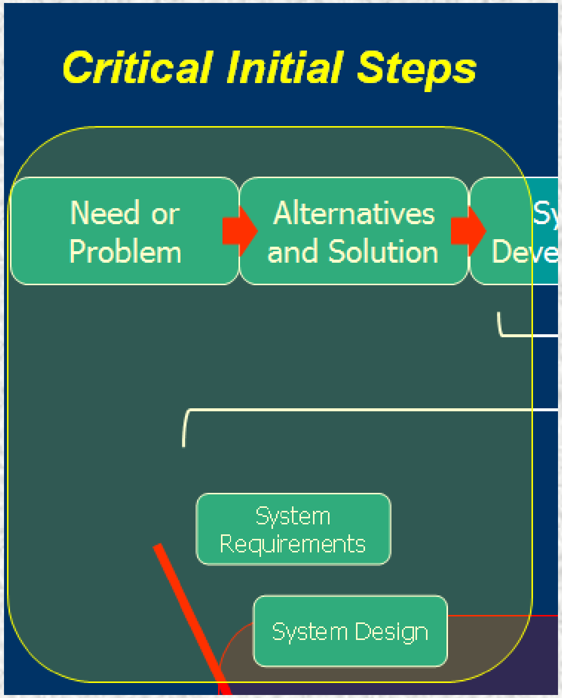


180

System Lifecycle



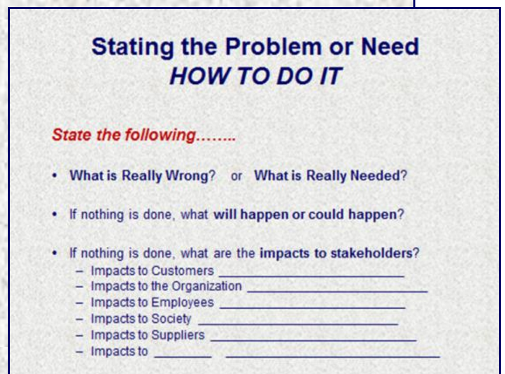
Initial System Development Steps



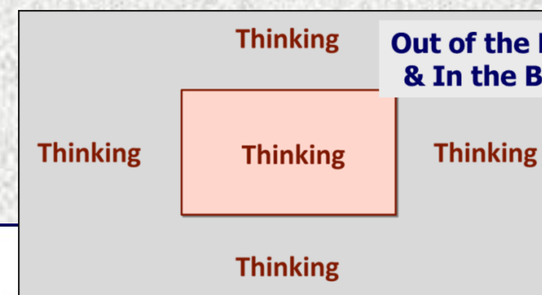
1) State and Validate the Problem or Need
2) Identify Solution Concept Alternatives
3) Identify Solution Boundaries
4) Select the Right Solution
5) Description and Feasibility Check
6) Develop Level Requirements
7) Design the System

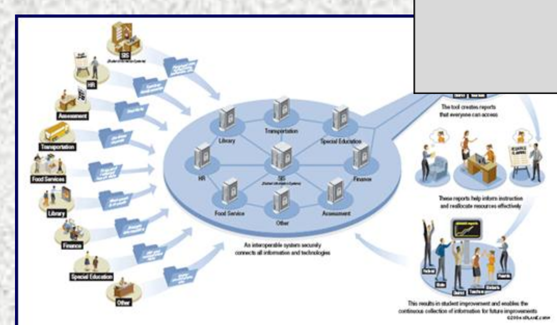
Early System Development Steps

1) State and Validate the Problem or Need



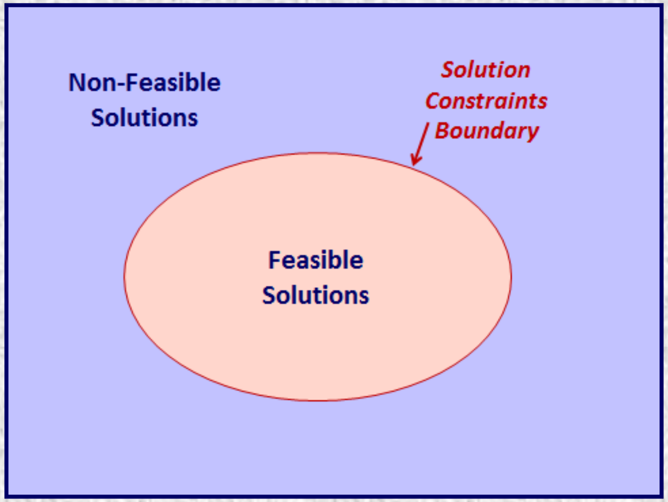
2) Identify Solution Concept Alternatives





Early System Development Steps

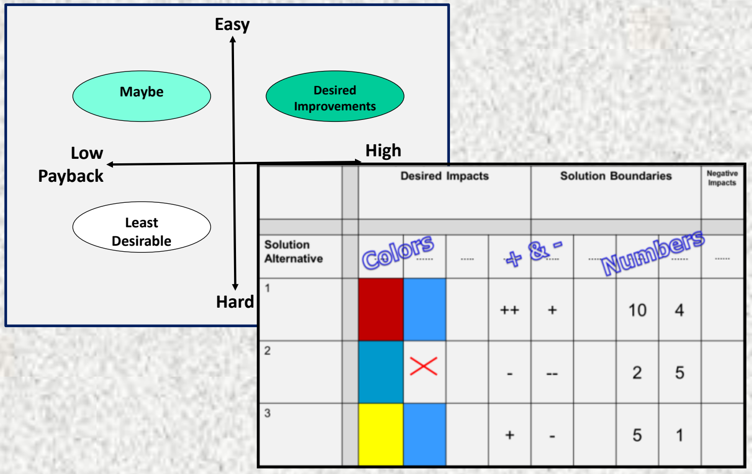
3) Define Solution Boundaries



Criteria and Weighting

- *Cost*
- *Performance*
- *Acceptability*
- *Technology*
- *…..*
- *…..*

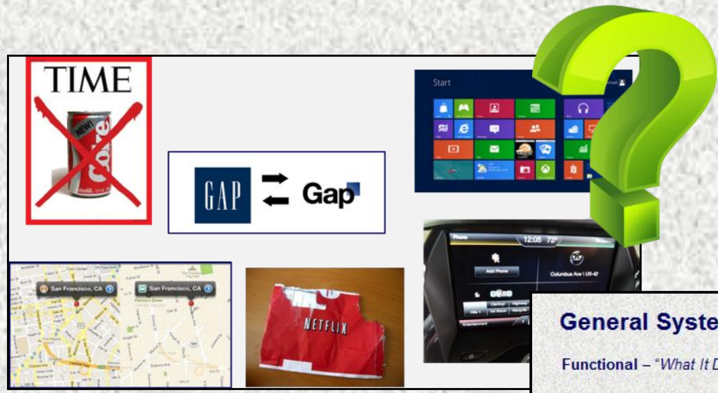
4) Solution Concept Selection



		Desired Impacts				Solution Boundaries				Negative Impacts
Solution Alternative		------	------	------	------	------	------	------	------	------
1					++	+		10	4	
2					-	--		2	5	
3					+	-		5	1	

Early System Development Steps

5) Solution Description and Feasibility Check



6) Solution Requirements

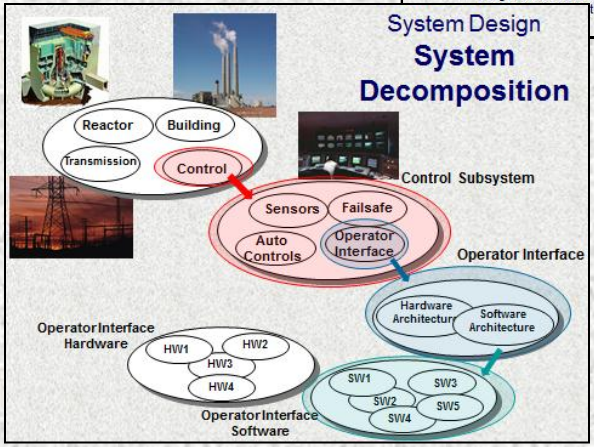

General System Requirements Categories 227

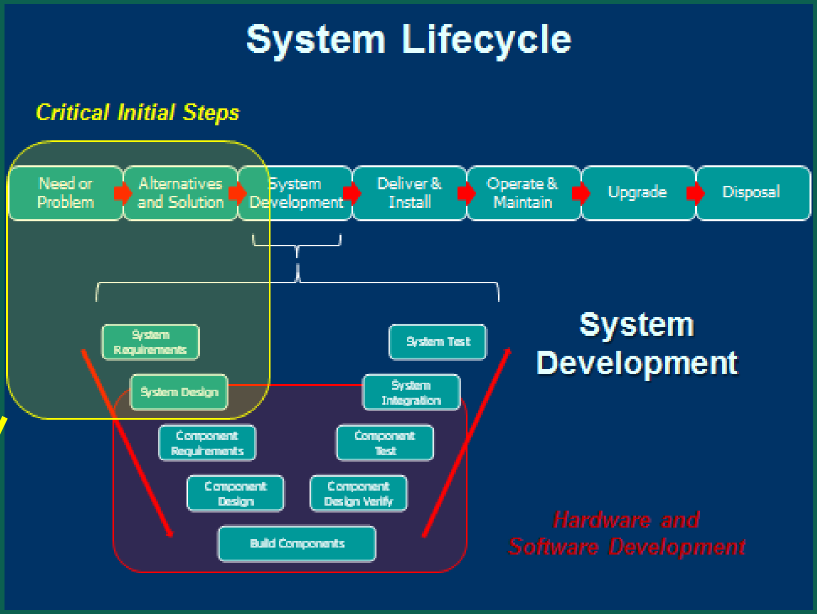
Functional – *"What It Does"*

Performance – *"How Well It Does It"*
- Speeds, Capacities, Throughput, Efficiencies, ……
- Up Time, Key Performance Measures, Reliability, Maintainability, …..

Constraints – *"Boundaries the System Must Fit Inside"*
- Impacts – Adjacent Systems, Adjacent Processes, Stakeholders, ……
- User – Skills, Interests, Capabilities, Demographics, …….
- Costs – Develop, Operate, Maintain, Dispose
- Safety – Operators, Users, Society, Maintenance Personnel, ……
- Physical – Size, Weight, Power Consumption
- Built-in Redundancy – Yes or No, Automatic or Not
- Maintenance – Test, Fault Isolation, Repair Times, Skills, Access, …….
- Upgradability – Yes, No, ……
- Compatibility – With Other Systems (current and prior), Interfaces, Processes, Information, ……..
- Conforming Rules – Standards, Regulations, Laws, ……
- Usage Environment – Temperature, Moisture, Sunlight, Wind, EMI, ……
- t Constraints – To Air, Ground, Water, ……..

7) System Design





1) Need or Problem

2) Solution Alternatives

3) Solution Boundaries

4) Solution Selection

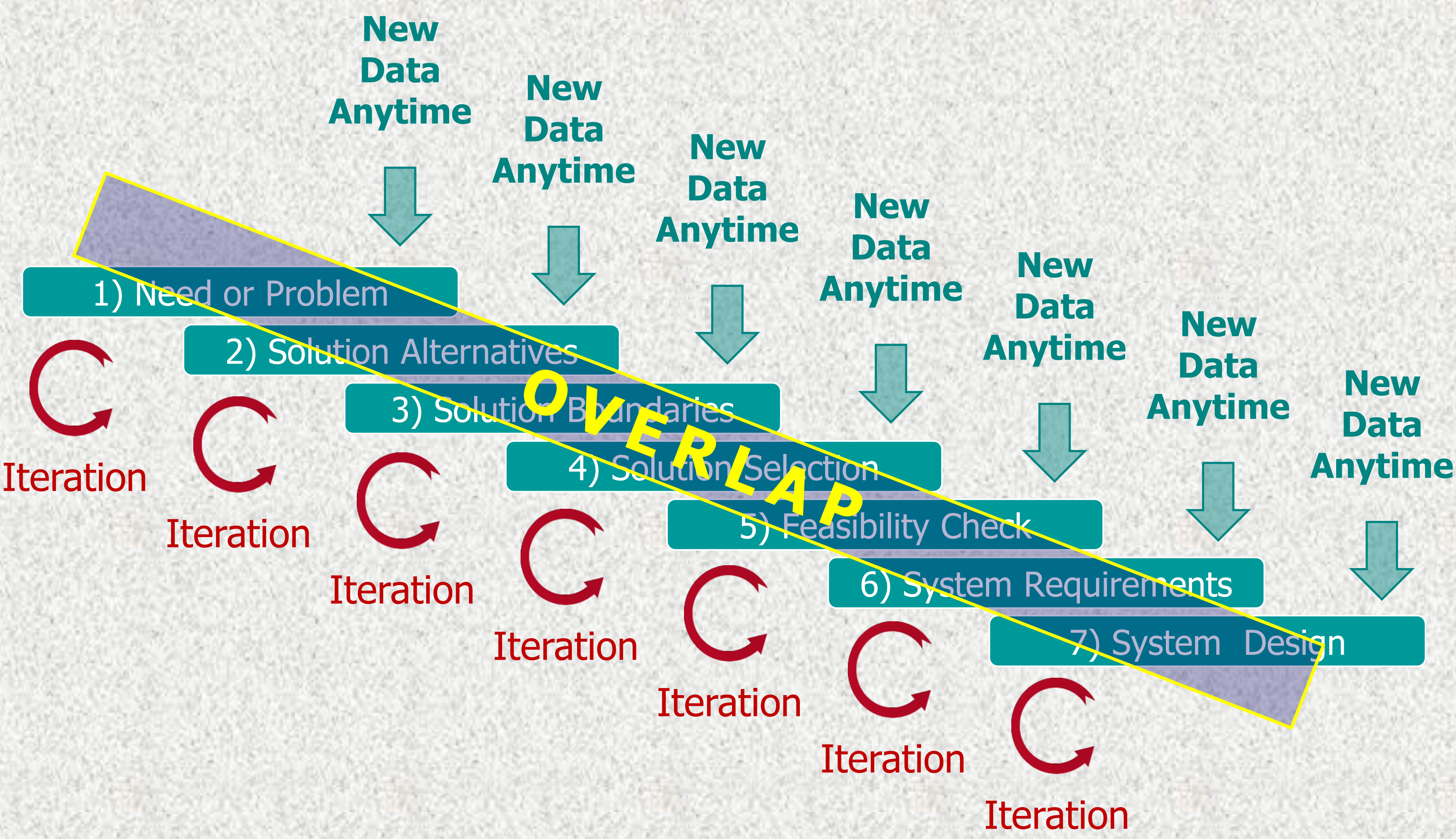
5) Feasibility Check

6) System Requirements

7) System Design

Critical Initial Steps

The Real World



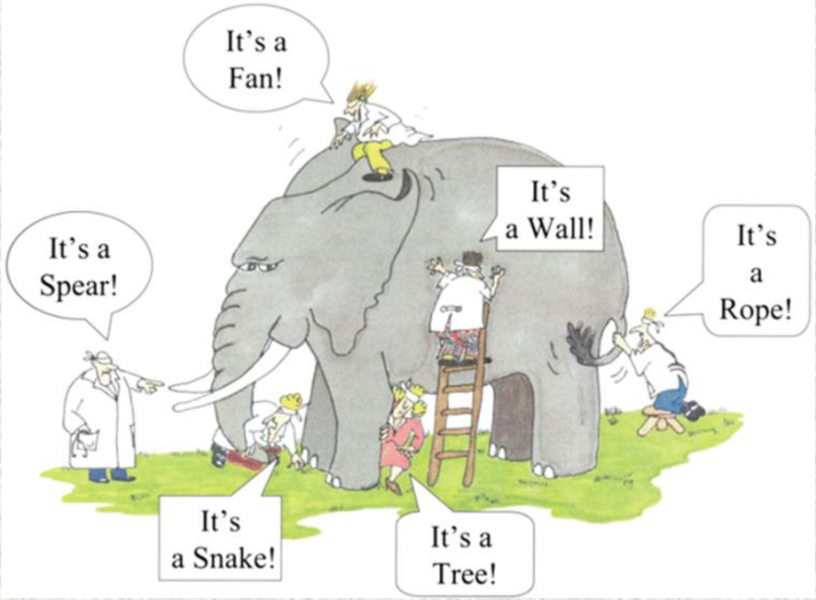
1) Need or Problem
2) Solution Alternatives
3) Solution Boundaries
4) Solution Selection
5) Feasibility Check
6) System Requirements
7) System Design

Stating the Problem or Need

What Exactly is Needed

State the Problem or Need

Not A "Systems View" of a problem or need



- What needs to change must be defined
- A clear understanding by all stakeholders of the WHOLE CONTEXT of the problem or improvement need is essential

NOTE: Get the problem wrong and it is hard to imagine how the solution will be right

Root Causes

What Really is the Cause of the Problem

Solutions Must Address Root Causes

Medical Diagnosis

Managing the symptoms of an illness does not cure the illness, diagnosis of the root cause is needed

Bank Examiner

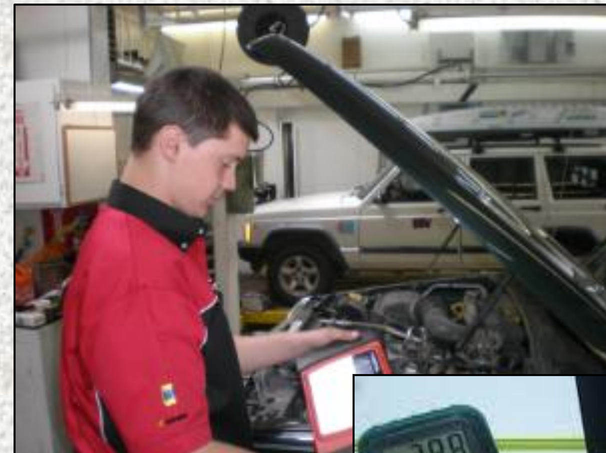
Bank financial problems are solved by analyzing for root causes

Accident Investigation

Making improvements in air safety uses root causes of accidents

Solutions Must Address <u>Root</u> <u>Causes</u>

Car Diagnosis



PC Problems

Business Problems

Root Causes Can Be Surfaced With Multiple "WHY" Questions

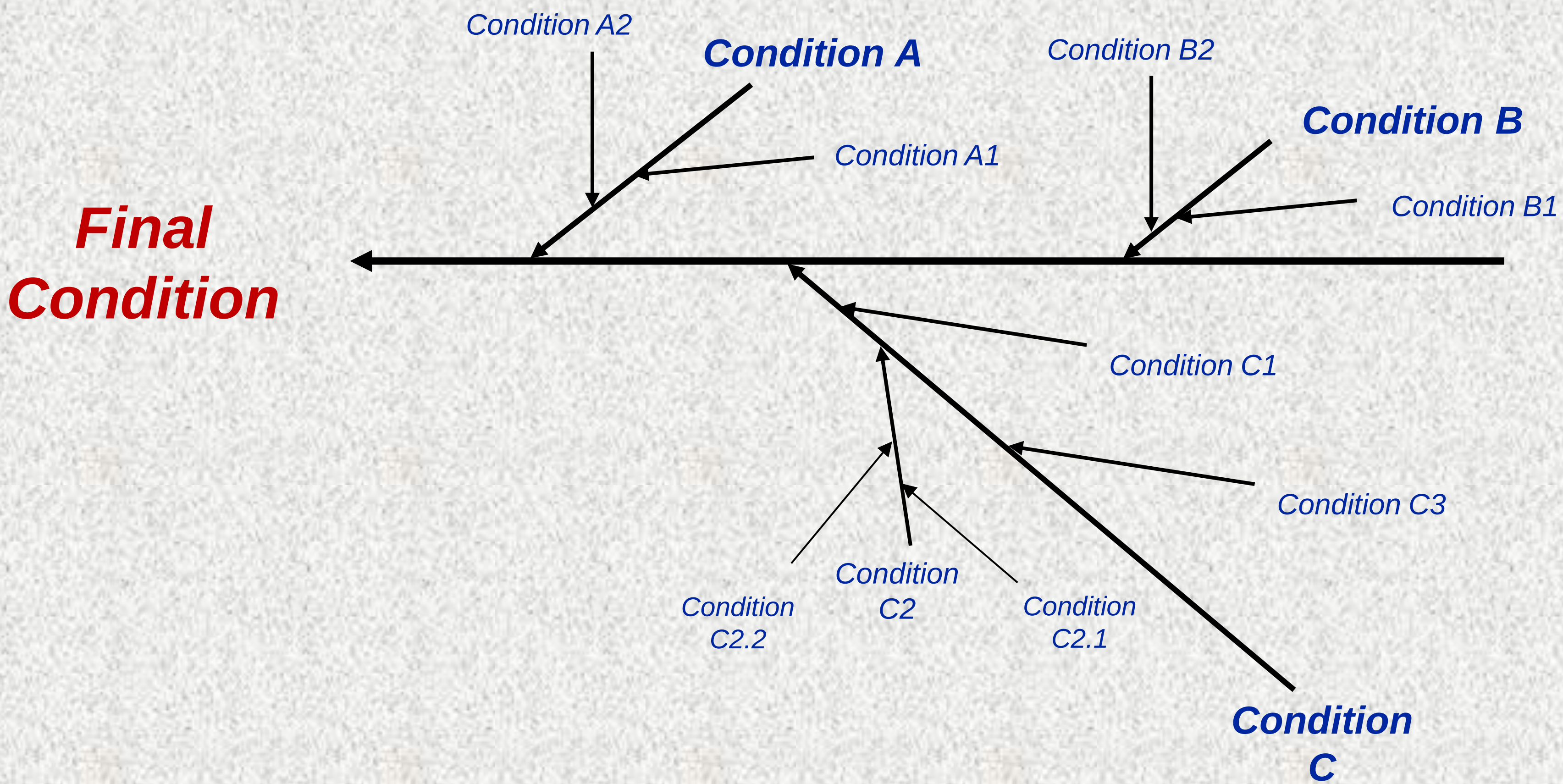
Take the example of a project with late schedule performance.

