Requirements Management

John Hrastar

NASA Project Management Conference
March 30-31, 2004
University of Maryland Conference Center

Introduction

Three aspects of requirements management

- Requirements in the beginning
  - *What are they?*
  - *How are they derived?*
    -
    -
- Requirements in the middle
  - *How are they maintained*
  - *Can they be changed?*
    -
    -
- Requirements in the end
  - *Verification*
  - *Validation*
    -
    -

Introduction

- Dictionary definitions
  - *"… a thing demanded or obligatory…"*
  - *"… a need or necessity"*
  - *"… some quality or performance demanded"*

Strong words

Introduction

- Requirements are the single thread that goes through a project from conception through build, test and flight
  - *Whole project is constructed so you can meet the requirements*
- Based on the need to measure a physical phenomena high level requirements are envisioned for a system to meet the need.
  - *Same quality or performance is demanded to be able to make the necessary measurements*
- Requirements are then refined, expanded, and flowed down to lower levels through an iterative process
- They are decomposed to the lowest levels where one person is responsible for that system of interest.

Introduction

Requirements run through the entire Project cycle.

Introduction

"Project requirements start with what the user really needs (not what the provider perceives that the user needs) and end when those needs are satisfied"

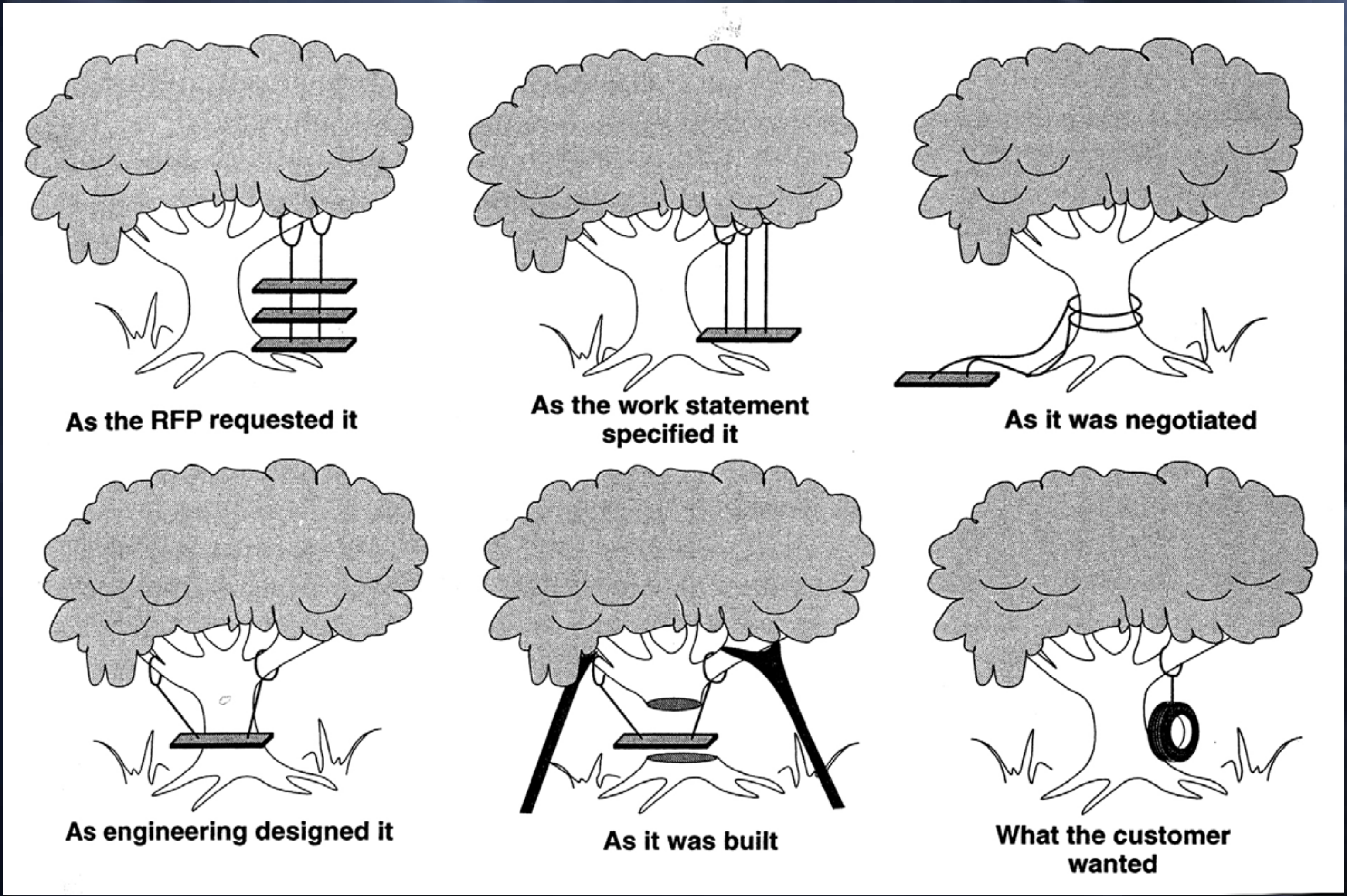
"Visualizing Project Management"
Forsberg, Moog, Cotterman