Example:

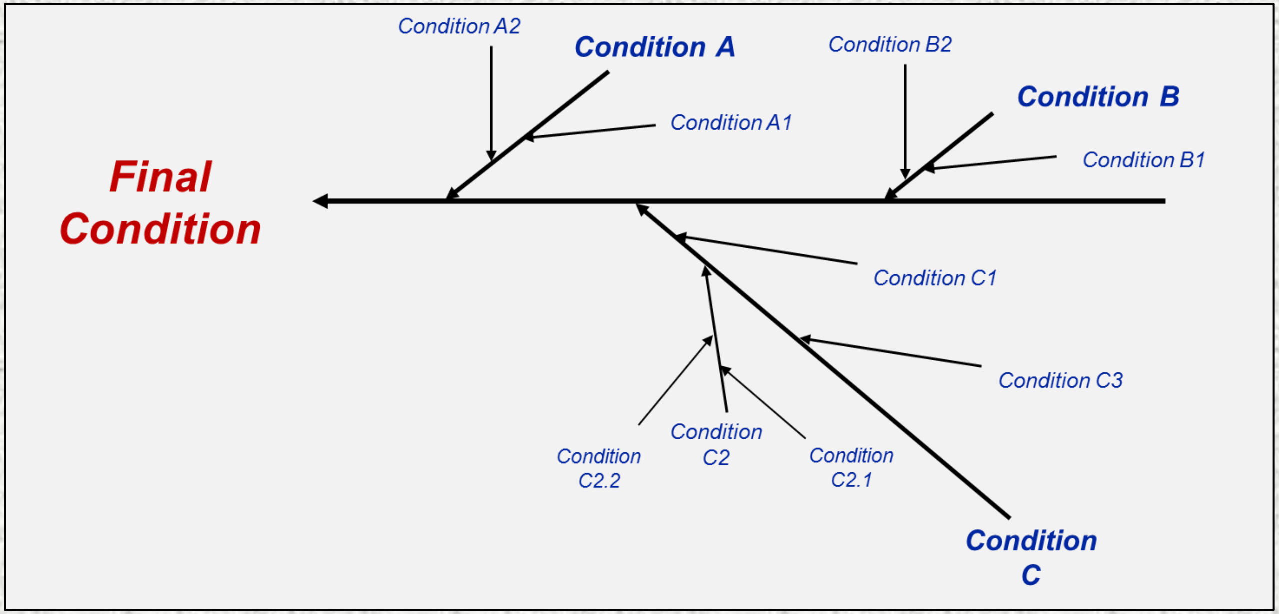
- **A project is worried about meeting schedule (Why?)**
- **Low productivity is a concern in one area (Why?)**
- **We are experiencing significant rework in that area (Why?)**
- **Unstable requirements are causing rework (Why?)**
- **The project is not managing the client**

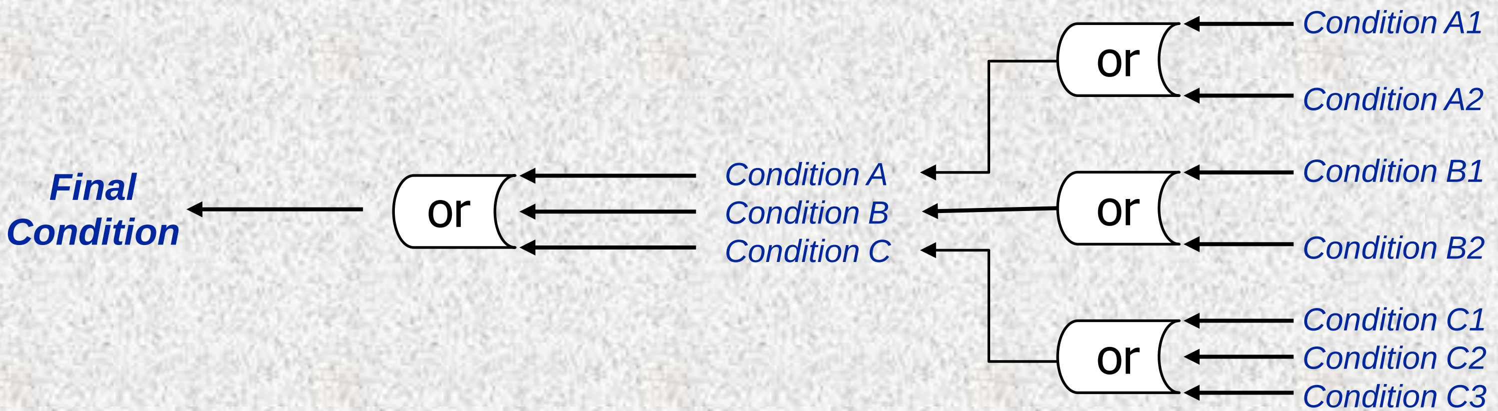
Here, client management and unstable requirements are problem, the other items listed are only symptoms of these root causes!

Root Cause Condition Relationships

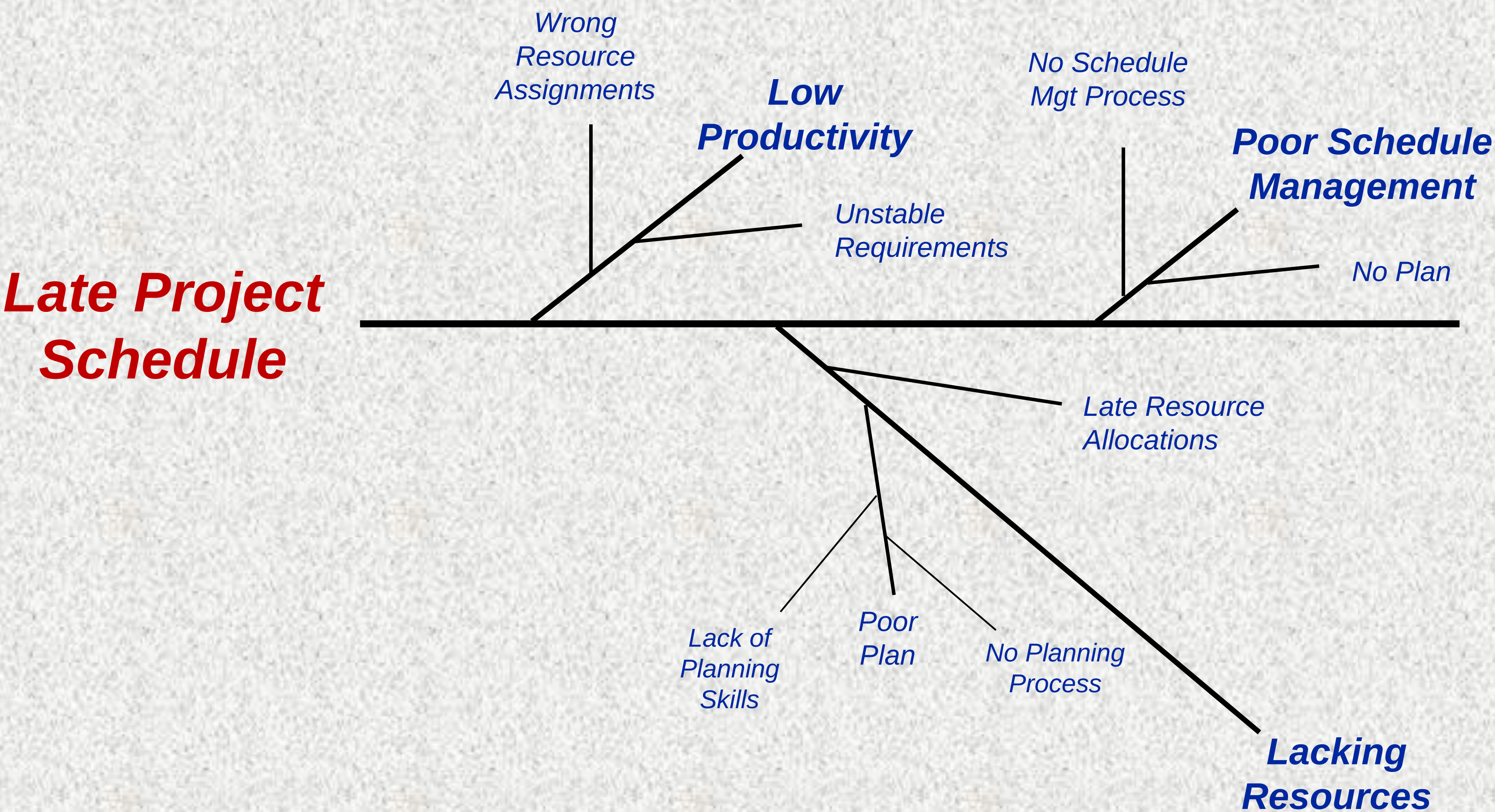


Root Cause Condition Relationships



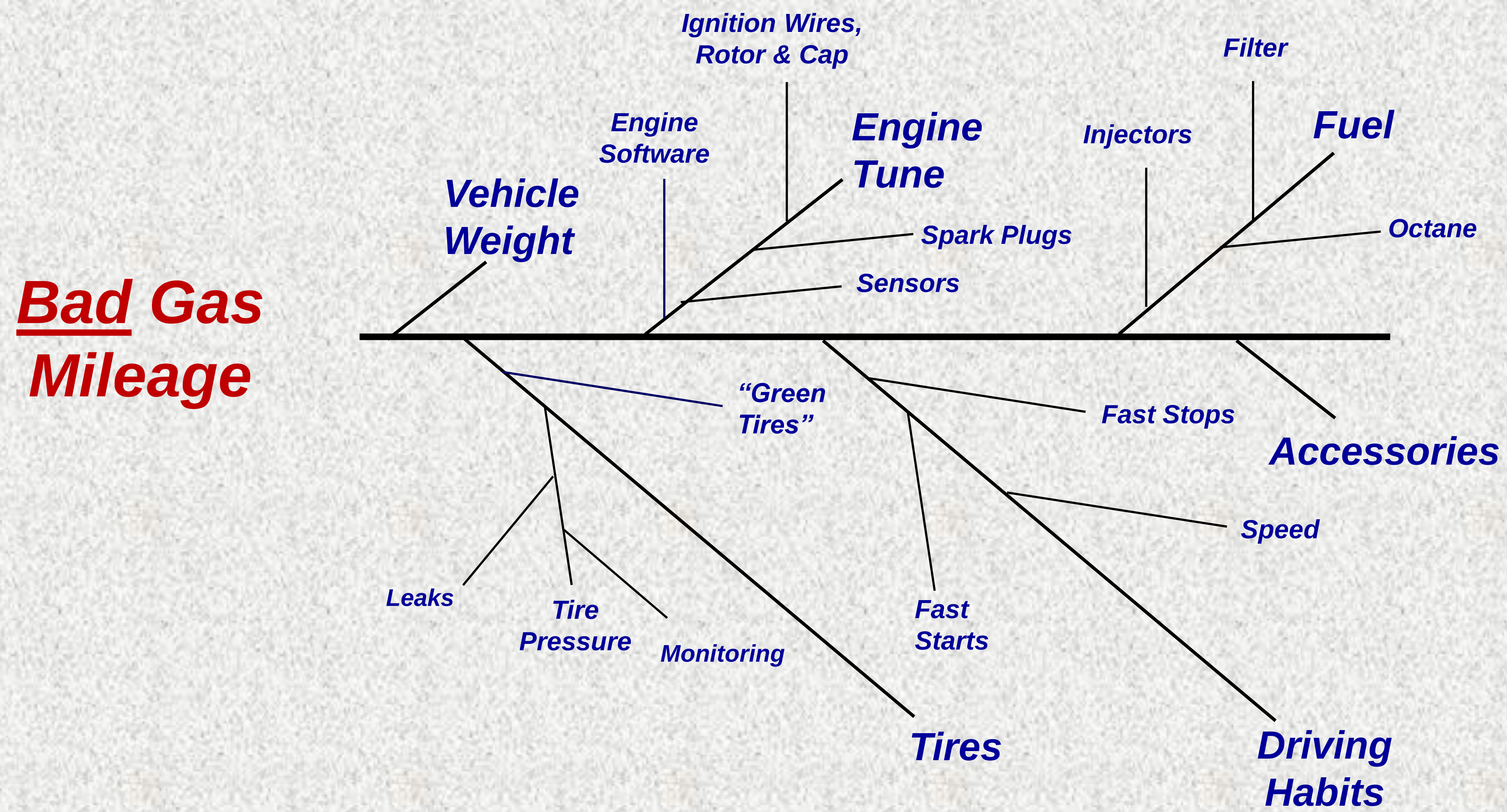


Project Schedule Problem Root Cause Analysis

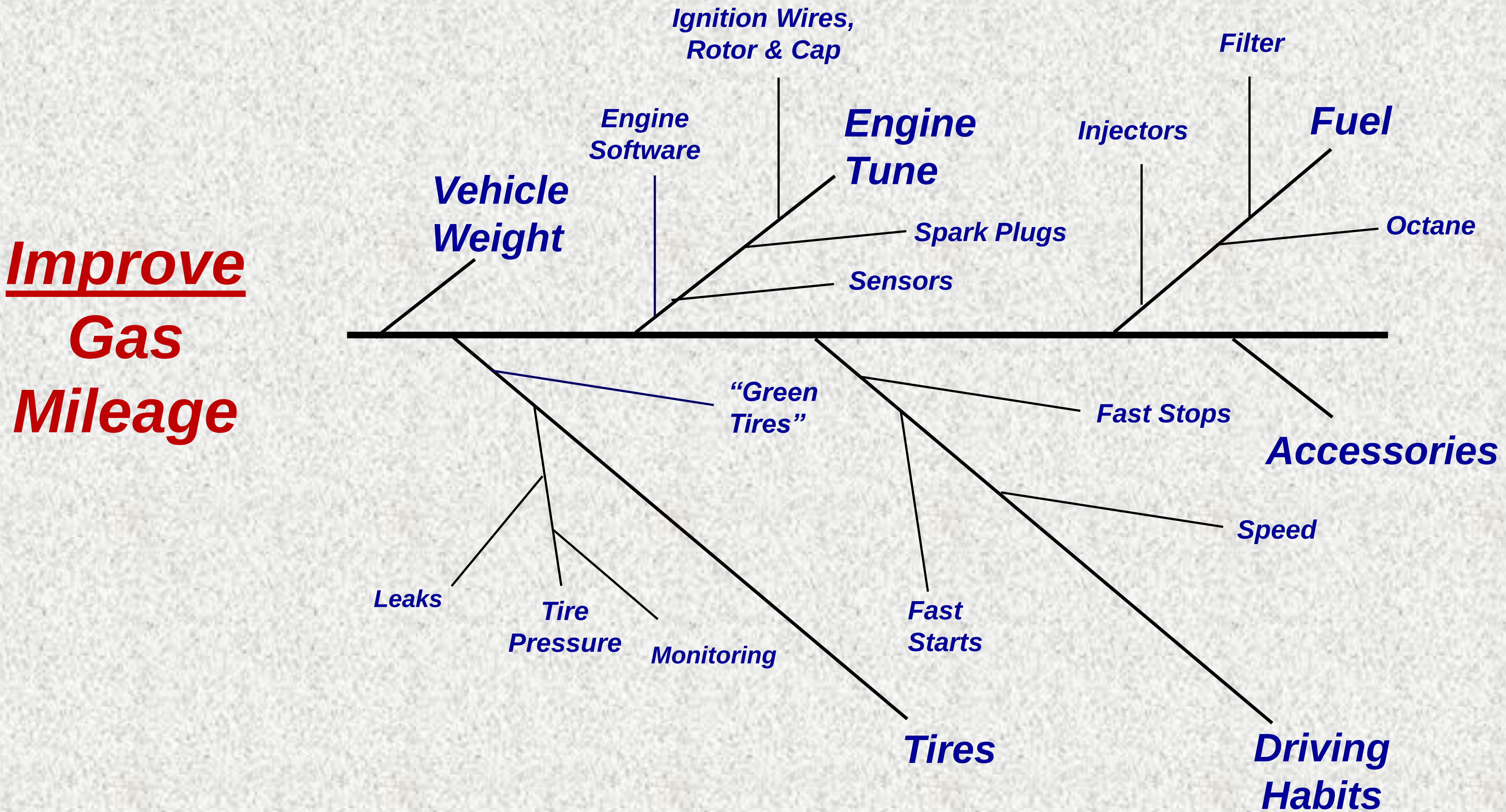


Bad Gas Mileage Root Cause Analysis

Identifies the "Drivers" of Mileage



Improvements Focus on "Root Drivers" <u>Same</u> <u>As</u> "Root Causes"



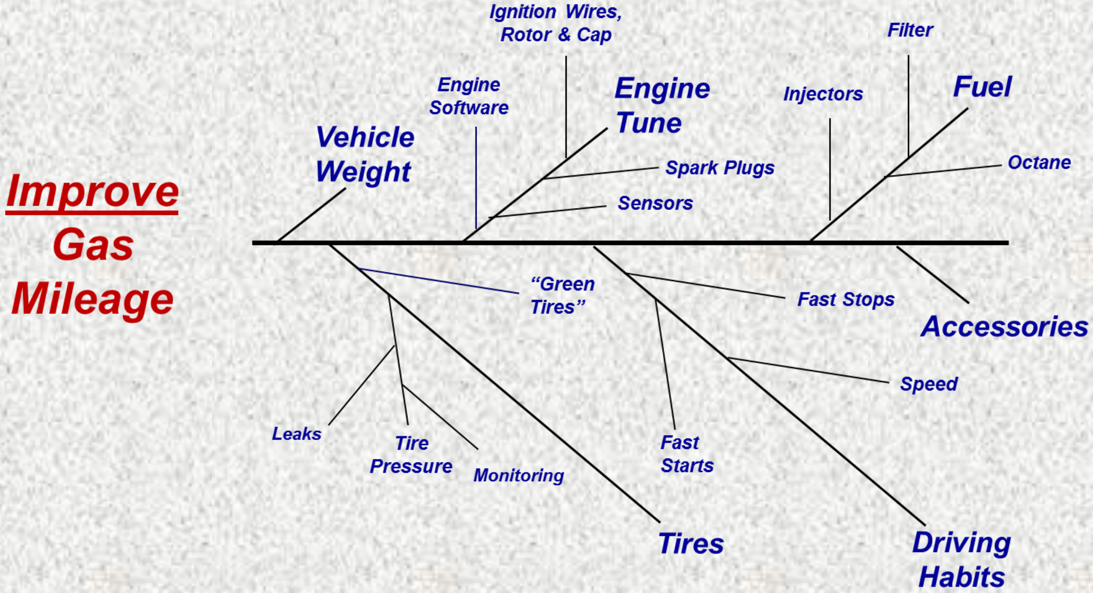
Sensitivity Analysis

"Vary Root Drivers, Measure Results, Define Sensitivities"

Needed Where Multiple Root Causes Exist

Sensitivity Analysis

1. Used to determine effects of variances in root drivers
2. Used to find primary drivers for problem resolution or improvements
3. Used to identify relationships of one driver to the others

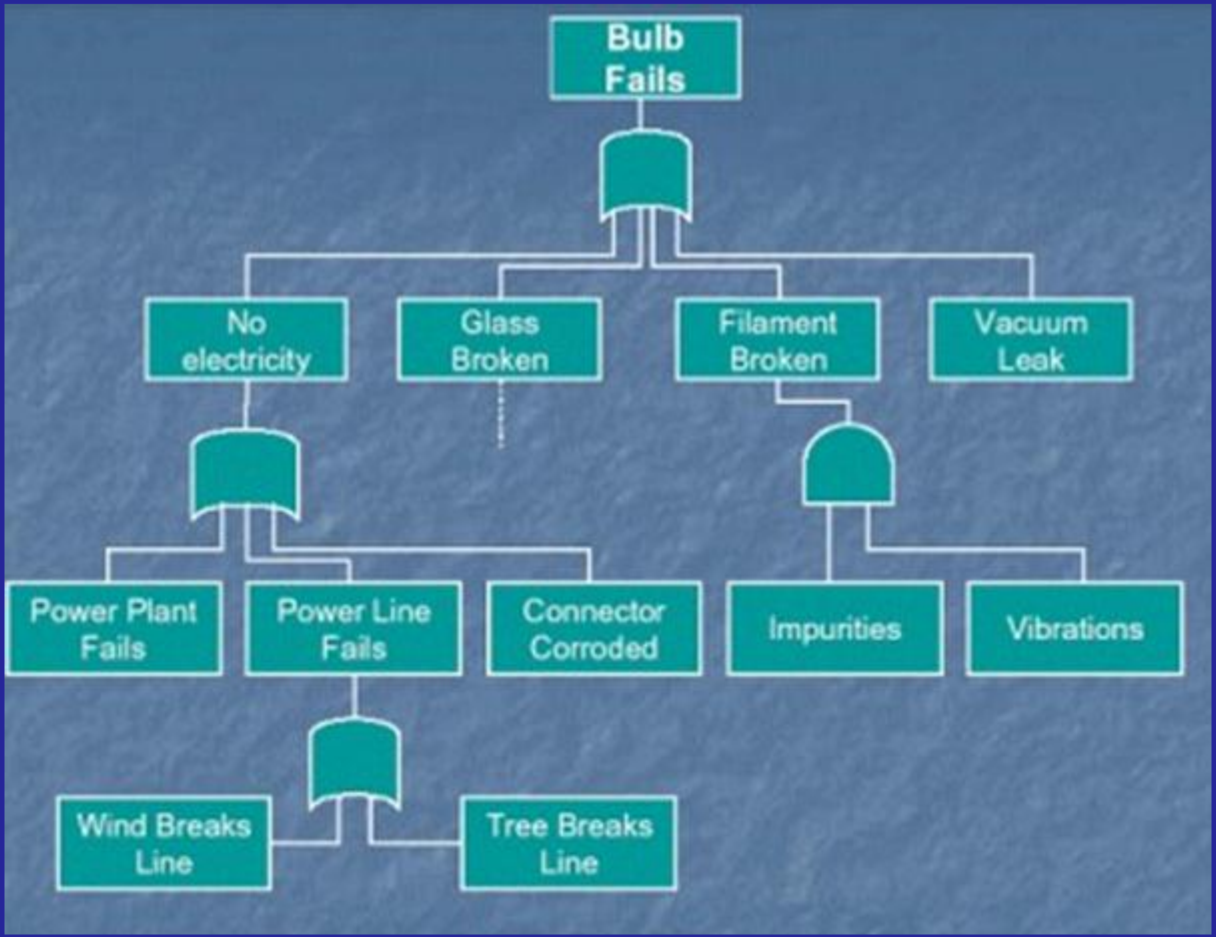


EXAMPLES

- *Vary Tire Pressure, Measure MPG*
- *Vary Driving Habits, Measure MPG*
- *Remove Accessories, Measure MPG*
- *Change Engine Tune, Measure MPG*
- *Combinations of Above, Measure MPG*

Fault Tree Analysis

Another Root Cause View



Systems Engineering Utilizes….

Operating Models

To Support Defining the Problem or Need

What Are Operating Models?

- **A description of how something works**
- **Can apply to** a system, a business, a factory, a government agency, ……
- **For Existing Systems, it is "a model of operations"** needed to understand the nature and context of the system so that a problem or improvement can be identified and understood
- **For New Systems, It is** a means to define conceptually **how the new system will function**

Operating Models

- The Environment and Context
- The Top Level Structure
- How Things Fit Together
- Sequences of Work
- Flow of Information and/or Products
- Roles and Responsibilities

Operating Model

Sequential

Start Condition

TIME

Function 1 *Work* → *Work*

Function 2 *Work* → *Work* → *Work*

Function 3 *Work* → *Work* → *Work*

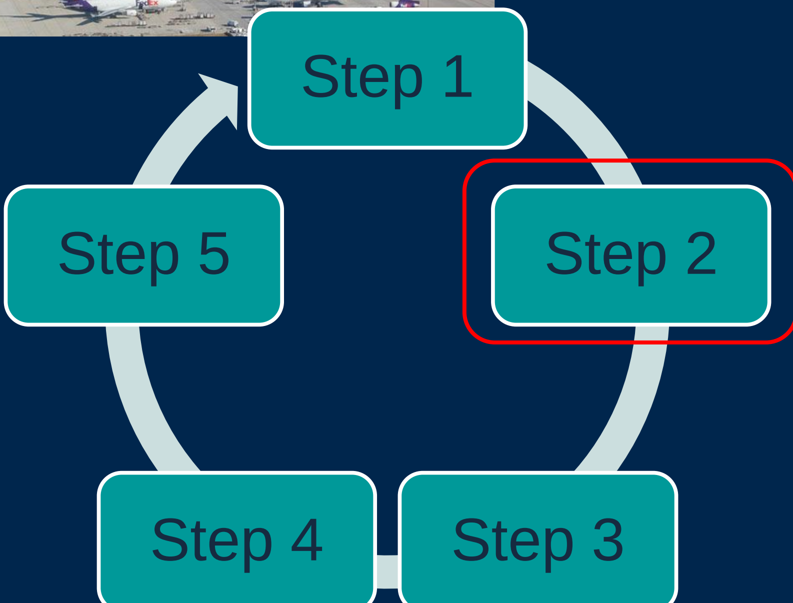
Function 4 *Work* → *Work* → *Work*

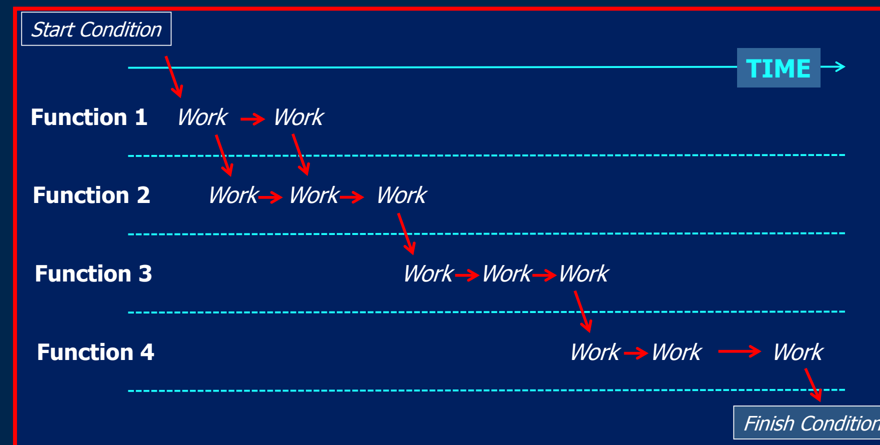
Finish Condition

Operating Model

Iterative

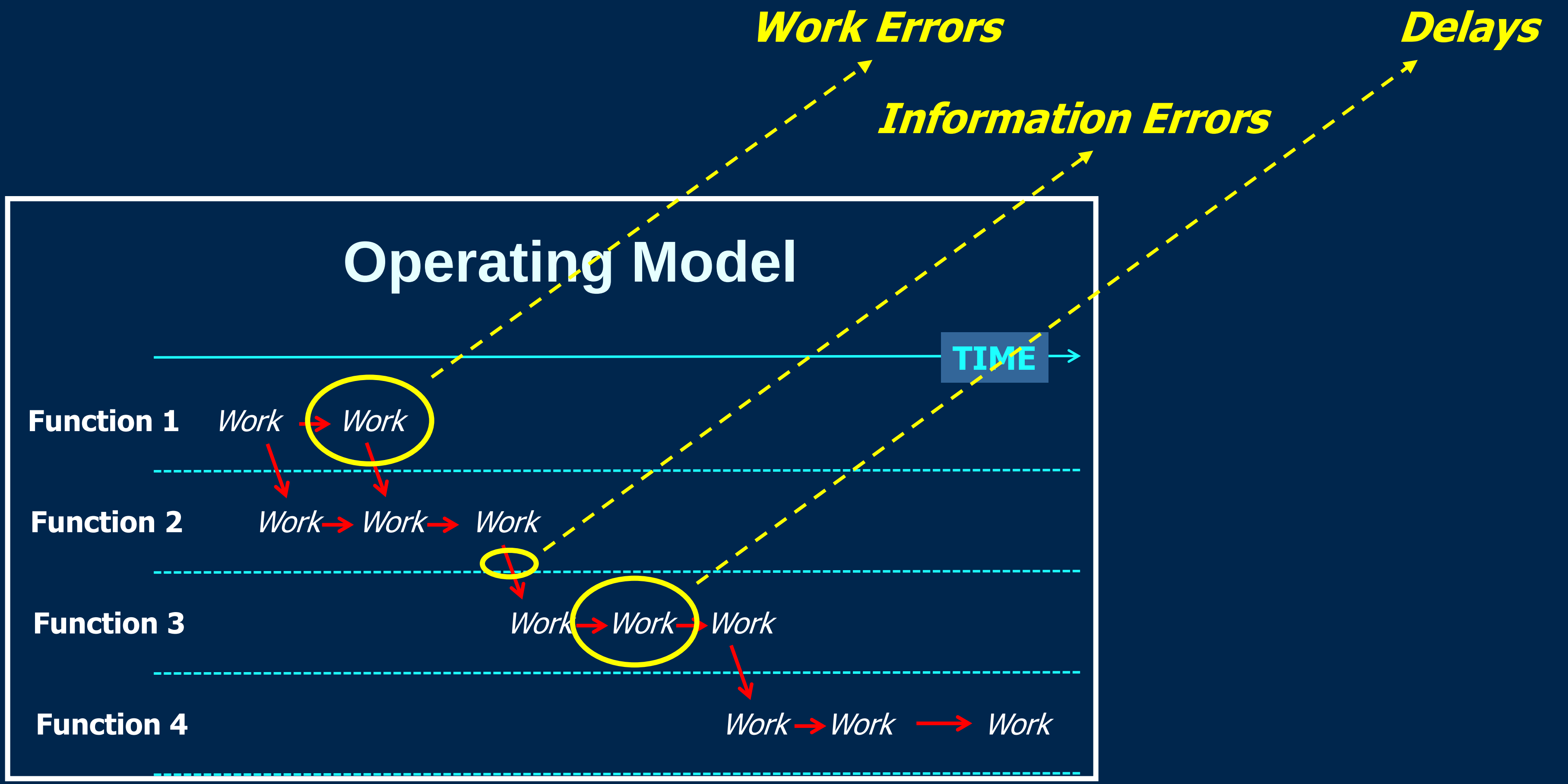
Operating Model
Combinations





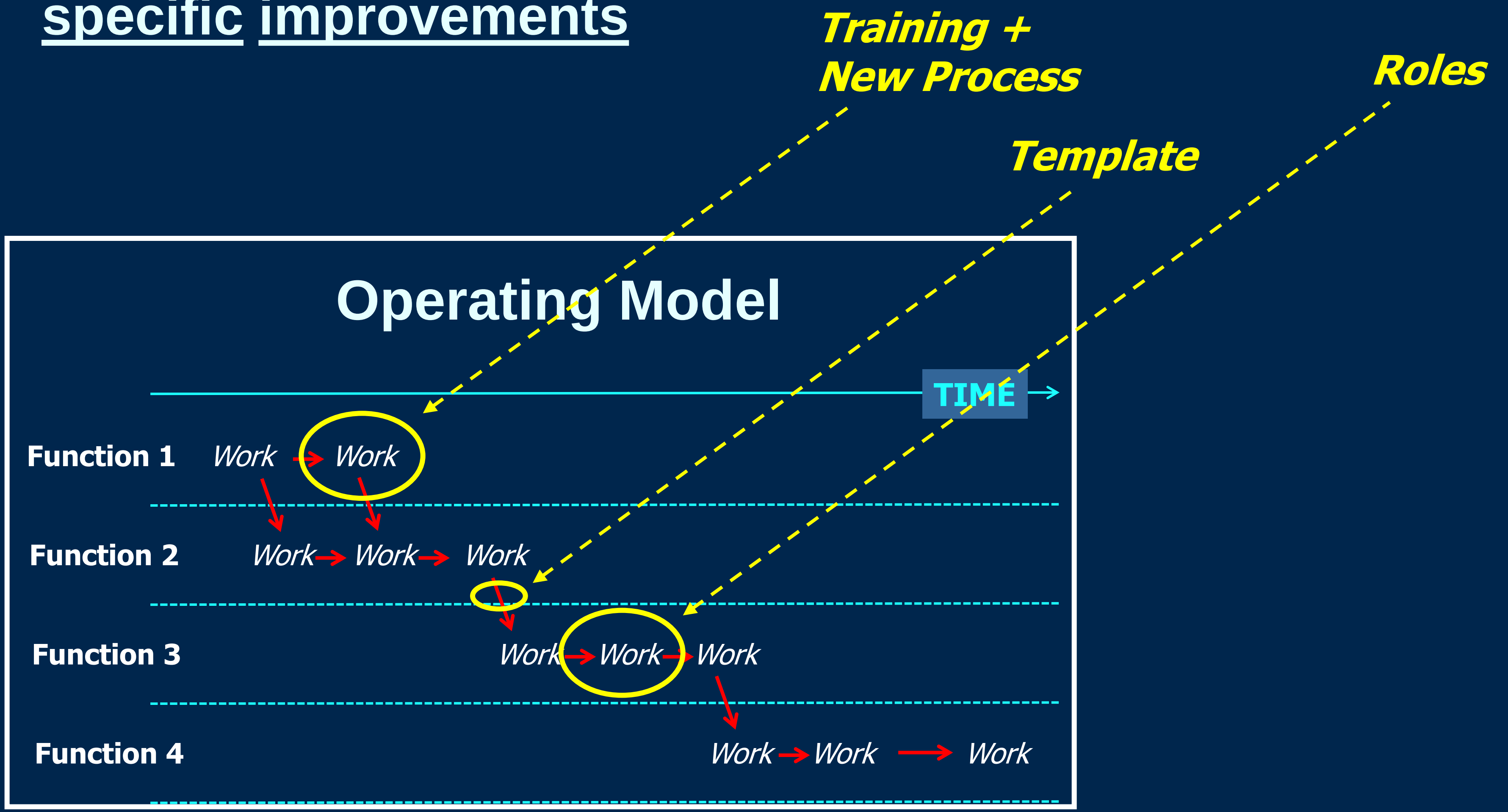
What To Improve?

An operating model supports identification of <u>problems</u>



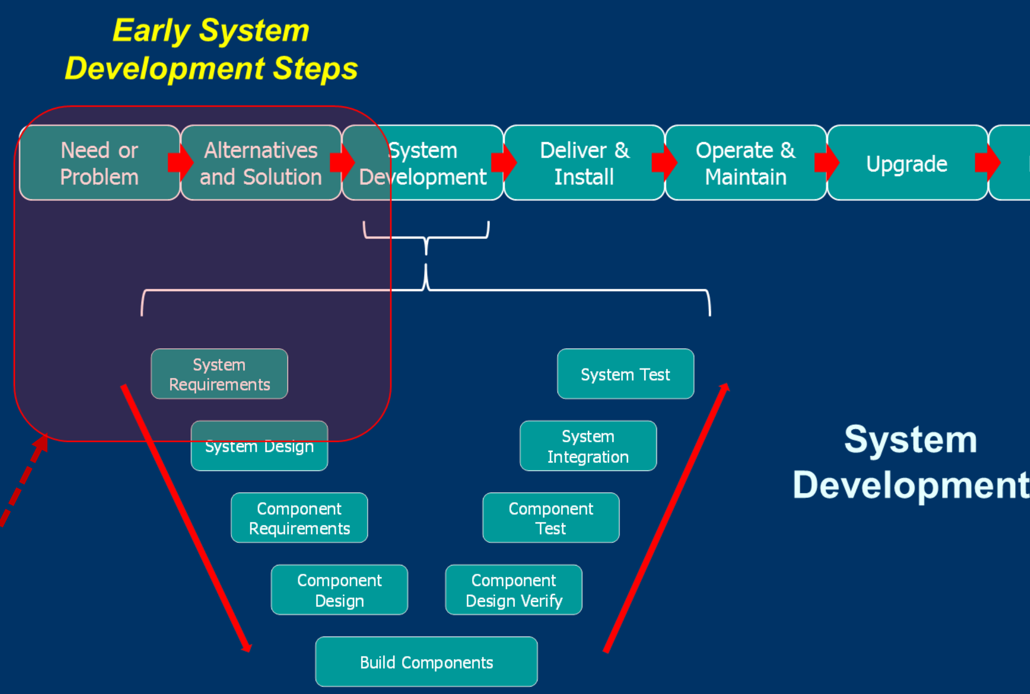
What is Done?

An operating model supports identification of __specific__ __improvements__



Fit of Operating Models

Operating Models Support Front End System Development Steps



1) Need or Problem

2) Solution Alternatives

3) Solution Boundaries

4) Solution Selection

5) Feasibility Check

6) System Requirements

7) System Design

Operating Model Requirements

General Requirements for An Operating Model

- Sufficiently simple
  - To be understood and accepted
  - To be widely applicable

- Sufficiently substantive
  - To include all repeatable work
  - To communicate work flows, roles
  - To communicate the context of work involved

- Understood and accepted by all relevant stakeholders

Operating Model Application On Different Project Types

PROJECT TYPES | **APPLICATION**

Repair, Refactor or Remodel Project, Model Exists

Operating model is the current "model of operations," *no new model*

Ground Up Development, Model Exists

New operating model is a revision of a past model

Ground Up Development, No Model Exists

New operating model is a new "Concept of Operations"

What Can Go Wrong?
What Do You Do?

POTENTIAL PROJECT FAILURE

- If you have no “model of operations” or an inadequate model for what you are trying to improve, then your understanding of what is required and how to do it will be diminished

WHAT TO DO

- Understand the need for operating models, educate others
- Develop an appropriate model for your area of work
- Involve appropriate stakeholders to complete and validate the model
- Update the model as needed

Operating Models

What To Do

Reference

You Are A Client

- Have an operating model
- Understand your model
- Make sure your contractor understands your model

You Are Management

- Understand the importance of models, educate project teams
- Understand the model of client operations
- Ensure the development team understands the client's model

You Are An SE, Project Manager or Technical Lead

- Understand the need for models
- Educate your team on the need
- Understand your client's model
- Ensure the development team understands the client's model
- Use the model to drive project plans and details

You Are A Team Member

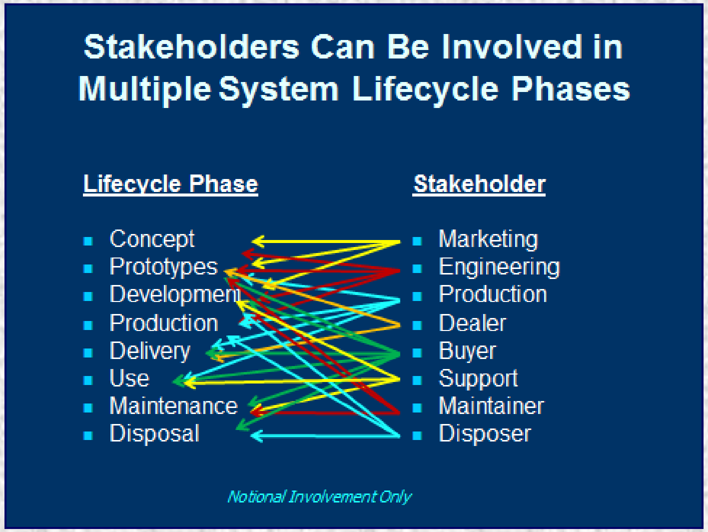
- Understand the model of client operations to the extent needed
- Constructively raise concerns where lack of an understanding of the client's operating model impacts the development

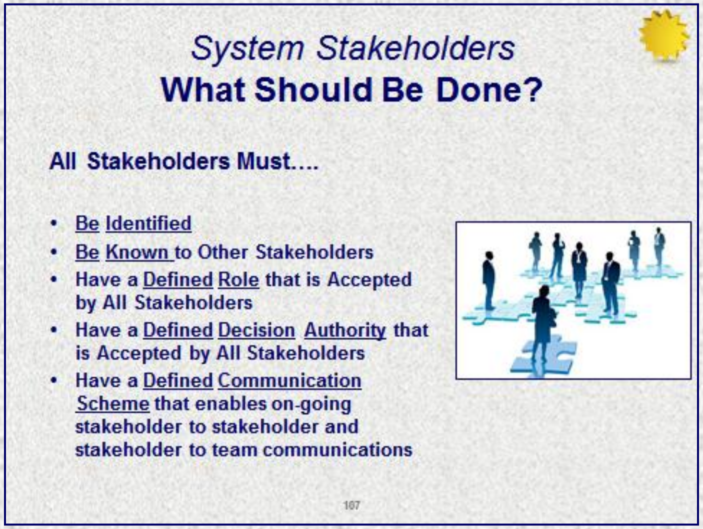
Stakeholder Needs

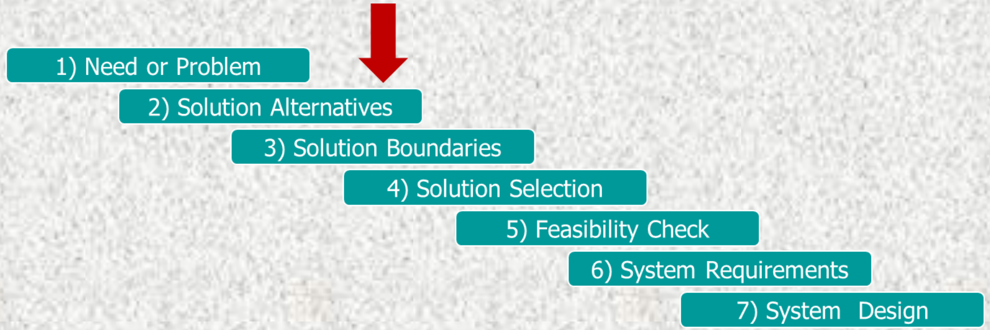
Needs of all Stakeholders are Key Requirements

Stakeholder Needs Are A Part of Solution Requirements

- **Stakeholder needs** are **critical requirements** for any solution
- A clear understanding of stakeholder needs is **key** to making a good **solution selection**
- **Consider all stakeholders** even though some may appear to be more important than others





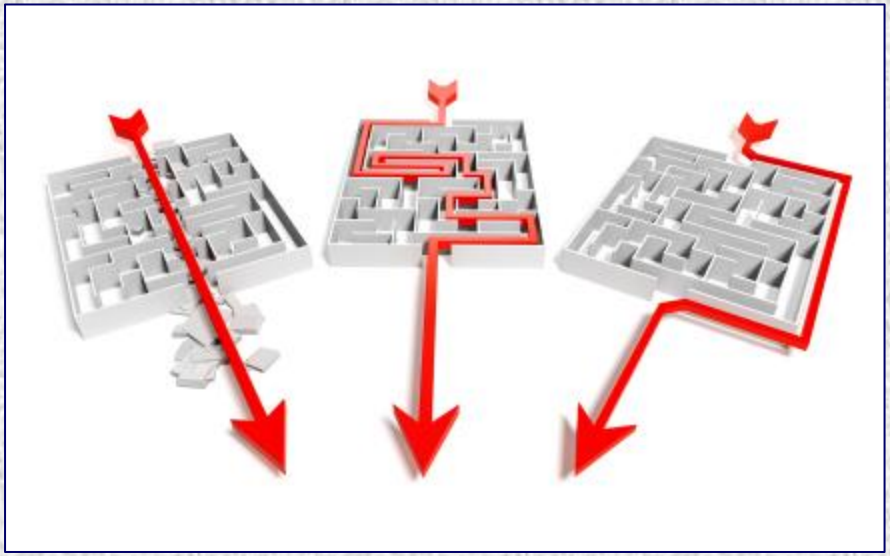


Initial Alternative Solutions

Alternatives Need to be Considered

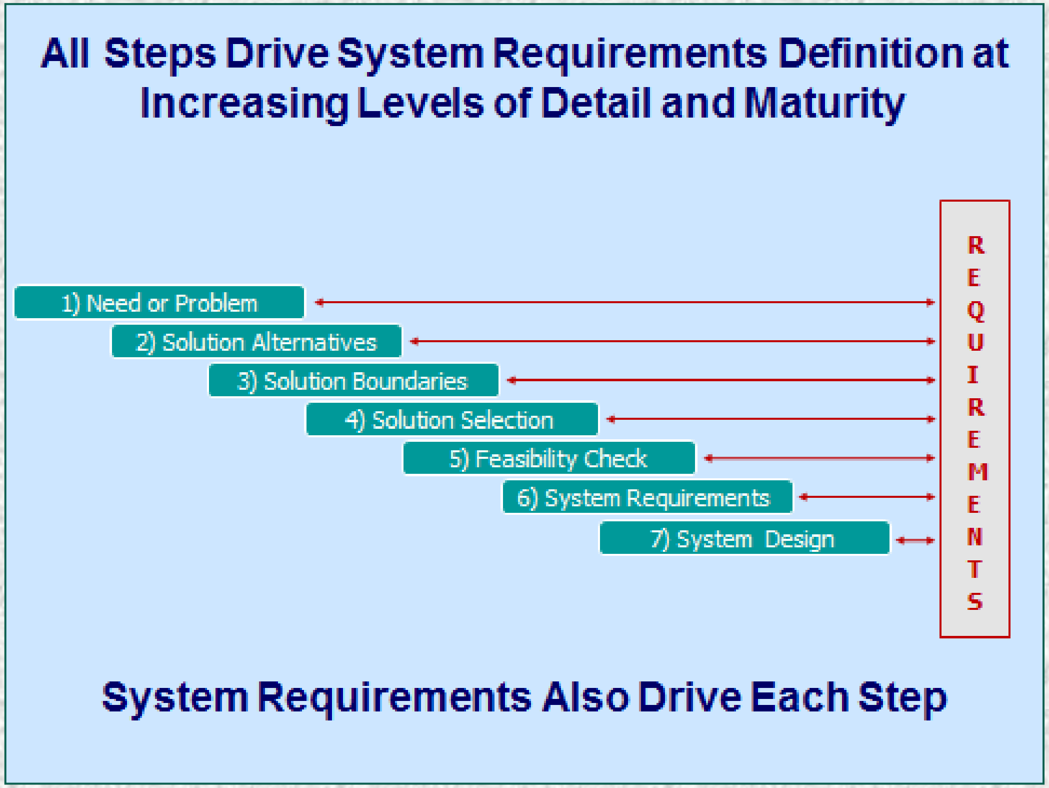
Why Alternative Solutions?