Customers, Users, Stakeholders, Developers

- User: Anyone who will work with the system. Usually the scientists (PI's) looking for the measurements.
- Customer: Person or entity you are responding to. This includes the users, Program Office, Enterprise
- Stakeholder: Anyone affected by the system including users, customers, developers
- Developers: Team that develops the system for the users

It is critical that all top level requirements are well iterated between users, customers, stakeholders, developers

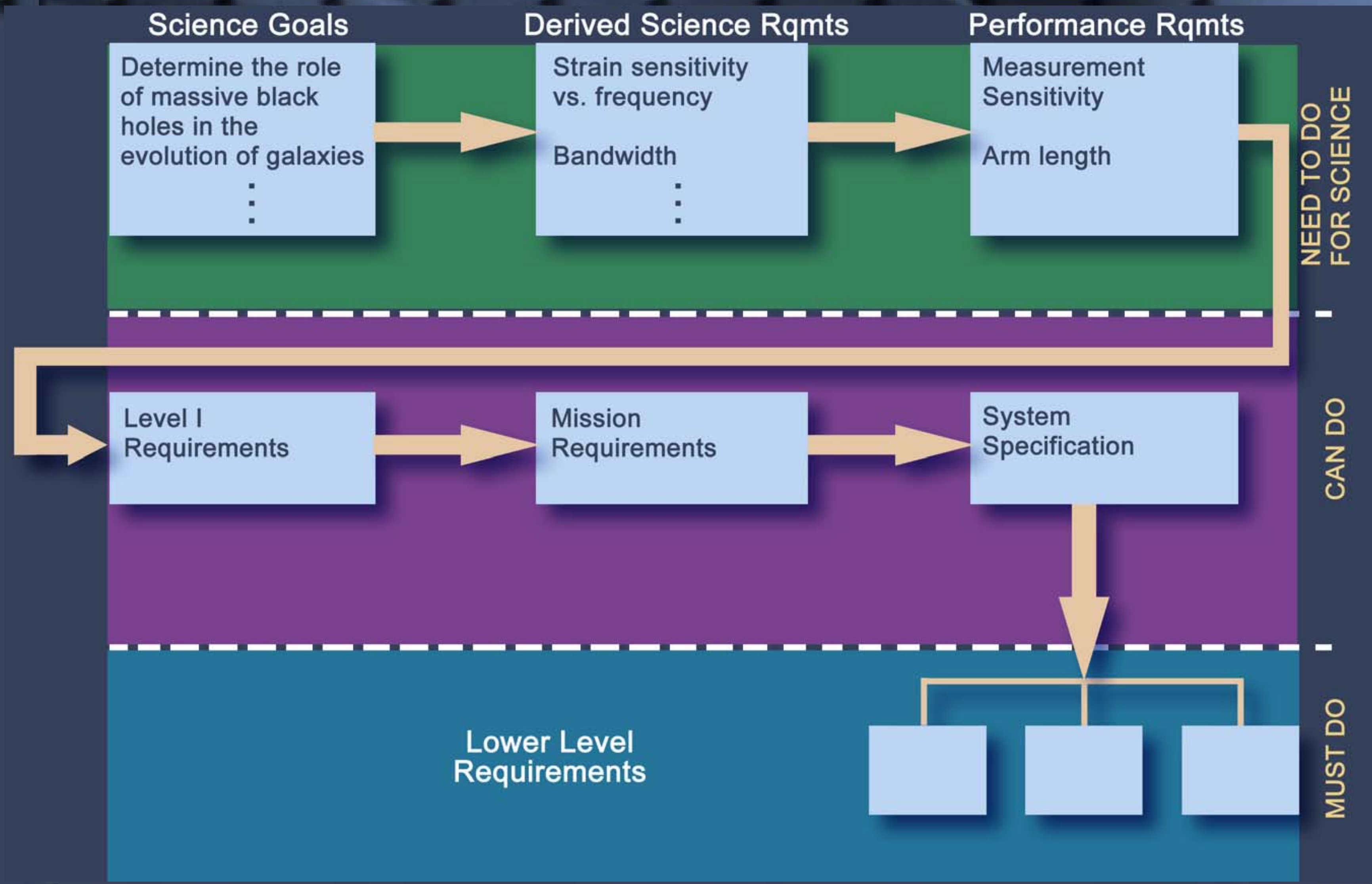
Failure to Satisfy Customers Needs



Requirements Process

- Iterative process between stakeholders, users, customers, and developers at the beginning
  - *What needs to be done?*
    - *Developers must understand user needs*
  - *What can be done?*
    - *User must understand what can be developed*
  - *What new technologies are required to achieve feasibility?*
- Maintain the interaction among these groups throughout development
  - *Re-evaluate needs*
  - *Clarify needs*
  - *Change requirements if necessary*
- Must separate ‘needs” and “wants” during concept selection
  - *Requirements are agreed “needs” of the user with what can be done*
  - *Additional “wants” (like to have) are over-specification which should be deleted*
  - *Challenging project might require technology development to high TRLs before “buildable” requirements can be met*

Requirements Process



Requirements Development

- What are the top level requirements?
  - *Concept of operations document to set context*
- What must the system do?
  - *Functional requirements*
- How well must it perform?
  - *Performance requirements*
- How do we record requirements?
  - *Organized into a hierarchy that flows through to lower systems of interest*
  - *Requirements flow follows the Project Product Breakdown Structure*
  - *Levels of requirements are shown in a document tree*
  - *Level I requirements defined in the Project Plan, also include the Mission Success Criteria*
  - *Organize requirements into functional and performance*
    - *Functional – what it must do*
    - *Performance – how well it must do it*