- Is the first idea always the best idea?
- Once an idea is identified, does it change later?
- Has a past project started with the wrong solution?



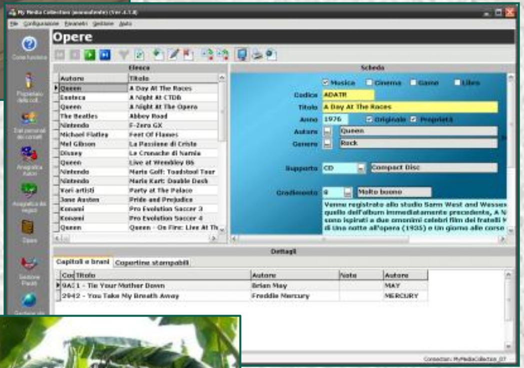
Shouldn't We Define Requirements First?

- Actually we are starting to define requirements
- Requirements are usually built up successively over many steps



Shouldn’t We Define Requirements First?

- At this point we do not know if the solution will be a “kitchen sink”, a new “business app” or a “field of banana plants” or some combination of these



- If too much detail is initially defined for requirements, viable alternatives may be inappropriately filtered out

- The focus should be on “The Solution” to “The Problem”

How to Identify Solution Alternatives

- **Review your problem and brainstorm with the team**
- **Brainstorm with other stakeholders**
  - Management, Customers
  - Peer Projects
  - Experts, Internal and External
  - …(others)…..

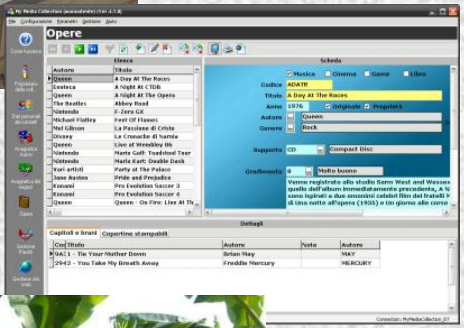
- **Look for adjacent models**
  - Similar past solutions in your organization
  - Similar solutions in other organizations
- **Look for non-adjacent models**
  - Construction project aspects related your problem
  - Government or a different industry project related to your problem

Consider All Solution Alternatives

- **Any alternatives that "appear" to be even remotely feasible should be considered**

- **Do not invalidate any alternatives**
  - **Bad alternatives will be removed later**
  - **Some alternatives may not be bad**
  - **Keeping all ideas "on the table" keeps the team engaged and thinking**

- **Take your time, you may well find a great solution that was not obvious**



Identifying Alternative Solutions

Out of the Box
& In the Box

Thinking

Thinking

Thinking

Thinking

Thinking

Review Past Solutions

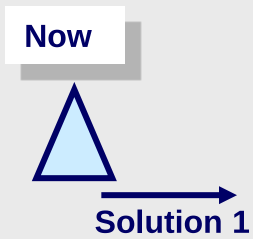
Solutions Now

Future Solutions

Consider Time Dimensions

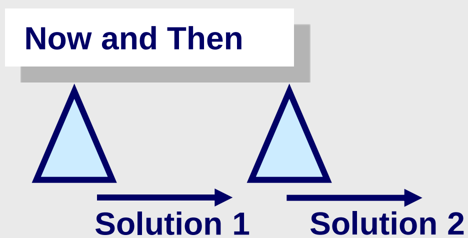
Consider Idea Extensions

Time Phased Solutions



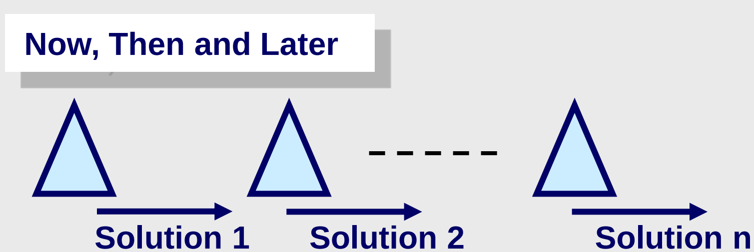
Near Term Singular Solution

1. Reduce train speeds, more track maintenance



Medium Term Phased Solutions

1. Reduce train speeds, more track maintenance
2. Lower gas content of crude, Prepare emergency teams



Long Term Phased Solutions

1. Reduce train speeds, more track maintenance
2. Lower gas content of crude, Prepare emergency teams
3. Develop new energy policies and regulations

1) Need or Problem
2) Solution Alternatives
3) Solution Boundaries
4) Solution Selection
5) Feasibility Check
6) System Requirements
7) System Design

Solution Boundaries

What are Necessary Constraints on the Solution?

Solution Boundaries EXAMPLES

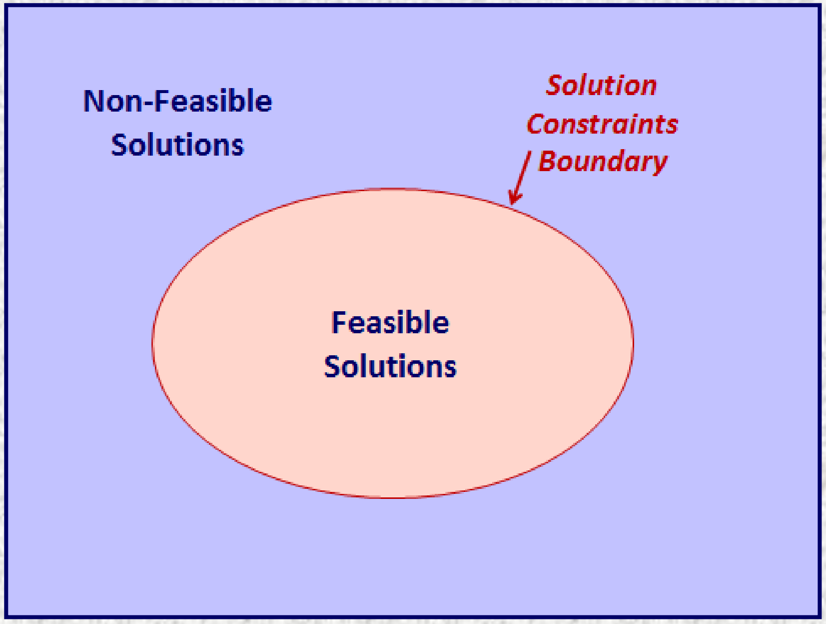
<u>Example</u> <u>Solution</u> <u>Boundaries</u>

- **User Related Boundaries** – Skills, language, interests, ……..
- **Cost Limitations** – Development, Production, Implementation, Use
- **Schedule Limitations** – Development, Production, Implementation
- **Resource Limitations** – People, Facilities, Systems, …….
- **Implementation Related Boundaries** – Security, Access, Time, …
- **Technology Boundaries**
- **Political or Societal Boundaries**
- **Regulatory and Legal Boundaries**
- **“Own Organization” Cultural Boundaries**
- **…..(more exist)………**

Define
Solution Boundaries

Solution Boundaries…….

- Are a set of **constraints** the solution must fit within
- Are NOT all detail system requirements, but are requirements
- Are needed to identify feasible solutions and reject unfeasible solutions



- *Cost*
- *Performance*
- *Acceptability*
- *Technology*
- …..
- …..

1) Need or Problem
2) Solution Alternatives
3) Solution Boundaries
4) Solution Selection
5) Feasibility Check
6) System Requirements
7) System Design

Alternative Solutions Trade-Off Criteria

Solution Selection Methods

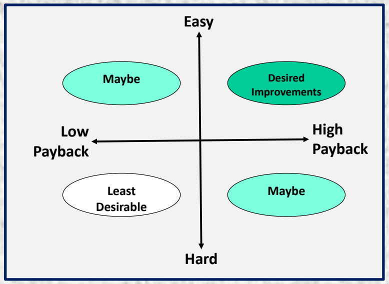
Selection Process

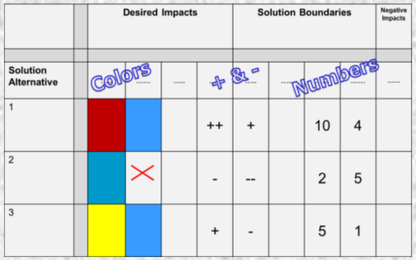
a) Define Solution Selection Criteria

b) Prioritize Criteria

c) Collect Criteria Information

d) Compare Information, Make Selection





Solution Selection Criteria

- **Needed to pick the right solution alternative**

- **Criteria will include things we have addressed…….**

  – **“Solution Boundaries”**
    – **Cost, Schedule, Technology, Environmental, Political, ……**

  – **“Stakeholder Needs”** - Measures of Attaining the “Desired Outcome”, i.e. “Solving the Problem”
    – **Quality, Value Imparted, Customer Needs, ….**
    – **Esthetics, Long Term Value, Operational Considerations, ……**

  – *…(other known selection criteria can exist)……*

Example
Generic Solution Selection Criteria

Improvement <u>Value</u>
- To Customers
- To Business
- To Staff
- To “Local” Operations
- To Adjacent Operations
- To Suppliers

Implementation <u>Needs</u>
- Approximate Cost
- Approximate Schedule
- Approximate Resource Needs

Anticipated <u>Improvements</u>
- Increased Quality
- Decreased Costs
- Increased Safety
- Decreased Schedule
- ……(other values)…….

Improvement Impacts <u>Short Term</u>
- To Customers
- To Business
- To Staff
- To “Local” Operations
- To Adjacent Operations
- To Suppliers

Improvement Impacts <u>Long Term</u>
- To Customers
- To Business
- To Staff
- To “Local” Operations
- To Adjacent Operations
- To Suppliers

Alternative Solutions Comparison & Selection

Common Causes of Bad Solution Selections

- Unclear knowledge of problem or need
- Not having a sufficient number of alternatives
- Not considering all the alternatives
- Lack of time to analyze alternatives
- Lack of comparison information
- Lack of a basic comparison and decision process
- Inaccurate impact assessment of alternatives
- Lack of constructive debate
- Impulsive decisions
- Strong personalities overpower weaker personalities

Don’t Do These, Use a Selection Process

Initial Filtering of Alternatives

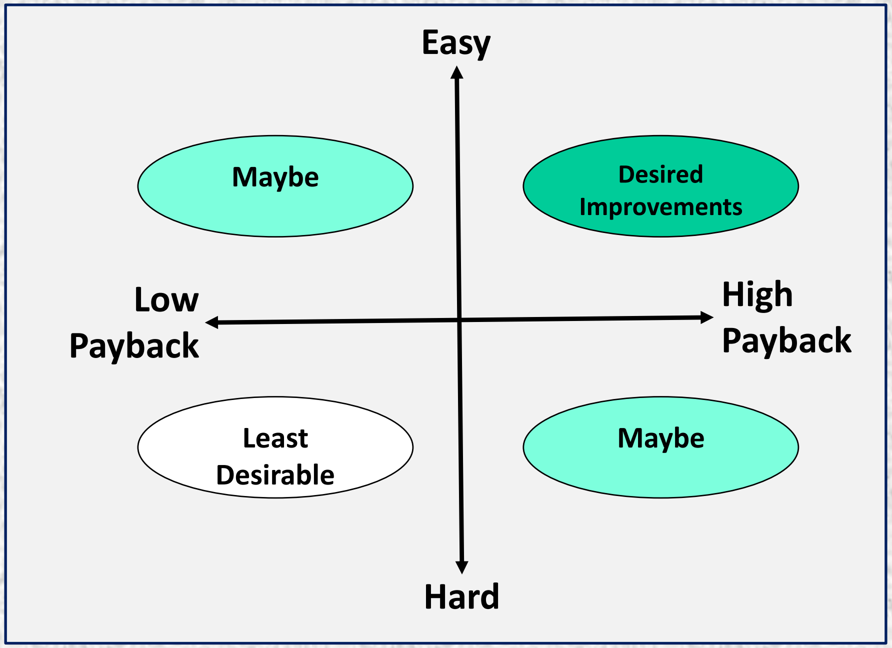
Where Many Alternatives Exist

High vs. Low <u>Payback</u> can be characterized relative to:

- Increasing Customer Value
- Aligned with Success Factors
- Financial or Schedule Paybacks
- Quality Paybacks
- Secondary Effects

Easy vs. Hard <u>Effort</u> to Accomplish can be characterized relative to:

- Resource Requirements
- Costs to Implement
- Schedule to Implement
- Liability or Safety Risks
- Unintended Consequences



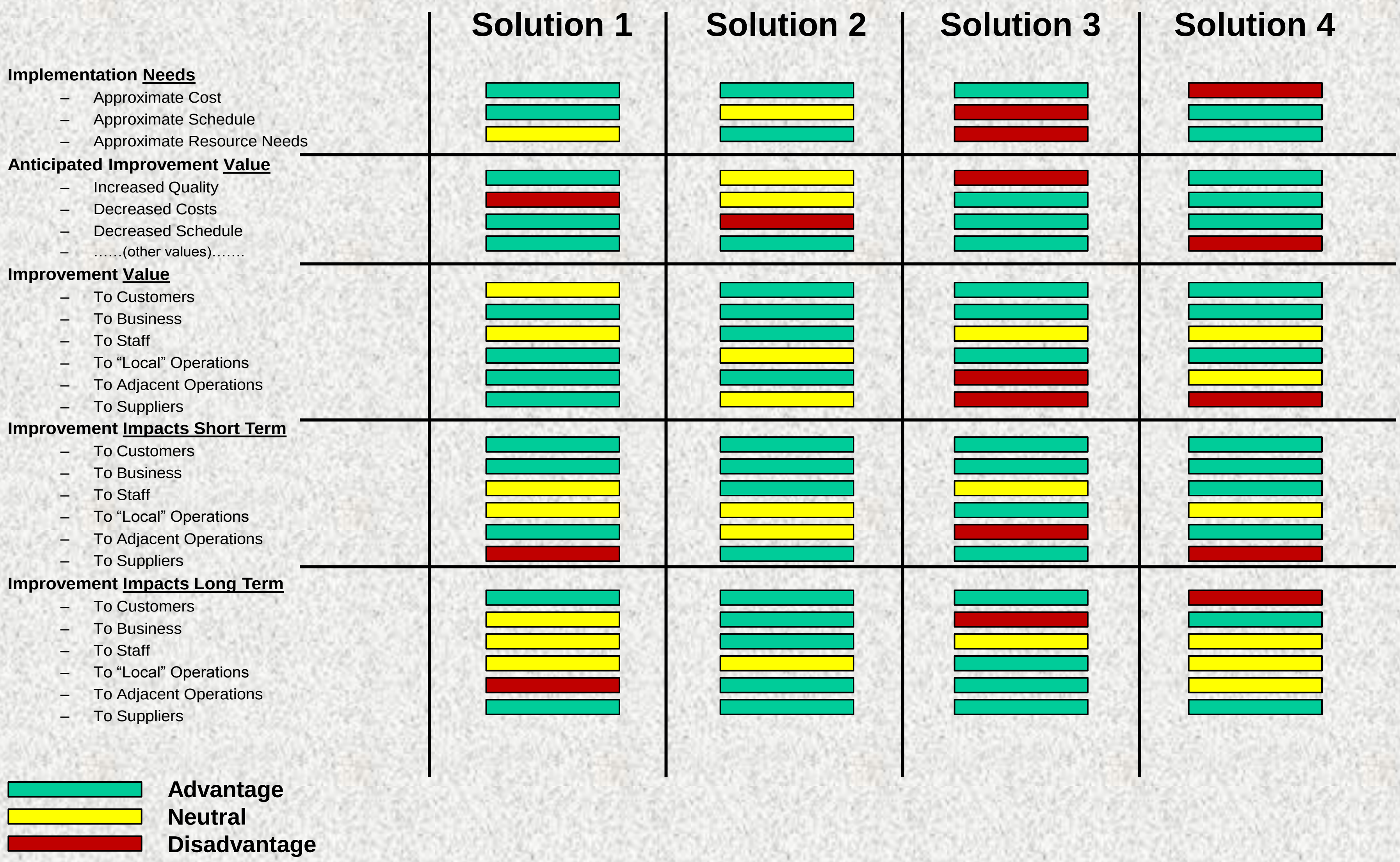
Basic Comparison Matrix

Solution	***Cost***	***Time***	***Effort***	***Effectiveness***	***Stakeholder Impacts***		
1							
2							
3							
4							

COMPARISON DISCUSSION METHODS

- ***Color Cells*** Blue = Great, Yellow = OK, Red = Not Good
- ***Showstopper Category***

Example Color Code Selection Chart



In-Depth Comparison Matrix

		Desired Impacts				Solution Boundaries				Negative Impacts
Solution Alternative		...... Colors	......	......	...... + & -	......	......	...... Numbers	......	......
1					++	+		10	4	
2					-	--		2	5	
3					+	-		5	1	

<u>Comparison</u> <u>Discussion</u> <u>Methods</u>

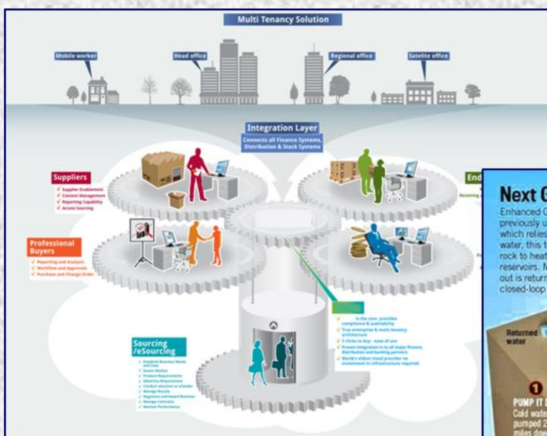
- ***Color Cells*** Blue = Great / Yellow = OK / Red = Not Good
- ***Showstopper Category***
- ***Scored Weighting*** 1 to 10 or H / M / L or ++ / + / 0 / - / --

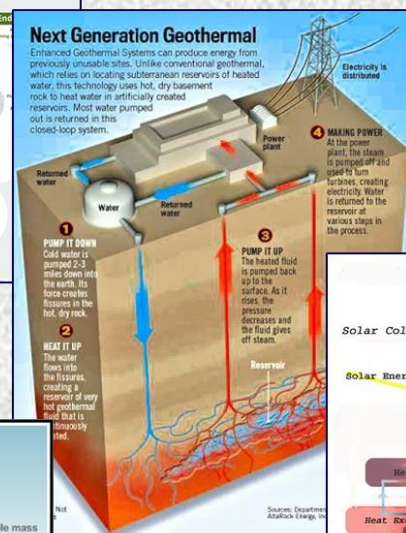
1) Need or Problem
2) Solution Alternatives
3) Solution Boundaries
4) Solution Selection
5) Feasibility Check
6) System Requirements
7) System Design

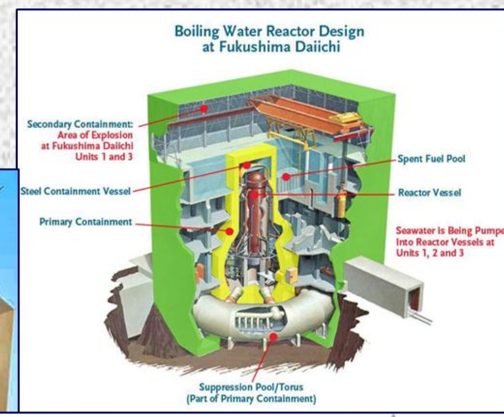
Describe Solution & Early Feasibility Check

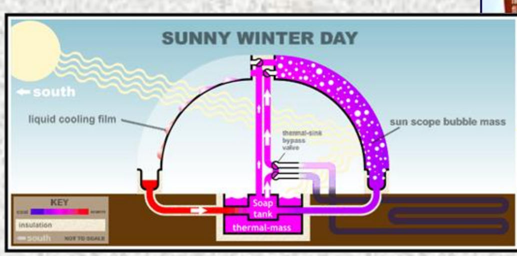
Following Solution Selection

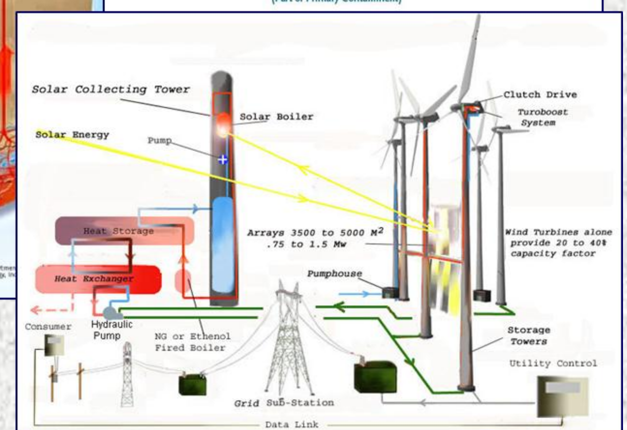
Describe the Solution











Early Feasibility Check

Not a detailed business case, but a real world discussion to know if this solution will be viable

____

____

____

____

____

____

____

____

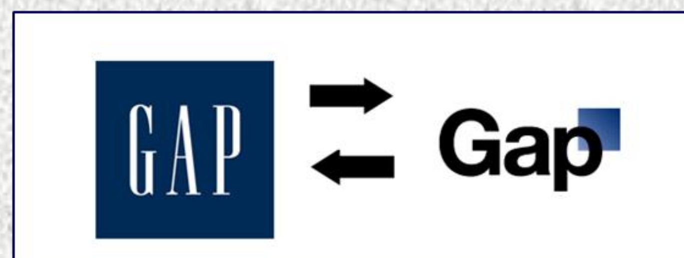
____

____

____

- Will this solution really work?
- What's wrong with this solution?
- Will the needed outcomes result?
- Any showstopper obstacles?
- Will the solution be accepted by all stakeholders?
- Are early cost and schedule estimates acceptable?
- Are resources available?
- Does management and the customer support this?
- Are there any legal or regulatory showstoppers?
- ______________________________________?
- ______________________________________?

Why Do Feasibility Checks?
Some Forgot to Do IT



1) Need or Problem
2) Solution Alternatives
3) Solution Boundaries
4) Solution Selection
5) Feasibility Check
6) System Requirements
7) System Design

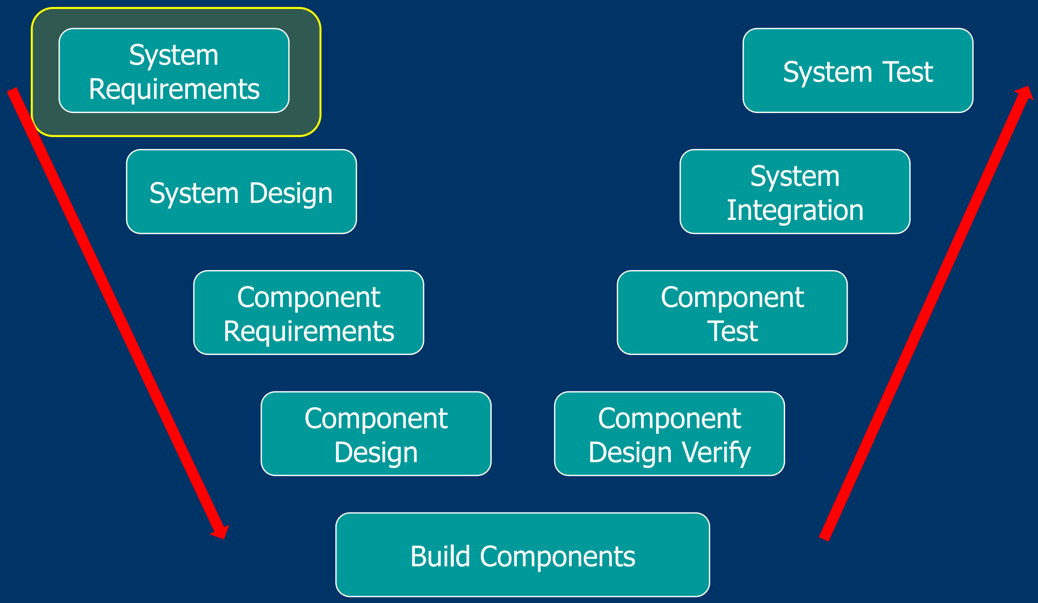
Systems Engineering Defines

System Requirements

System Lifecycle

System Requirements

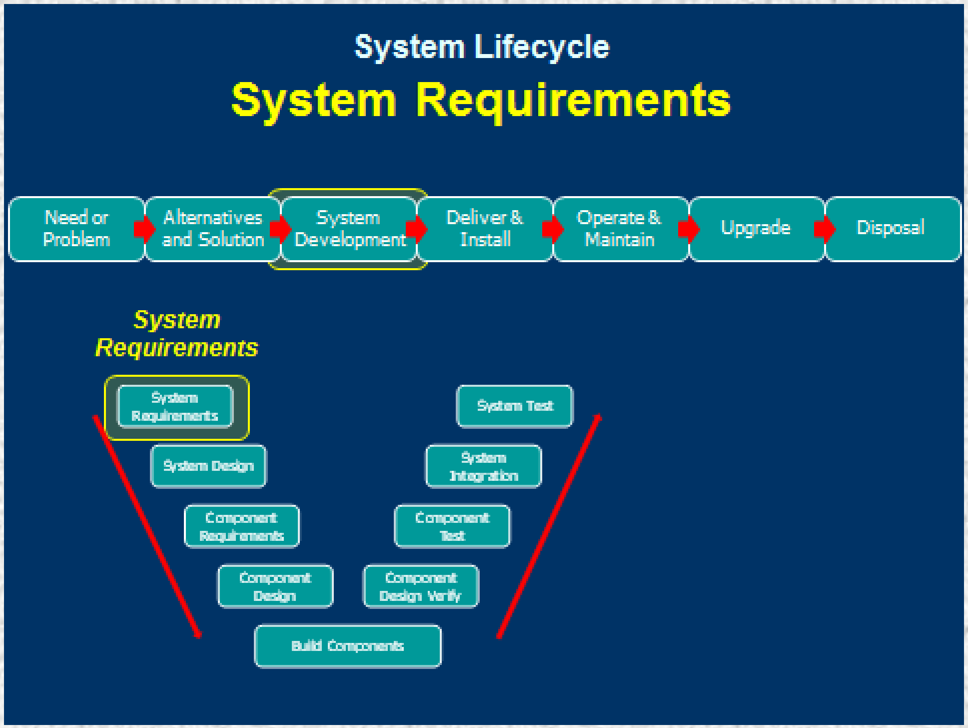
System Requirements



What Are System Requirements?

System Requirements

- The set of information that describes the purpose, functions, performance and constraints related to the respective system
- Follows system solution concept selection
- Is the start of system development
- Precedes system design



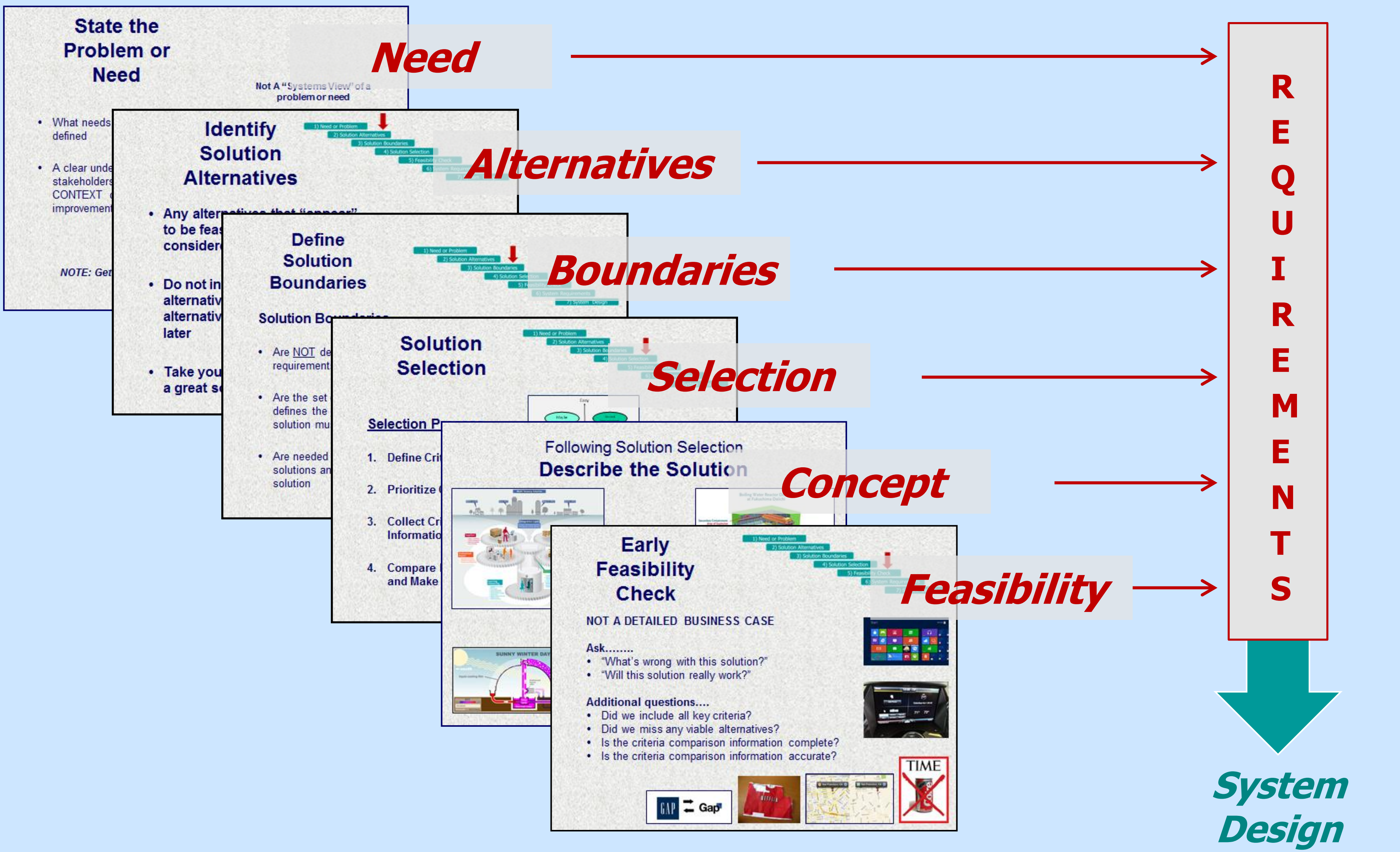
Requirements Timing

Requirements Typically Evolve Over Time
vs. Surfacing in a Single Event

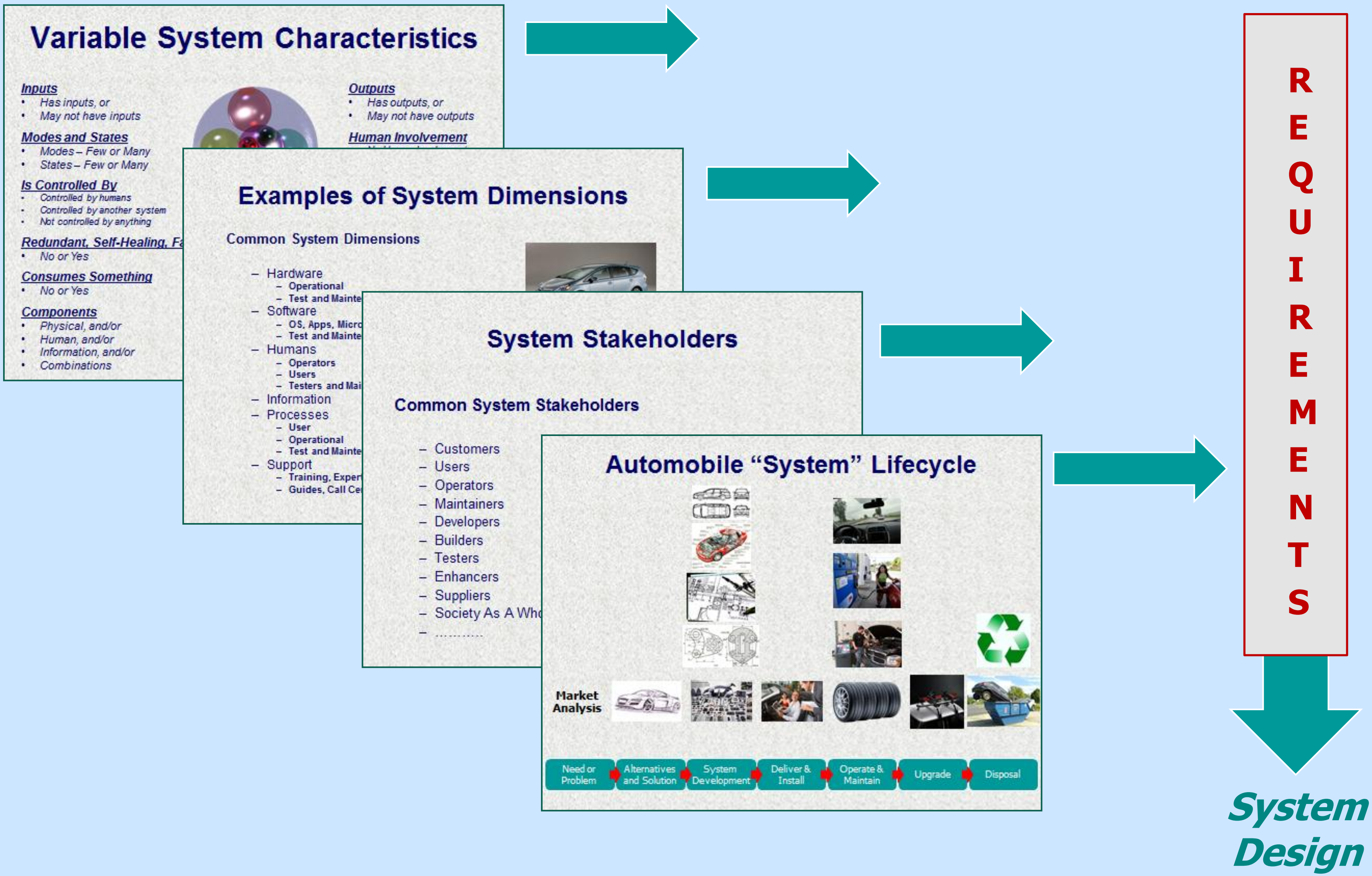
Usually

Usually Not

All Steps Drive System Requirements Definition at Increasing Levels of Detail and Maturity



Variables, Dimensions, Stakeholders and Lifecycle Needs Drive System Requirements



General System Requirements Categories

Functional – "*What It Does*"

Performance – "*How Well It Does It*"

- Speeds, Capacities, Throughput, Efficiencies, ……
- Up Time, Key Performance Measures, Reliability, Maintainability, …..

Constraints – *"Boundaries the System Must Fit Inside"*

- Impacts – Adjacent Systems, Adjacent Processes, Stakeholders, ……
- User – Skills, Interests, Capabilities, Demographics, …….
- Costs – Develop, Operate, Maintain, Dispose
- Safety – Operators, Users, Society, Maintenance Personnel, ……
- Physical – Size, Weight, Power Consumption
- Built-in Redundancy – Yes or No, Automatic or Not
- Maintenance – Test, Fault Isolation, Repair Times, Skills, Access, …….
- Upgradability – Yes, No, ……
- Compatibility – With Other Systems (current and prior), Interfaces, Processes, ….
- Conforming Rules – Standards, Regulations, Laws, ……
- Usage Environment – Temperature, Moisture, Sunlight, Wind, EMI, ……
- Environmental Impact Constraints – To Air, Ground, Water, ……..
- Disposal Limitations

Requirements Failures in the Commercial World

Year	Company	Outcome (Cost in US$)
2005	Hudson Bay Co. (Canada)	Problems with inventory system; $33.3M loss
2004-5	UK Inland Revenue	Software errors contribute to $3.45B tax credit overpayment
2004	Avis Europe PLC (UK)	Enterprise Resource Planning system cancelled after $54.5M spent
2004	Ford Motor Co.	Purchasing system abandoned after costing $400M
2004	J. Sainsbury PLC (UK)	Supply chain management system abandoned after deployment costing $527M
2004	Hewlett-Packard Co.	Problems with ERP contribute to $160M loss
2003-4	AT&T Wireless	Customer Relations Management (CRM) upgrade problems lead to loss of $100M
2002	McDonald's Corp.	Information purchasing system cancelled after $170M spent
2002	Sydney Water Corp (Aus)	Billing system cancelled after $33.2M spent
2002	Cigna Corp.	Problems with CRM system contribute to $445M loss
2001	Nike Inc.	Problems with supply chain management system contribute to $100 loss
2001	K-Mart Corp.	Supply chain management system cancelled after $130M spent

From "Why Software Fails", http://www.spectrum.ieee.org/print/1685

Common Requirements Issues and Solutions

Issue	Solution
• No Real Requirements Existed	***DEFINE THEM***
• Stakeholders not Involved	***INVOLVE THEM***
– …known…	***……***
– …acceptance…	***……***
– ...understand….	***.……***
• Incomplete	***REVIEW and COMPLETE***
• Not Mutually Compatible	***REVIEW and CHANGE***
• Not Documented	***DOCUMENT***
• No Change Management	***MANAGE***
• Not Communicated	***COMMUNICATE***

Requirements Examples
Cell Phone

Functional

- **Modes** – Phone, App Appliance, Camera, …..
- **Interface** – Cell Communications, User Screens, Bluetooth, Menus, …..

Performance

- **User Interface** – Screen Brightness, Speed, Battery Life, …..
- **Capacities / Rates** – Memory, Data Rates, ….
- **Quality** – Voice Quality, Lifetime, reliability, ….
- **Operational** – Size, Sound Quality, Battery Life, Image Quality, …..

Constraints

- **Physical** – Size, Weight, Drop Resistance, Scratch Resistance, Water Resistance, ……
- **Costs** – Production Cost

Requirements Examples
Automobile

- **Functional** – Accommodates 5 People, 4WD,
- **Performance** – MPG, Acceleration, Braking, Crash Performance, Reliability,
- **Physical** – Exterior Size, Weight, Interior Size, Tow Capacities, ...
- **Interface** – Driver Controls, Instrumentation,
- **Modes** – On, Off, Parked, Forward, Backward, ….
- **Quality** – Lifetime, Reliability, Finishing Detail, ….
- **Maintenance** – Amount, complexity, consumables, skills needed
- **Disposal** – Recycle requirements

Requirements Example
Building

Functional

- **Modes** – Day Populated, Night Vacant, Holiday Mode, Emergency
- **Interfaces** – Water, Power, Sewer, …

Performance

- **User Interface** – Traffic, environmental control, windows, entrances,
- **Operational** – Energy efficiency, ease of re-configuration, ……

Constraints

- **Physical** – Size, wind, solar, rain, earthquake, flooding, building codes, ...
- **Quality** – Lifetime, reliability, Esthetics,…..
- **Costs** – Build, Operate, Maintain
- **Maintenance Limitations** – Cleaning, Painting, ……

Characteristics of good requirements

Reference

Unitary - The requirement addresses only one thing.

Complete and Standalone - The requirement is complete and does not require the existence or understanding of other requirements.

Consistent - The requirement does not contradict any other requirement.

Traceable - The requirement can be traced to some authority (person, document) that caused the requirement to exist.

Characteristics of good requirements

Reference

Current - The requirement has not been made obsolete by the passage of time.

Feasible - The requirement can be implemented.

Unambiguous – The requirement means the same thing to all readers and is not miss-interpretable.

Defines Importance – Some level of importance is known or is included with each requirement or groups of requirements.

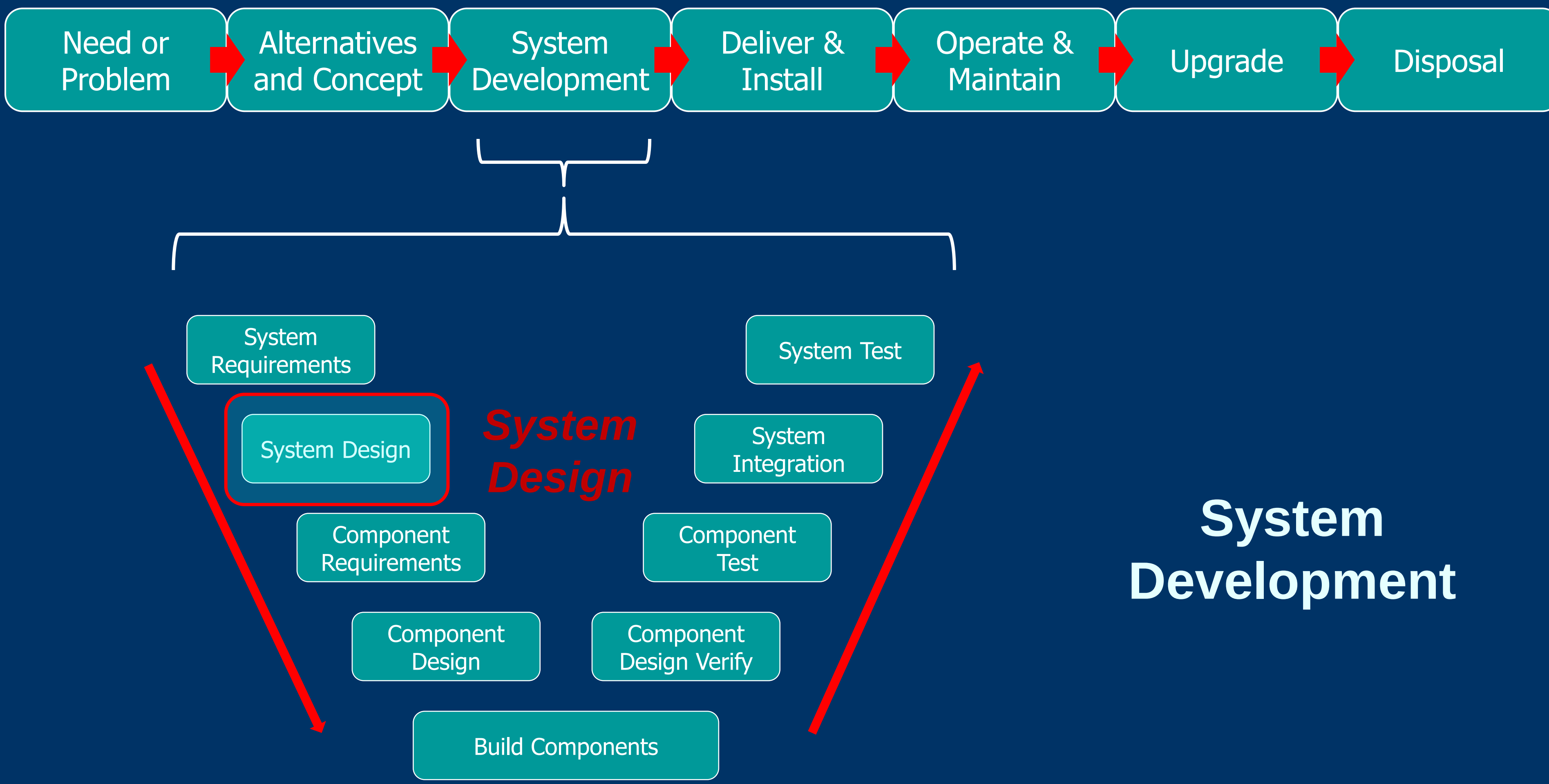
Verifiable - The requirement can be determined to have been met, i.e. the requirement can be tested and validated in some manner.