Performance Requirements must be validated and verifiable

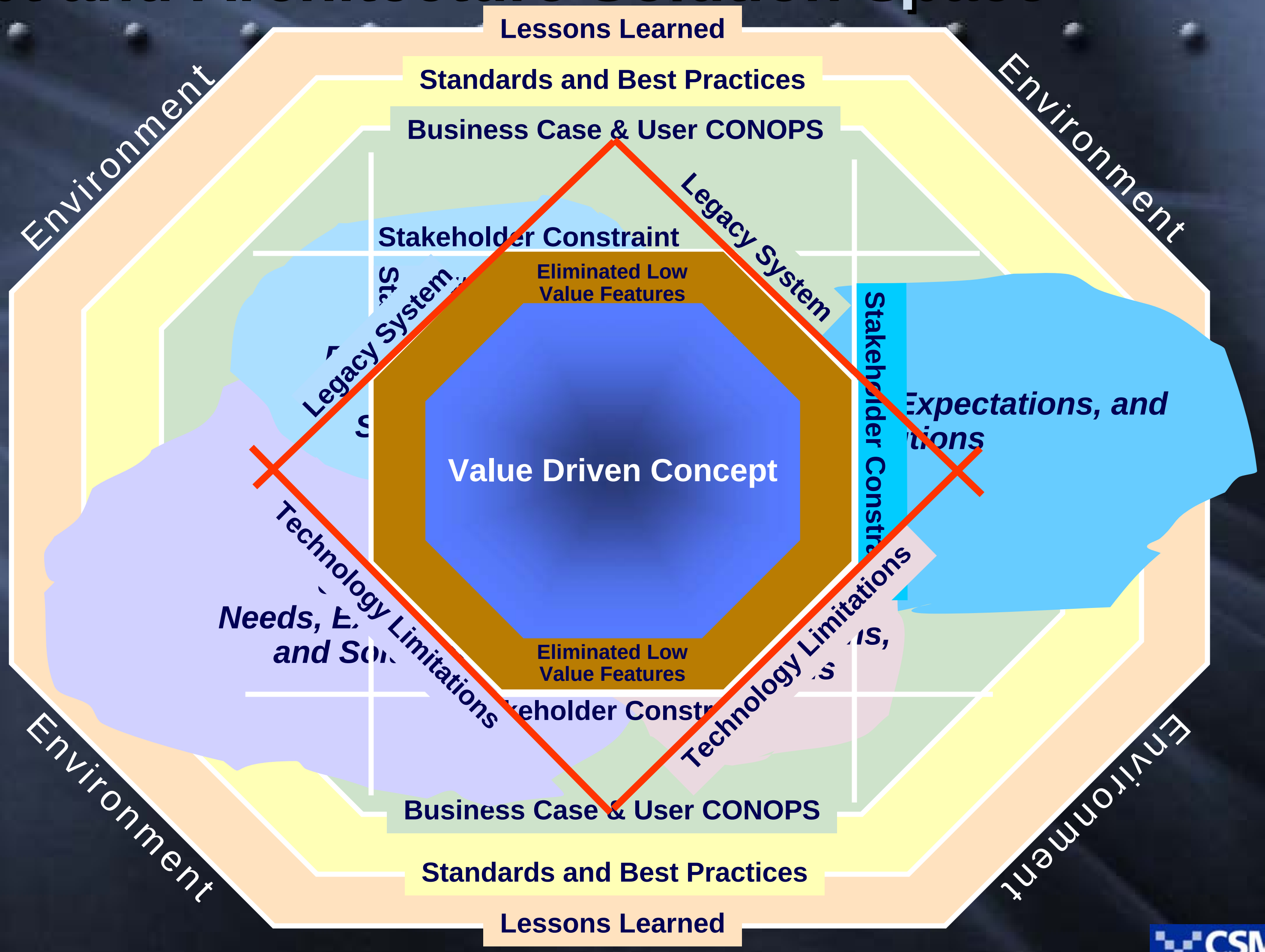
Operations Concept Development

- Puts the requirements in context
- Describes how the design can accomplish the mission described by the objectives
- Done early in the mission feasibility studies
- Trade studies to develop
- Describes system characteristics and performance from an operations perspective
- Helps better understand the capability and performance of the system within the proposed mission, use, and function
- Helps scope mission development costs, schedule, constraints
- Provides information on:

  What Who Why Where When How

- Serves as a validation reference for design throughout the life cycle

Concept and Architecture Solution Space



Mission Success Criteria

Full Mission Success Criteria: Requirements that must be met for full mission success (Cost, Schedule, Technical)

Some options available including descope, which can be exercised to implement a successful science mission but one that is less than the full mission