1) Need or Problem
2) Solution Alternatives
3) Solution Boundaries
4) Solution Selection
5) Feasibility Check
6) System Requirements
7) System Design

Systems Engineering Develops the

System Design

System Lifecycle

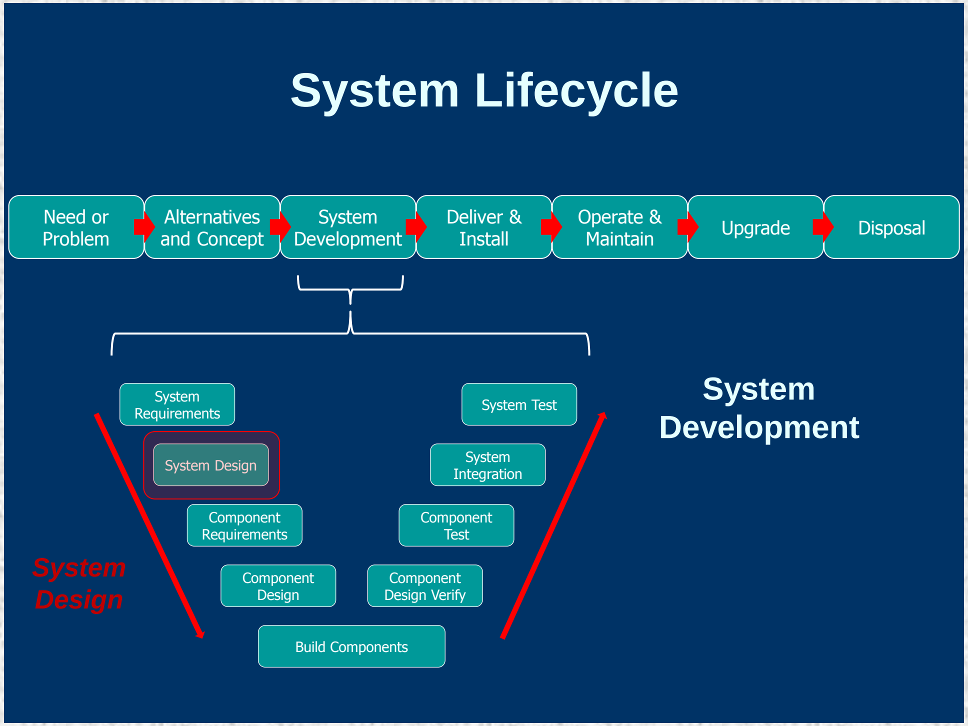


System Development

What Is System Design?

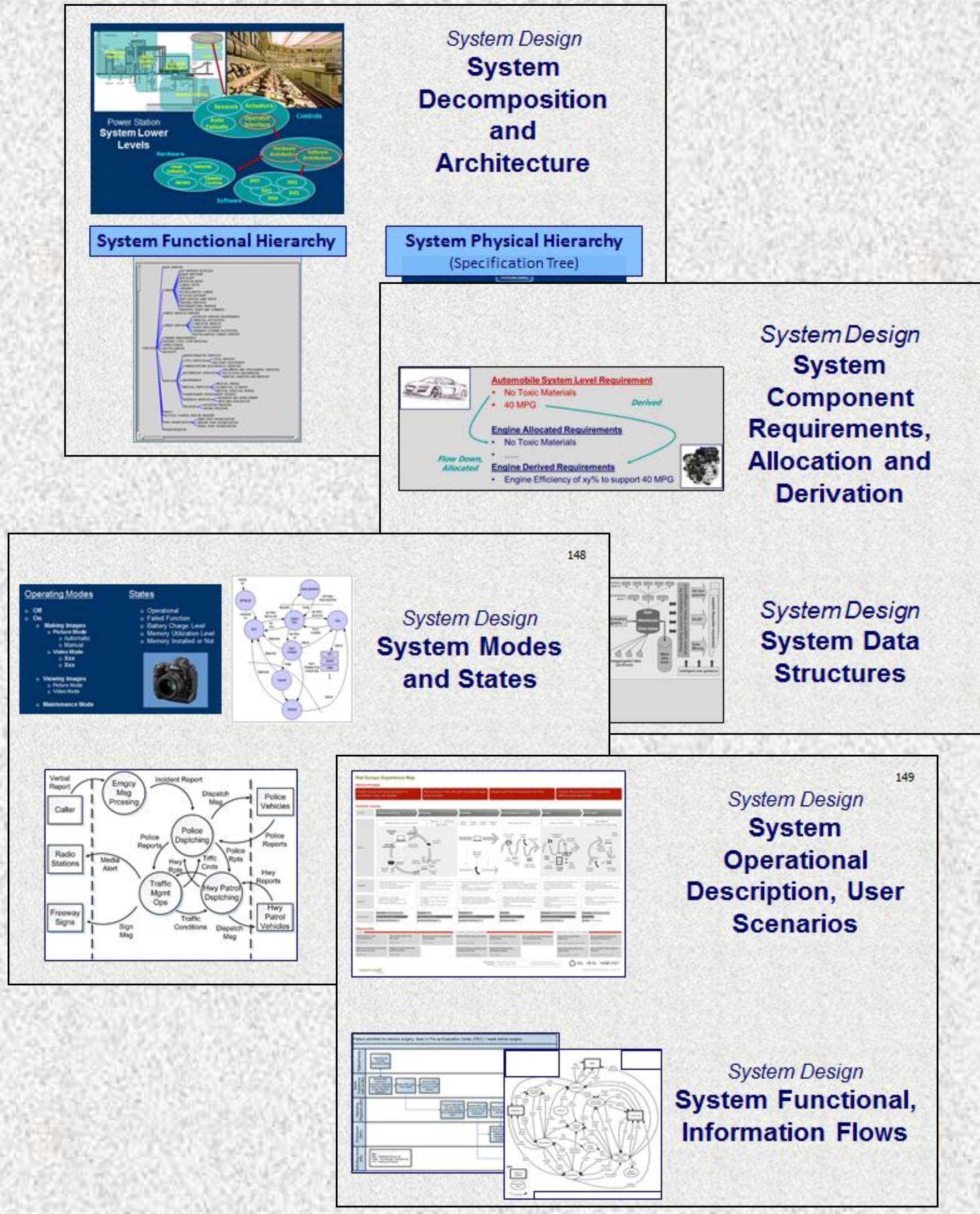
System Design

- The activity that forms the system architecture of system components, defines the functions and requirements of each component, the interfaces between components and the interfaces to the external environment.
- Follows system requirements.
- Precedes requirements for system components.



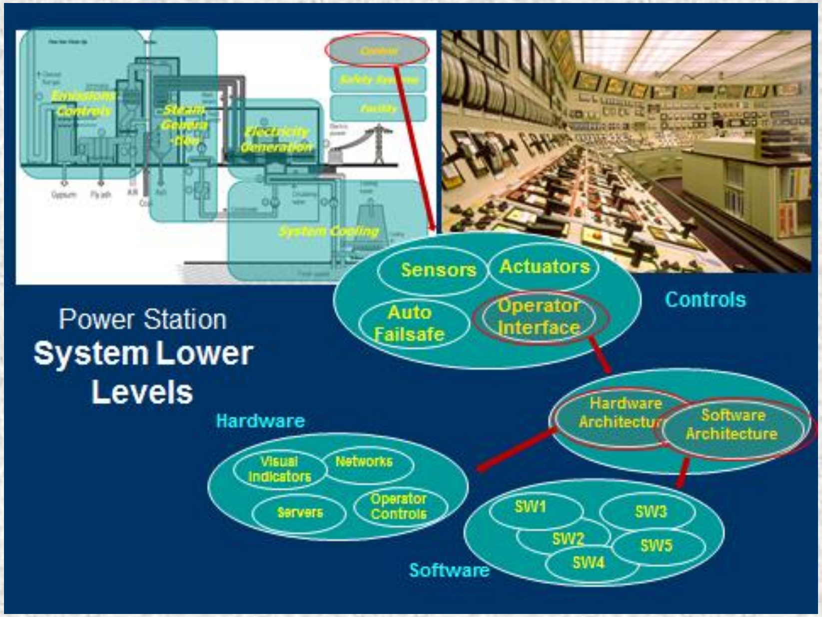
System Design

- Has Multiple Components
- Components are often done in parallel
- Significant iteration and overlap in system design component development is appropriate

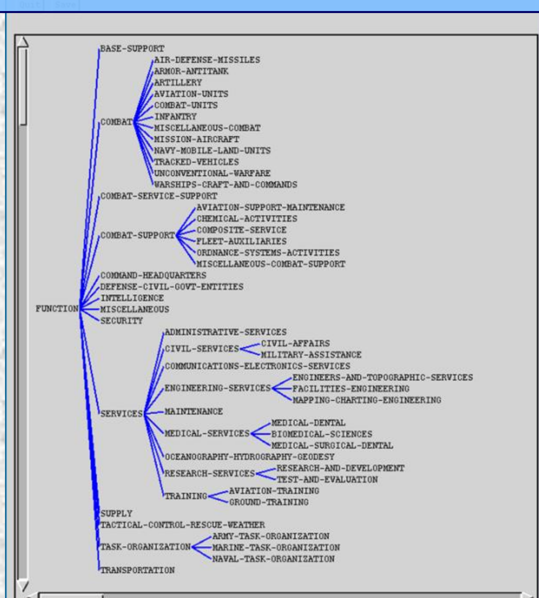


System Design

System Decomposition and Architecture

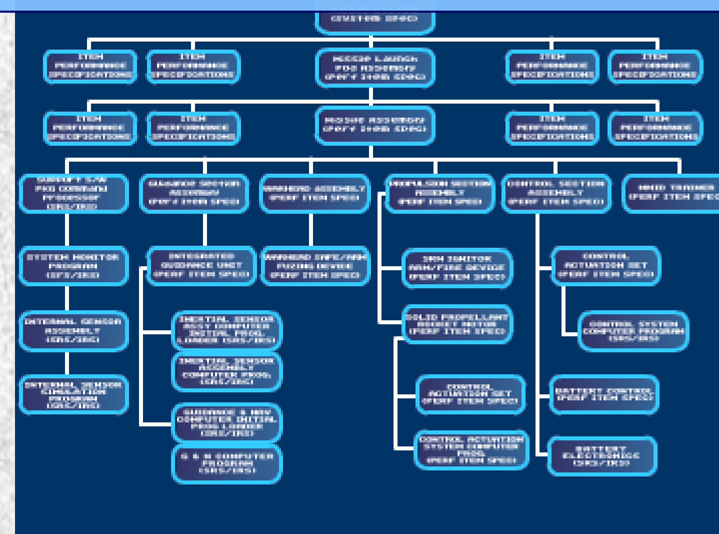


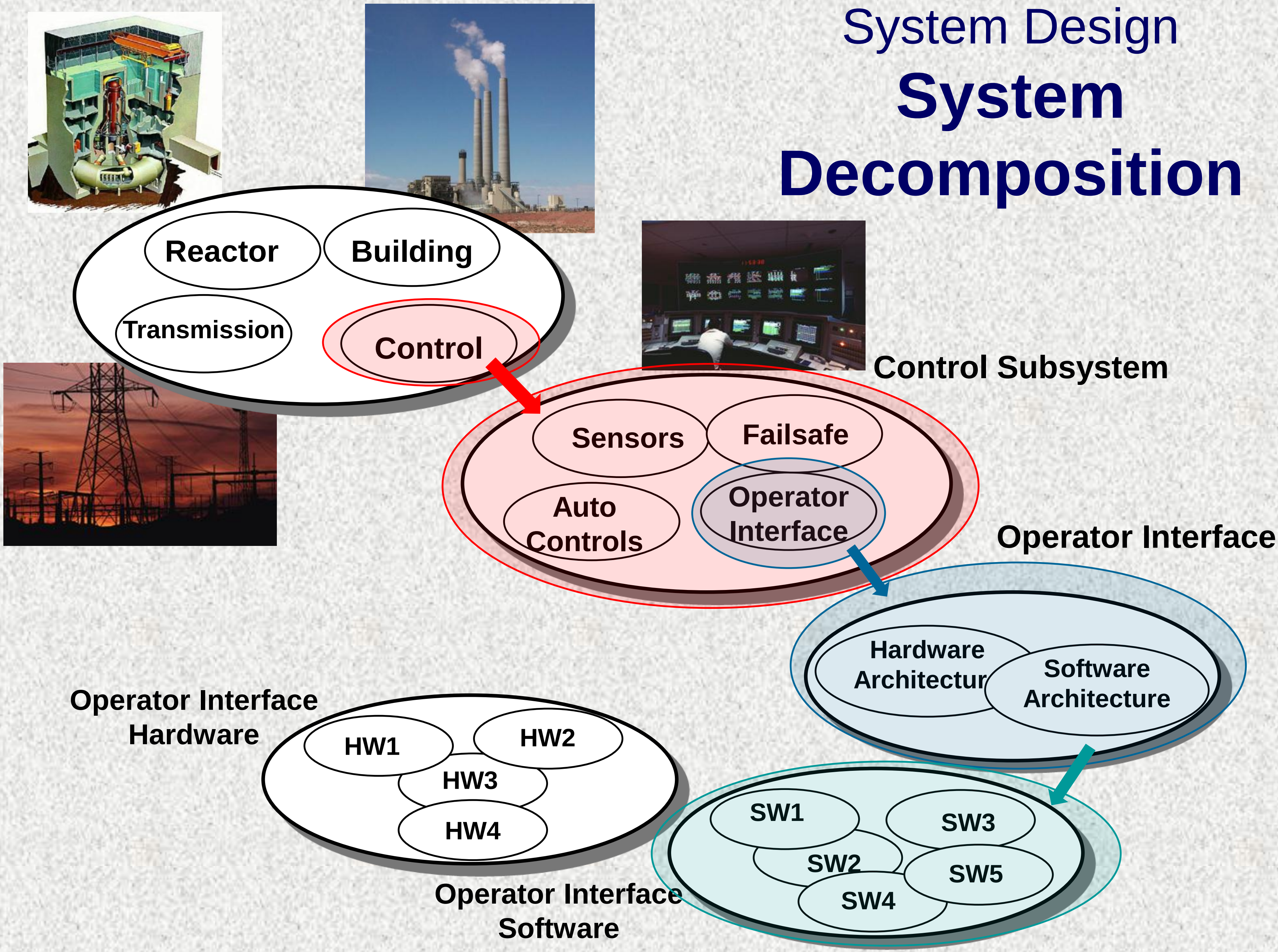
System Functional Hierarchy

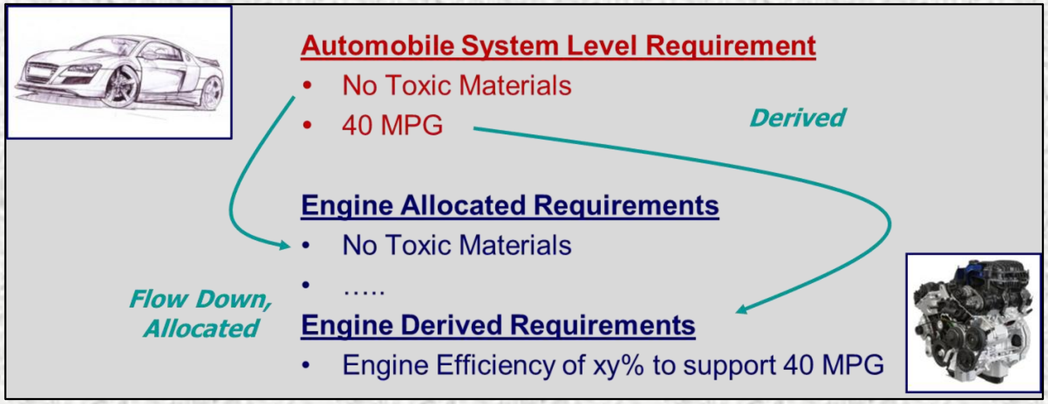


System Physical Hierarchy

(Specification Tree)

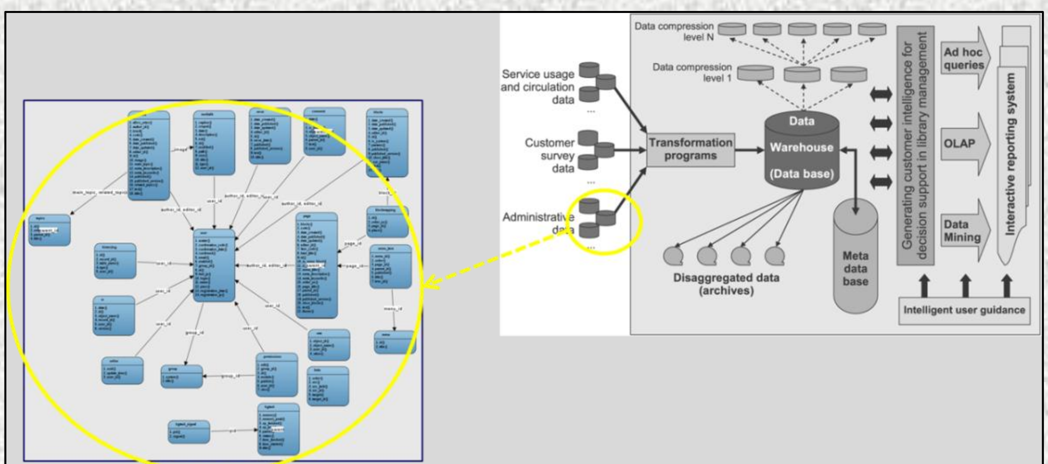



System Design
System Decomposition
Reactor
Building
Transmission
Control
Control Subsystem
Sensors
Failsafe
Auto Controls
Operator Interface
Operator Interface
Hardware Architectur
Software Architecture
Operator Interface Hardware
HW1
HW2
HW3
HW4
Operator Interface Software
SW1
SW2
SW3
SW4
SW5



System Design

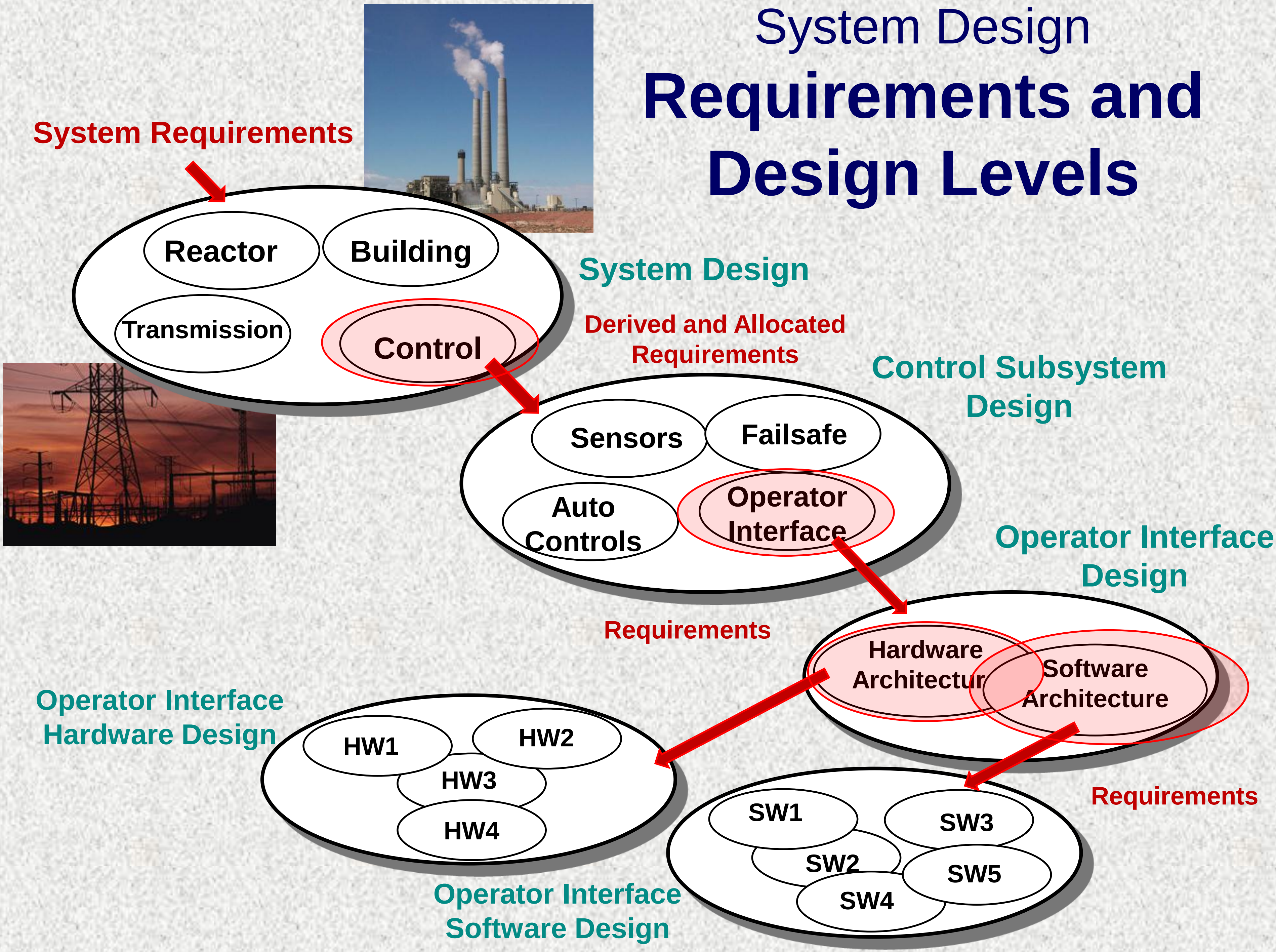
System Component Requirements, Allocation and Derivation



System Design

System Data Structures

System Design
Requirements and Design Levels

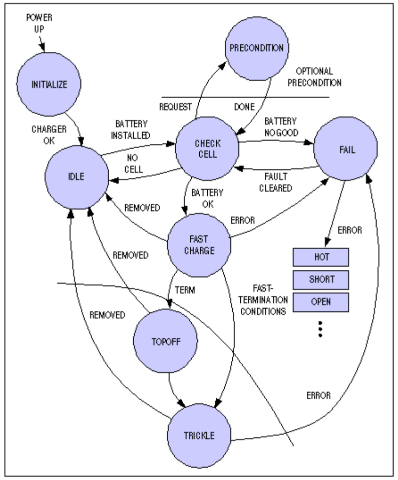


Operating Modes

- Off
- On
  - Making Images
    - Picture Mode
      - Automatic
      - Manual
    - Video Mode
      - Xxx
      - Xxx
  - Viewing Images
    - Picture Mode
    - Video Mode
  - Maintenance Mode

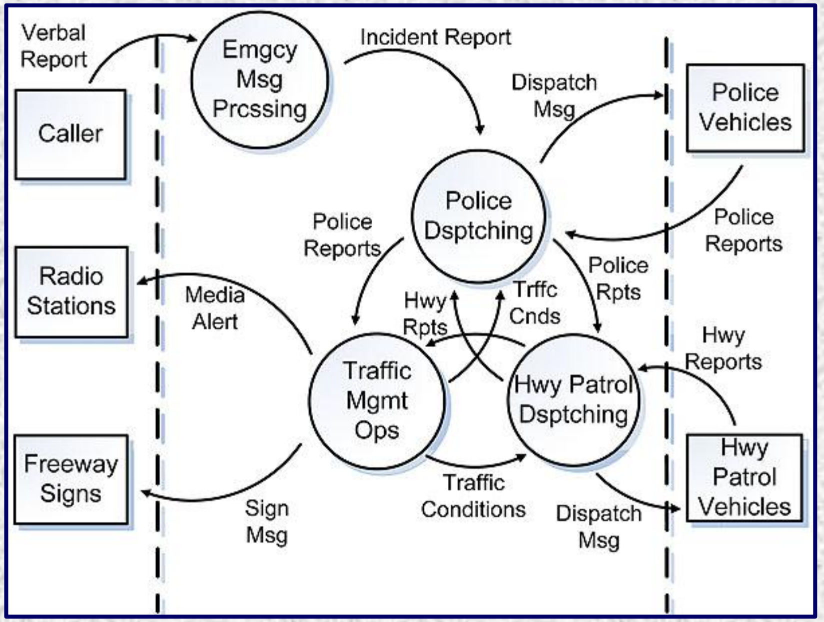
States

- Operational
- Failed Function
- Battery Charge Level
- Memory Utilization Level
- Memory Installed or Not