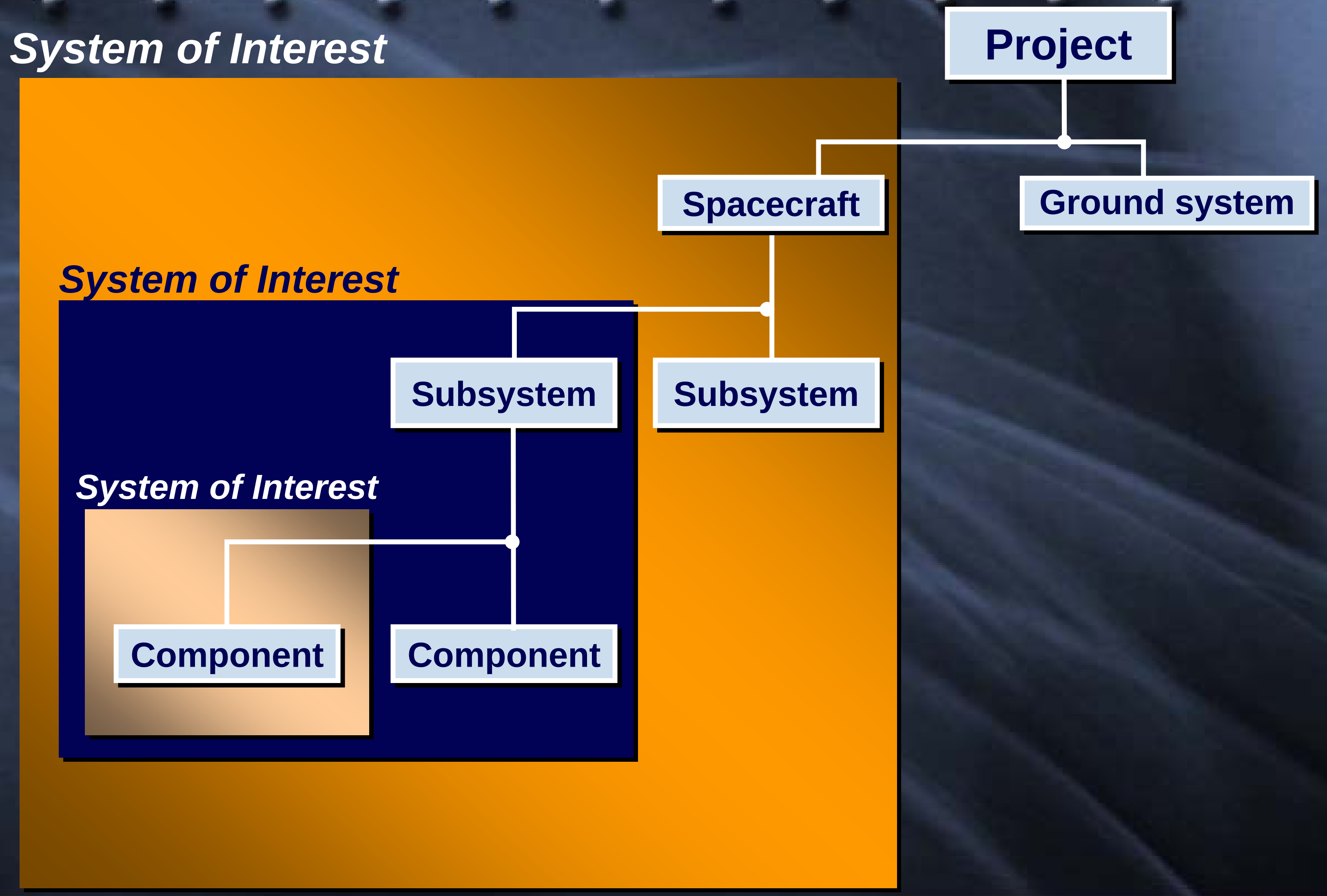
Minimum Mission Success Criteria: Requirements that must be met for minimum mission success (Cost, Schedule, Technical)