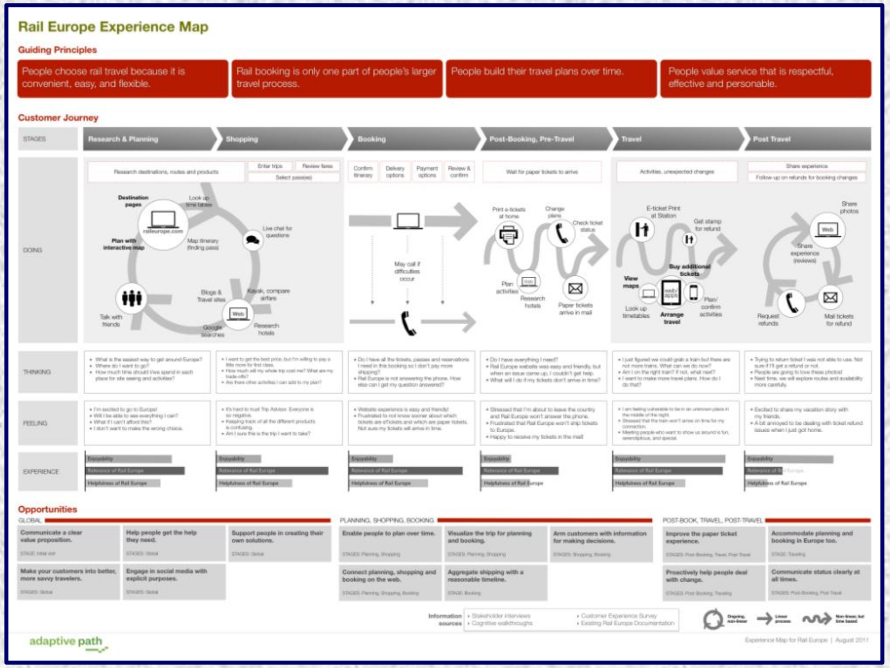
System Design

System Modes and States



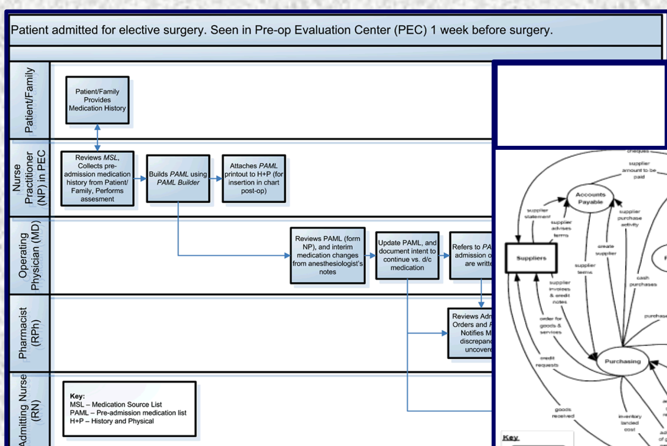
System Design

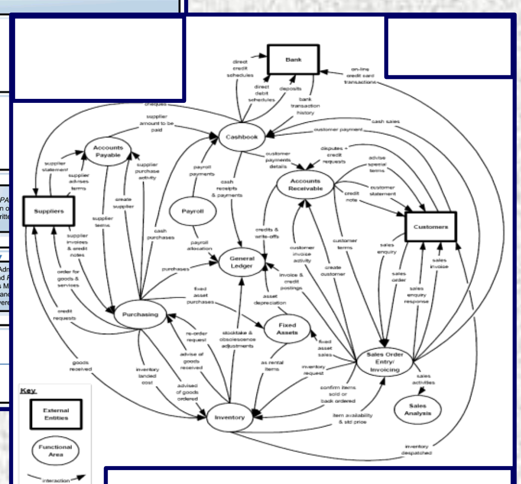
System Internal and External Interfaces



System Design

System Operational Description, User Scenarios





System Design

System Functional, Information Flows

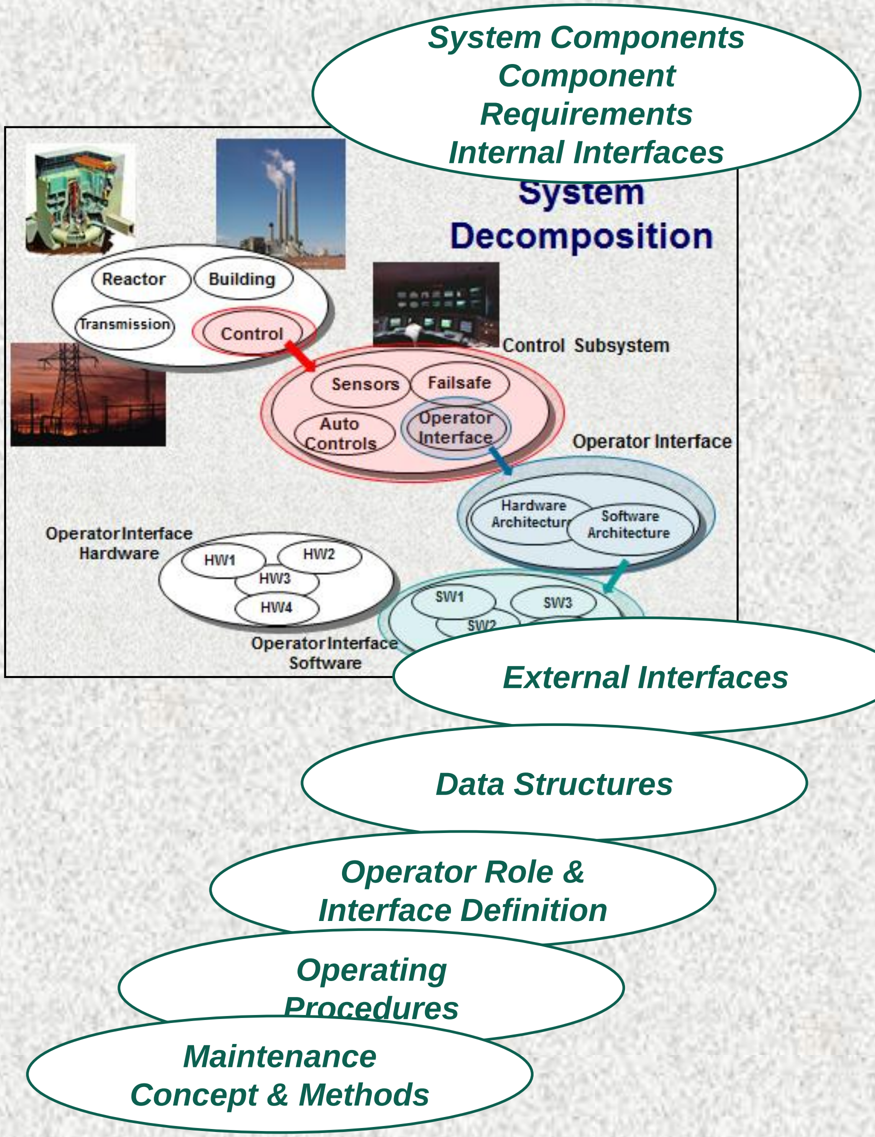
System Design Outputs

- System Components
- Requirements for each System Component

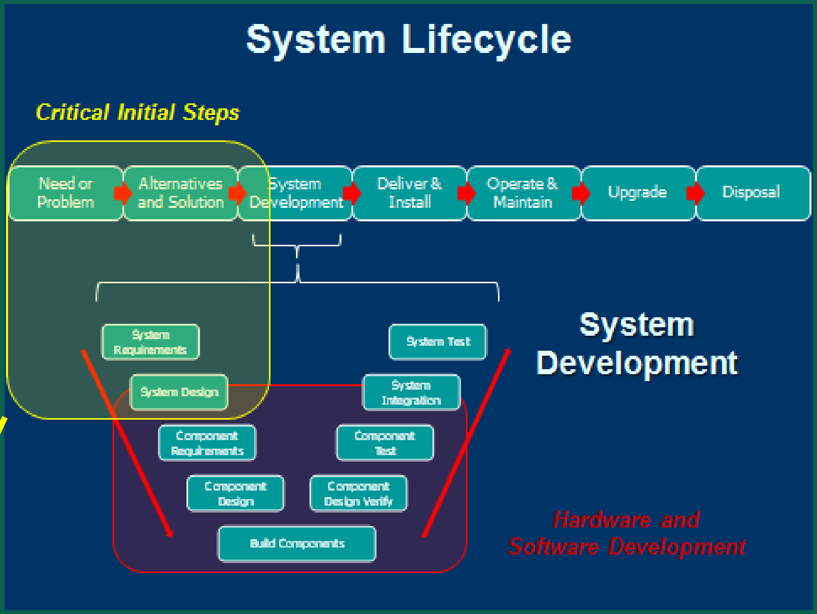
- INTERNAL Functional and Physical Component to Component Interfaces
- EXTERNAL Functional and Physical Interfaces

- System Data Structures
- Operator Role, Interfaces and Skills

- General Operating Procedures
- Maintenance Concept and Methods



Summary



1) Need or Problem

2) Solution Alternatives

3) Solution Boundaries

4) Solution Selection

5) Feasibility Check

6) System Requirements

7) System Design

Early System Development Steps

What To Do

Reference

You Are A Client

- Understand early development steps
- Support your contractor in ensuring all steps are completed

You Are Management

- Understand early development steps
- Educate project leads
- Make sure project teams complete all appropriate steps
- Don't "short-circuit" the process

You Are An SE, Project Manager or Technical Lead

- Understand early steps
- Educate team members
- Make sure your team completes all steps
- Don't be pressured into skipping steps, everyone will pay later

You Are A Team Member

- Understand early steps
- Voice concern if steps are skipped, everyone will suffer later if key steps are not accomplished

The Role of the Systems Engineer

What Is Systems Engineering?

What Can Go Wrong In Developments?

Partial List

1) The Problem or Need is Misunderstood
2) Alternative Solutions Not Identified and Evaluated
3) Wrong Technical Solution Selected
4) All System Level Requirements Not Defined
5) Requirements Not Mutually Compatible or Misunderstood
6) All Appropriate System Dimensions Not Addressed
7) All System Lifecycle Phases Not Addre
8) Poor Overall System Level / Architectu
9) Detail Requirements Before System Le
10) Users Insufficiently Involved in Develop
11) Maintenance Approach Not Considere
12) Upgradeability Not Considered in Requ
13) Stakeholders Not on the Same Page fo
14) Inaccurate Estimates of Time and Res
15) Inadequate Lower Level Requirements

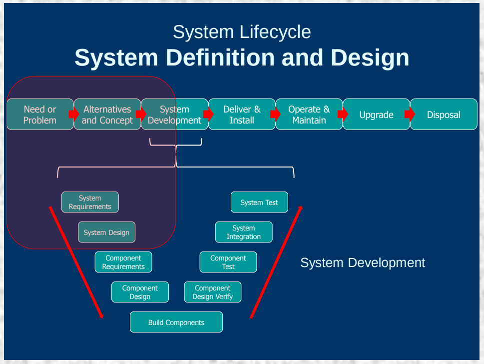
What Can Go Wrong In Developments?

Partial List Continued

16) Inadequate Interface Definitions
17) Inadequate Testing at Lower Levels Prior to System Integration
18) Lacking or Poor Change Management
19) Poor Work Plan
20) Inadequate Developer Technical Skills
21) Inadequate Development Processes or Tools
22) Development Methods Not Aligned with Complexity
23) Poor Management of Project Work and Issues
24) Inadequate Stakeholder Communications
25) Inadequate Internal Team Communications
26) Inadequate Accountabilities Defined
27) Inadequate Technical Skills in Decision Makers
28) No Risk Management
29) Inadequate Sponsor Management
30) Over Designed

Systems Engineering Addresses These Pitfalls

What Is Systems Engineering?

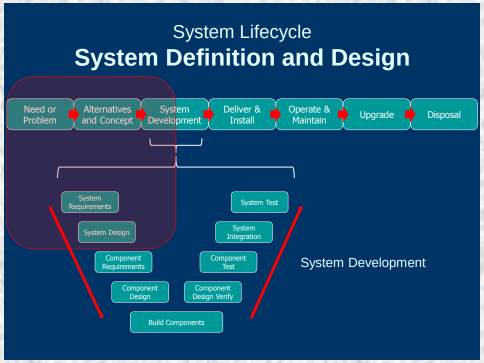


The technical discipline that:

- Fully <u>understands a specific problem</u> or need
- Identifies a set of <u>alternative solutions</u>
- Understands needs of <u>all stakeholders</u>
- <u>Considers all lifecycle phases</u> during requirements generation
- <u>Selects the best solution</u> considering all solution criteria that includes all stakeholders and all timeframes
- Addresses all <u>system dimensions</u>
- <u>Defines the solution detail requirements</u>

What Is Systems Engineering?

(Continued)



The technical discipline that:

- Designs the solution <u>at the system level</u> that…….
  - Meets all requirements
  - Takes into consideration of all dimensions
  - Addresses the needs of all system stakeholders
  - Forms an adequate foundation for the development of all system components
- <u>Defines and allocates requirements to system components</u>
- <u>Oversees component development</u>
- Conducts system <u>integration and testing</u>
- <u>Validates and verifies</u> the resulting system meets all needs
- Supports System Implementation, Use, Maintenance and Disposal

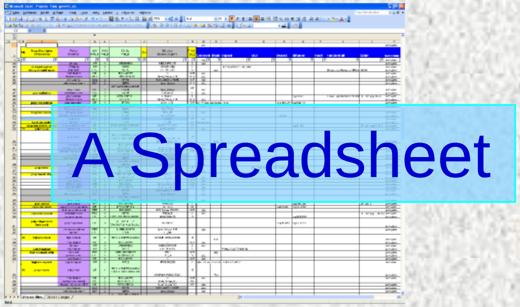
Does Systems Engineering Apply to Simple Developments? Yes

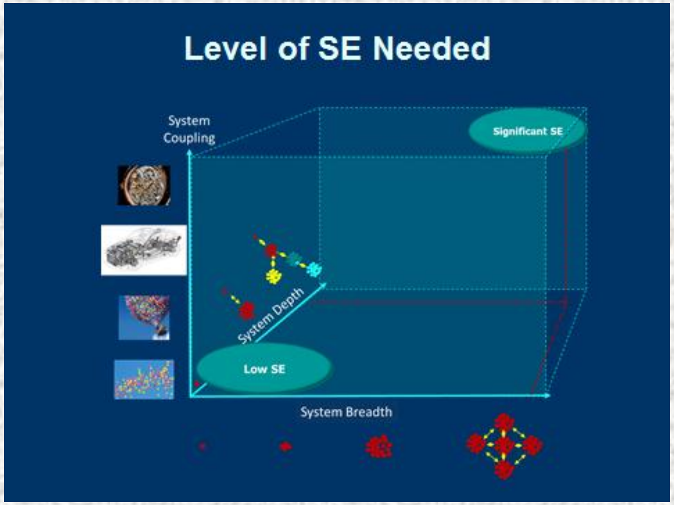
Requirements
- Functions
- Interfaces
- Impacts

Stakeholders
- User
- Maintainer
- Integrator

Dimensions
- Hardware
- Software
- People
- Information
- Procedures
- Support

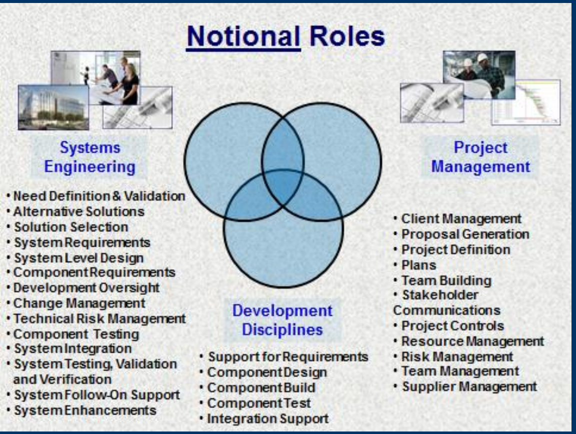
Lifecycle
- Defined Need
- Alternatives
- Develop
- Build, Deliver
- Install, Train
- Use
- Operate
- Maintain
- Upgrade
- Dispose

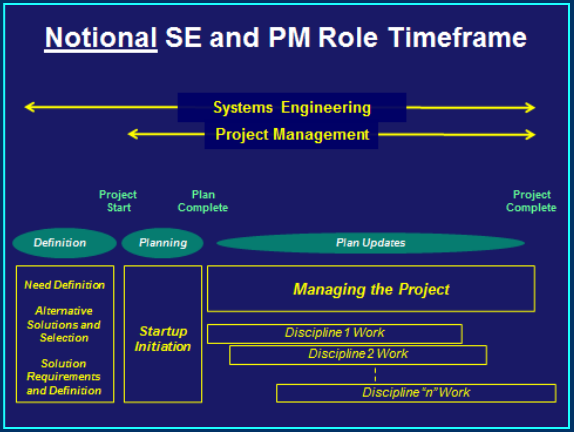




Systems Engineering
SE Role vs. People

The Role





The SE <u>role</u> may be performed by:

- A dedicated systems engineer
- An organization that performs systems engineering functions
- The project manager also acting in an SE role
- A technical lead acting in an SE role
- The team performing SE functions

System Engineering Role

What To Do

Reference

You Are A Client

- Determine if your contractor understands and applies systems engineering fundamentals and has a defined SE role or function

You Are Management

- Understand the SE role
- Ensure SE functions exist on projects
- Educate teams on SE
- Ensure each project uses SE fundaments, validate during project startup

You Are An SE, Project Manager or Technical Lead

- Understand the SE role and how it is applied on projects
- Have an SE function on projects
- Educate teams on SE
- Explain why SE is needed

You Are A Team Member

- Be aware of SE and the importance
- Ask questions of leadership about SE functions where they appear not exist

Conclusion

SUMMARY
What Is Systems Engineering?

Systems Engineering is a Function and a Discipline.

SE is a Mindset.

SE Is Not Necessarily a Person or a Role or an Organization, but Can Be.

Systems Engineering is Needed on Virtually any development.

Educating Others
Team, Management and Customers

- The need for clear problem statements
- Alternative solutions and criteria to choose the best
- The system development process – AS IT SHOULD EXIST IN YOUR ENVIRONMENT
  - Steps and Importance
  - Timing
  - Funding
  - Ramifications of Non-Adherence
- The Differences Between Different Project Types - “new construction, remodeling, refactoring”
- Need for an Operating Model
- Use of the right Development Model

Educating Others
What To Do

Reference

You Are A Client

- Understand systems engineering and the need
- Listen to development teams when concerns are expressed

You Are Management

- Understand systems engineering and the need
- Listen to development teams when concerns are expressed
- Educate development teams on SE practices

You Are An SE, Project Manager or Technical Lead

- Understand systems engineering and the need
- Educate your team on SE practices
- “Educate-Up” when needed

You Are A Team Member

- Learn what SE is and how it is accomplished
- Apply SE principles on your projects
- “Educate-Up” when needed

REFERENCE

System Design Documentation Examples

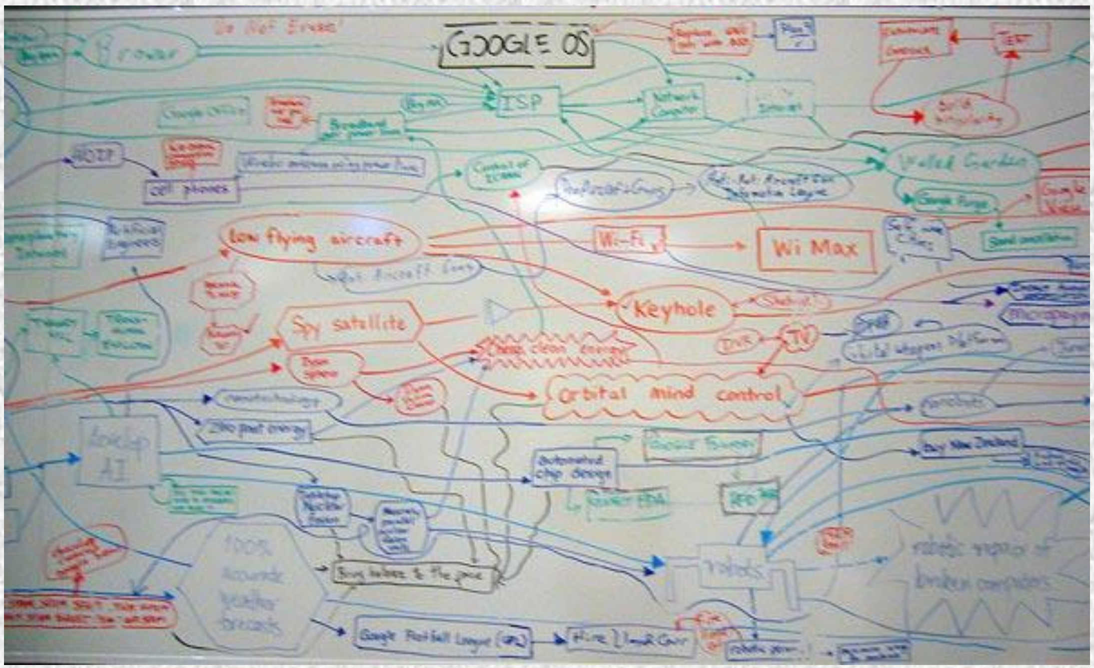
Capturing and Communicating System Information

Communication Needs

Most Project Problems and Failures Involve POOR Communication

Communicating System Design

Ever Find that One Diagram Does Not Meet All Needs?



Lines Are….