Below this level, the mission is not worth continuing

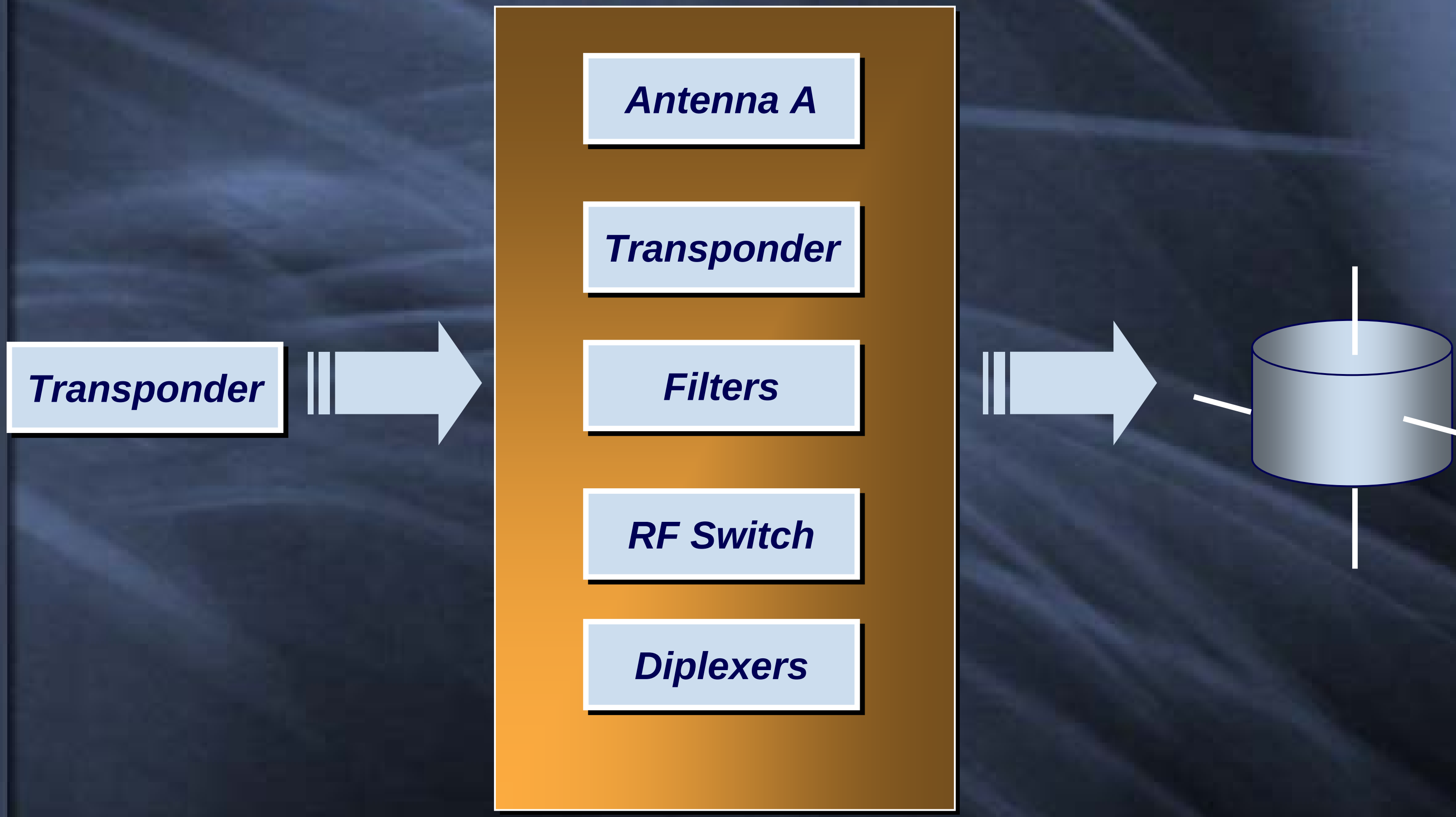
Requirements Flow: Decomposition and Integration

- Decomposition
  - *Hierarchical, functional, and physical partitioning of a system of interest into lower levels of systems of interest that can be assigned to a responsible manager*
- Fabrication and assembly of the system of interest
- Integration
  - *Successive combining and testing of hardware and software to progressively demonstrate performance and compatibility of various systems of interest*
- Verification
  - *Determination that the system meets all specified requirements*
- Validation
  - *Determination that the system satisfies what the customer (user) needs*

Hierarchical Relationships for Systems of Interest



Hierarchical Relationships for Systems of Interest

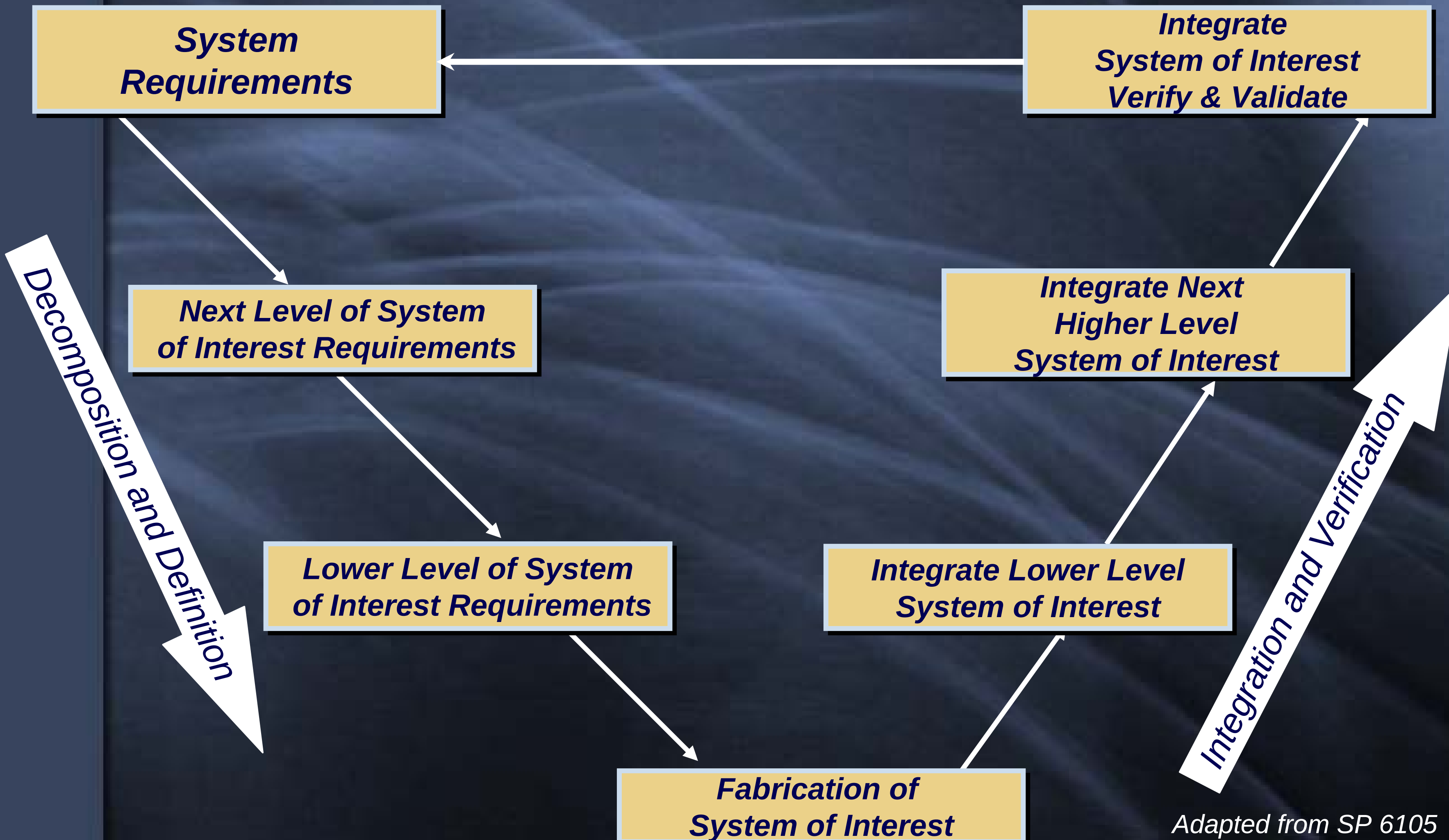


Transponder System of Interest

Communications System of Interest

Satellite System of Interest

Decomposition and Integration Cycle



Adapted from SP 6105

System Specification Allocation to Lower Leve

Table 3.1-1. Pointing and Aspect Error Allocations

Item	Degrees (3σ)	
	Aspect	Pointing
1) ACAD Pointing Error (1)	-	0.200
2) ACAD Control Dynamics (1)	-	0.200
3) ACAD Attitude Determination Error (2)	0.024	0.024
4) Spacecraft Alignment		
Bias (4)	-	0.100
Measured Uncertainty (3)	0.008	0.008
5) Spacecraft Elastic and Inelastic Deformation (4)	0.006	0.006
6) Spacecraft Structure Relaxation (4)		
Bias	-	0.012
Uncertainty	0.003	0.003
7) Spacecraft Thermal Deformation (4)	0.015	0.015
8) Instrument Alignment Uncertainty (5) (due to measurement errors and elastic, inelastic, and thermal deformation)	0.0117	0.0117
RSS Subtotal (6)	0.032	0.397 (7)
Reserve	0.008	0.264
RSS Total (6)	0.033	0.500 (7)

(1) Between attitude determination reference established by star