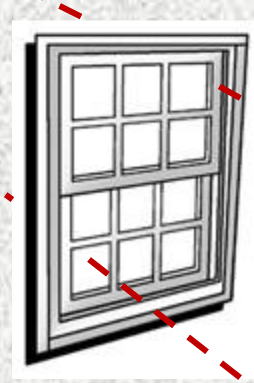
- Information? or
- Physical Elements? or
- A Relationship? or
- A Time Sequence?, or
- (lines are inconsistent)

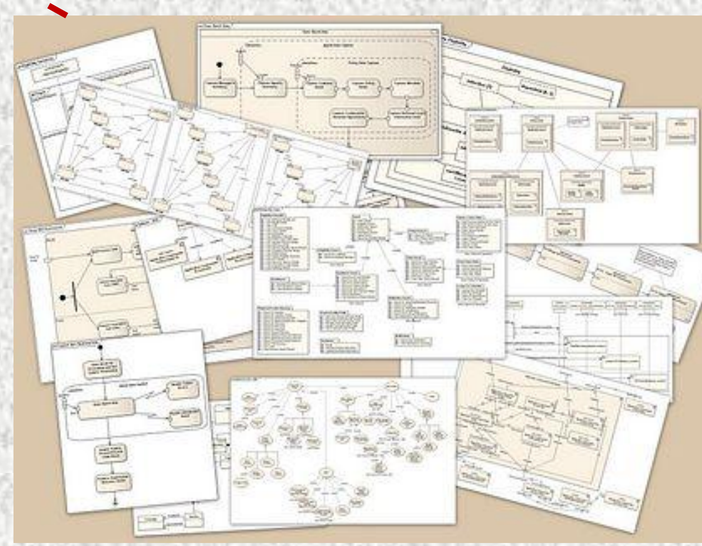
Shapes Are….

- Verbs? or
- Nouns? or
- Conditions? or
- (shapes are inconsistent)

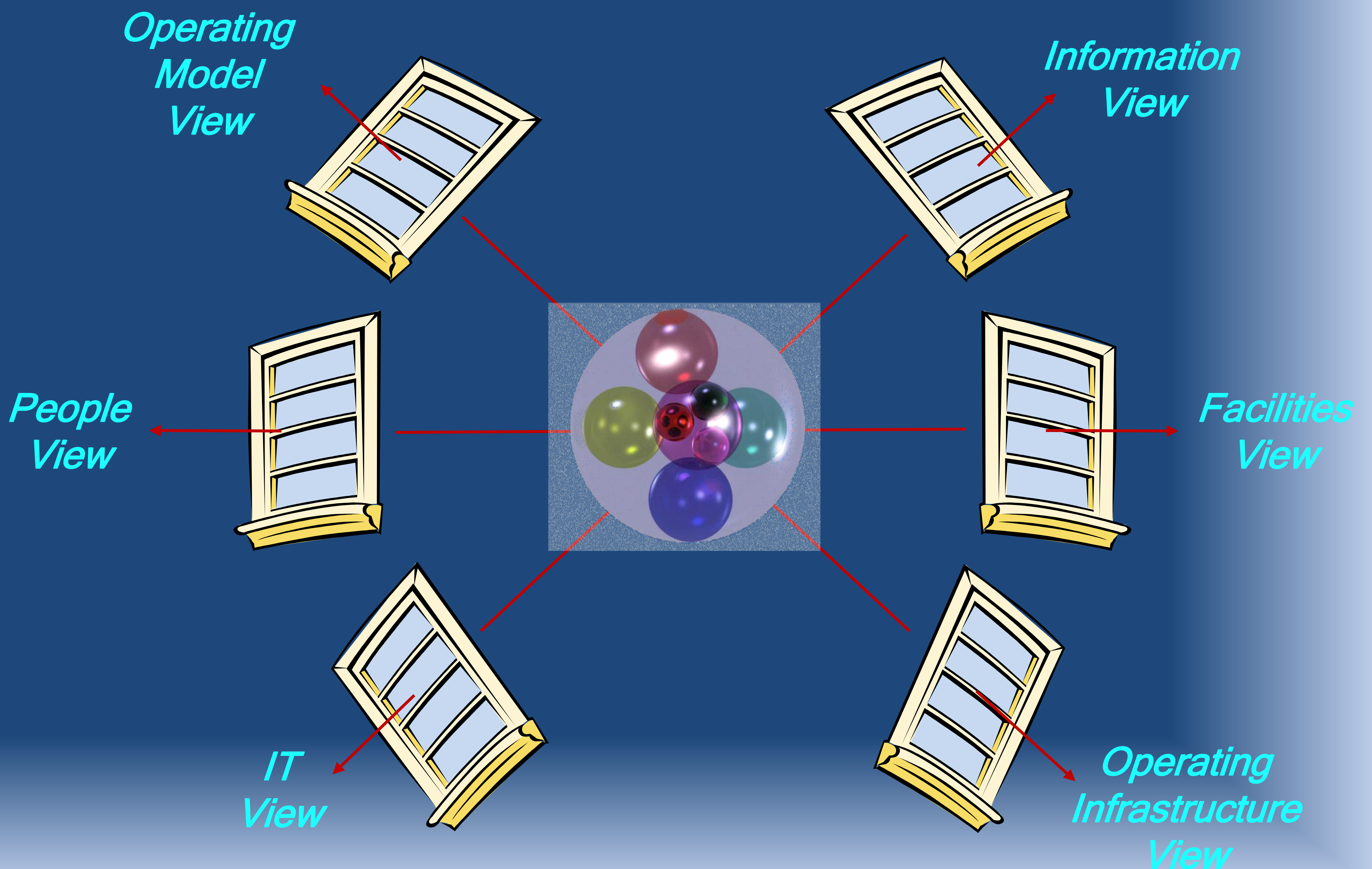
Different Views of A System Are Needed

- Physical Views
- Functional Views
- Information Flow Views
- Data Base Views
- Process and State Views
- Internal Interface Views
- External Interface Views





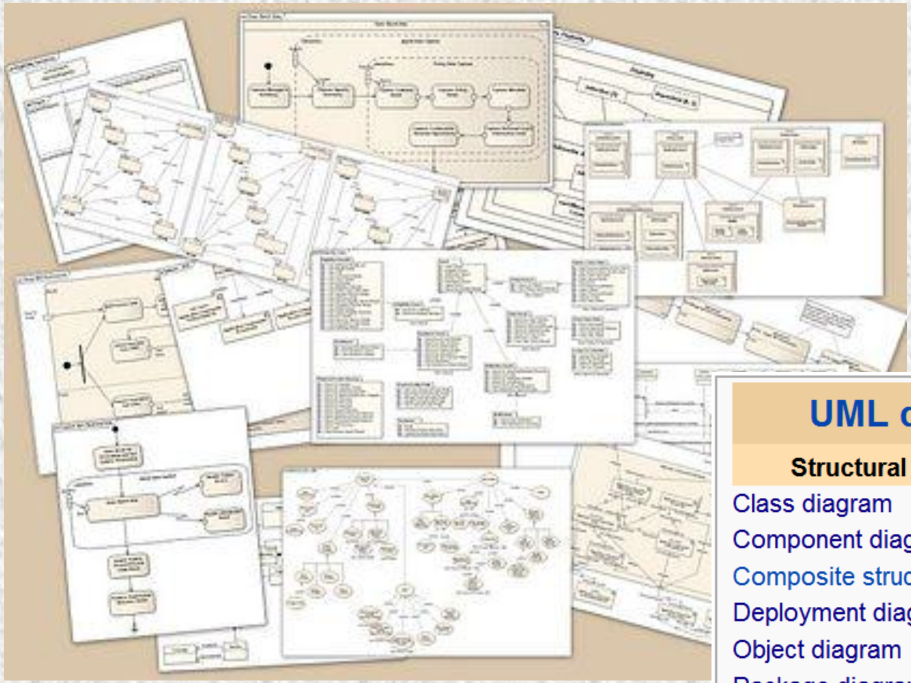
Different Views of A System



UML
Unified Modeling Language

UML

- Class diagrams
- Object diagrams
- Use case diagrams
- Sequence diagrams
- Collaboration diagrams
- Statechart diagrams
- Activity diagrams
- Component diagrams
- Deployment diagrams



UML diagrams
Structural UML diagrams
Class diagram
Component diagram
Composite structure diagram
Deployment diagram
Object diagram
Package diagram
Profile diagram
Behavioral UML diagrams
Activity diagram
Communication diagram
Interaction overview diagram
Sequence diagram
State diagram
Timing diagram
Use case diagram

DoDAF

US Government Architecture Framework

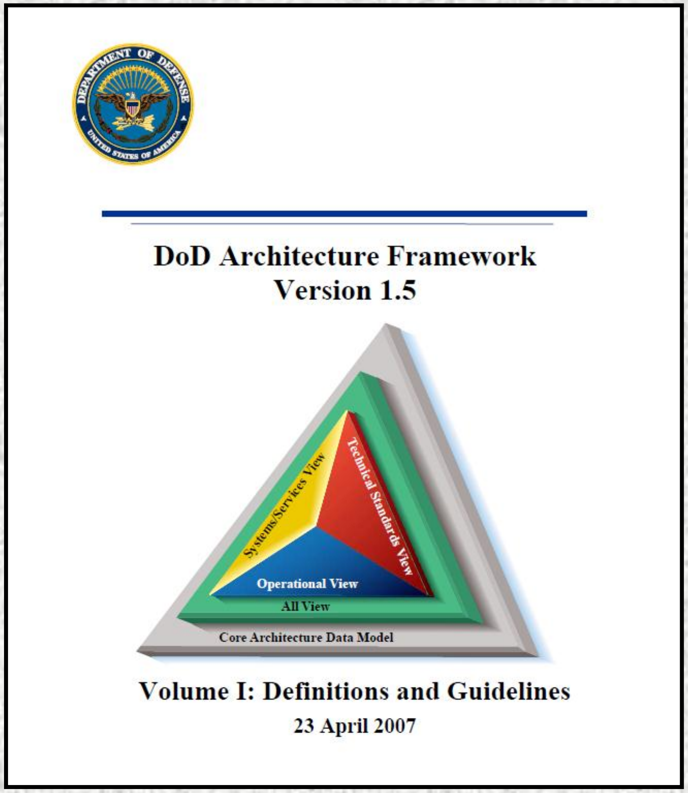


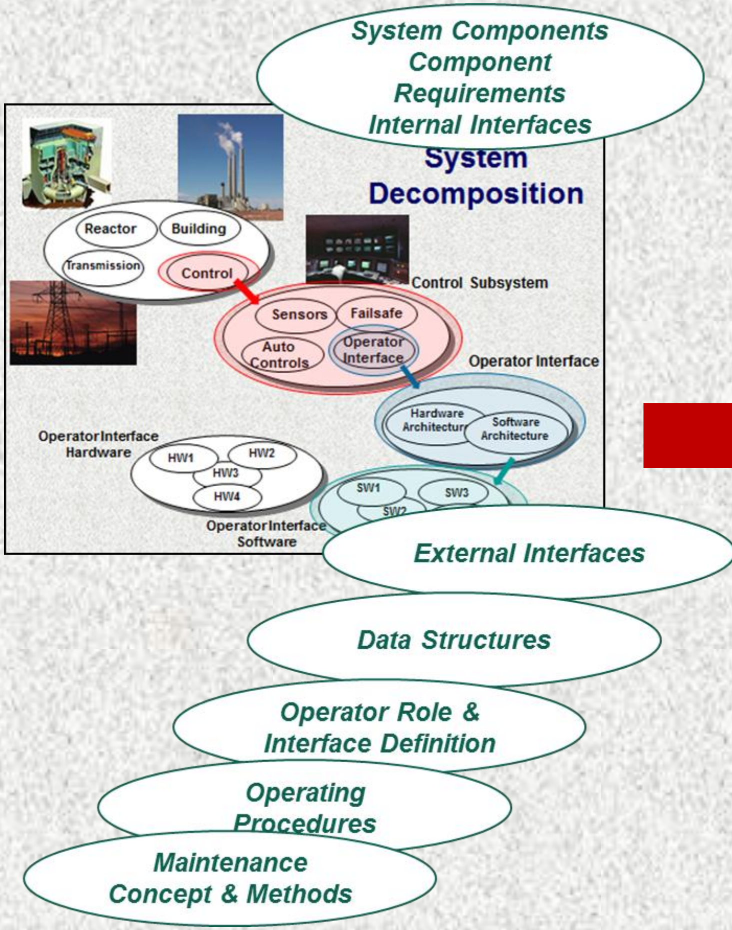
Table 2-1: List of Products

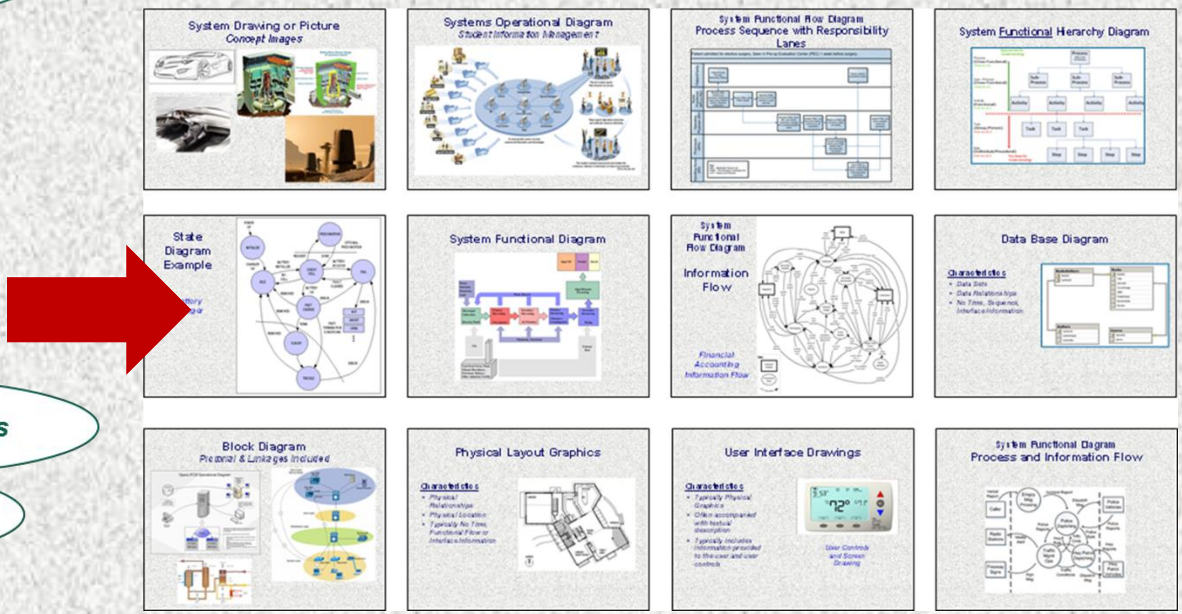
Applicable View	Framework Product	Framework Product Name	Net-Centric Extension	General Description
All View	AV-1	Overview and Summary Information	✓	Scope, purpose, intended users, environment depicted, analytical findings
All View	AV-2	Integrated Dictionary	✓	Architecture data repository with definitions of all terms used in all products
Operational	OV-1	High-Level Operational Concept Graphic	✓	High-level graphical/textual description of operational concept
Operational	OV-2	Operational Node Connectivity Description	✓	Operational nodes, connectivity, and information exchange need lines between nodes
Operational	OV-3	Operational Information Exchange Matrix	✓	Information exchanged between nodes and the relevant attributes of that exchange
Operational	OV-4	Organizational Relationships Chart	✓	Organizational, role, or other relationships among organizations
Operational	OV-5	Operational Activity Model	✓	Capabilities, operational activities, relationships among activities, inputs, and outputs; overlays can show cost, performing nodes, or other pertinent information
Operational	OV-6a	Operational Rules Model	✓	One of three products used to describe operational activity—identifies business rules that constrain operation
Operational	OV-6b	Operational State Transition Description	✓	One of three products used to describe operational activity—identifies business process responses to events
Operational	OV-6c	Operational Event-Trace Description	✓	One of three products used to describe operational activity—traces actions in a scenario or sequence of events
Operational	OV-7	Logical Data Model	✓	Documentation of the system data requirements and structural business process rules of the Operational View

Systems and Services	S			
Systems and Services	S			
Systems and Services	S			
Systems and Services	S			
Systems and Services	SV-4b	Services Functionality Description	✓	Functions performed by services and the service data flow among service functions
Systems and Services	SV-5a	Operational Activity to Systems Function Traceability Matrix		Mapping of system functions back to operational activities
Systems and Services	SV-5b	Operational Activity to Systems Traceability Matrix		Mapping of systems back to capabilities or operational activities
Systems and Services	SV-5c	Operational Activity to Services Traceability Matrix	✓	Mapping of services back to operational activities
Systems and Services	SV-6	Systems Data Exchange Matrix Services Data Exchange Matrix	✓	Provides details of system or service data elements being exchanged between systems or services and the attributes of that exchange

Systems and Services	S			
Systems and Services	S			
Systems and Services	S			
Systems and Services	SV	Services Rules Model		imposed on systems/services functionality due to some aspect of systems design or implementation
Systems and Services	SV-10b	Systems State Transition Description Services State Transition Description	✓	One of three products used to describe system and service functionality—identifies responses of a system/service to events
Systems and Services	SV-10c	Systems Event-Trace Description Services Event-Trace Description	✓	One of three products used to describe system or service functionality—identifies system/service-specific refinements of critical sequences of events described in the Operational View
Systems and Services	SV-11	Physical Schema	✓	Physical implementation of the Logical Data Model entities, e.g., message formats, file structures, physical schema
Technical Standards	TV-1	Technical Standards Profile	✓	Listing of standards that apply to Systems and Services View elements in a given architecture
Technical Standards	TV-2	Technical Standards Forecast		Description of emerging standards and potential impact on current Systems and Services View elements, within a set of time frames

The Output of System Design Is a Set of Documents



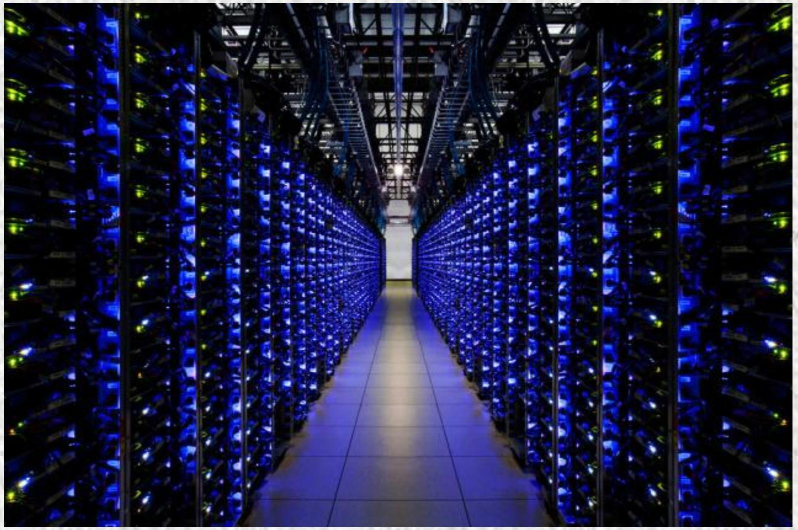


System Design Documentation

Example Documentation

Electronic System

- System Summary Description
- System Operational Diagram
- System Block Diagram
- System Functional Diagram
- System State or Mode Diagrams
- System Data and Data Relationships
- External Interface Diagrams
- System Physical Layout Diagram
- System User Concept Drawings
- List of Hardware
- List of Software
- List of Design and Build Standards
- List of New Development Components
- …………



Example Documentation
Building Construction

- Sketches, Artists Concept
- Architectural Drawings
  - Exterior
  - Interior
- Impact Documents
  - Environmental
  - Traffic
  - ………
- Detail Construction Drawings
  - HVAC Drawings
  - Electrical Drawings
  - Structural Drawings
- Standards and Inspections
  - List of Required Inspections
  - Procedures for Performance
  - Results and Approvals

Example Documentation

New Consumer Product

- Concept
  - Sketches
  - Textual Description
- Product Requirements
- Product Block Diagram
- Product Functional Diagram
- Interface Drawings or Diagrams
- Physical Layout Diagram
- List of Design and Build Standards
- List of Existing and New Development Components
- List and Descriptions of Tests
- ………………

Example Documentation
Automobile

- Summary Description
- Artist Drawing
- Auto Cutaway Diagram
- Diagrams of Key Features
- System Functional Diagram
- Dash Drawings or Diagrams
- Interior Physical Layout Drawings
- List of Existing and New Development Components
- …………..

Generic Consolidated System Documentation Set

1. System Summary Textual Description
2. System Drawing or Picture
3. Operational Diagrams
4. Block Diagrams
5. Functional Flow Diagrams
6. System Functional Hierarchy Diagrams
7. System Physical Hierarchy Diagrams
8. State or Mode Diagrams
9. Data Relationships Diagrams
10. Physical Layout Diagrams
11. System User Interface
12. Lists of Hardware and Software Components
13. List of Existing vs. New Development vs. Modified Components
14. List of Design and Build Standards
15. Training and Maintenance Documentation

Every Effort Will Have A Unique Set of Documents

System Design and Documentation

What To Do

Reference

You Are A Client

- Know what system design documents are needed
- Have the system design presented and explained

You Are Management

- Know what system design is needed and what documents are needed
- Have the system design presented and explained
- Question what design alternatives were considered and why the architecture was chosen

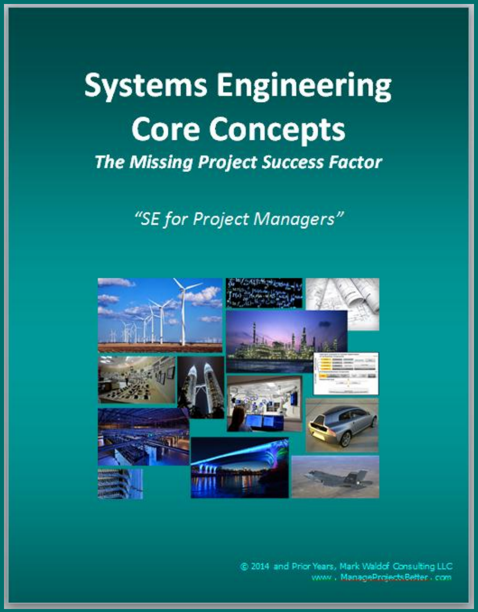
You Are An SE, Project Manager or Technical Lead

- Understand the importance of a sound system design
- Lead the team to complete a competent system design
- Make sure all team members understand the system design
- Ensure all “views” of the system are documented

You Are A Team Member

- Understand the basics of system design
- Voice concern if needed parts of the system design are missing or not compatible with other system components

End of
Systems Engineering
Core Concepts



Abbreviated Slide Set.

The current version of this seminar can be found at the URL below.

Mark Waldof Consulting LLC
www.ManageProjectsBetter.com
owner@manageprojectsbetter.com