Managing Requirements During Development

- Requirements are not always static during development
  - *They can change for many different reasons*
- Legitimate new requirements might be added
  - *System design must be reviewed to assess the impact*
  - *New resources must be added; e.g., budget, weight, schedule, etc.*
- Requirements can “creep” if one is not vigilant
  - *Addition of a capability that is highly desirable and seems to be “free”*
  - *It is not free*
- Contingency funds are necessary to correct problems in the development process to satisfy needed requirements
  - *They are not to be used to accommodate requirements creep*
  - *This is the only discretionary money a Project Manager has*

Managing Requirements During Development

- Examples
  - *STEREO – KSC clean room requirements*
  - *GRO – Level I requirement on fuel load*
  - *TDRSS – Major changes on a fixed-price contract*
  - *EOSDIS – Missed recognition of technology changes that could have caused requirements changes*
  - *TIMED – Missed recognition of changing environment that eventually led to requirements changes*

Validation and Verification

- “The purpose of verification is to ensure that the subsystems conform to what was designed and interface with each other as expected in all respects…”
- “Validation consists of ensuring that the interfaced subsystems achieve their intended results.”
- “While validation is even more important than verification, it is usually much more difficult to accomplish” (Very clear example later.)

Verification: “Is the system built right?”
Validation: “Has the right system been built?”

NASA Systems Engineering Handbook

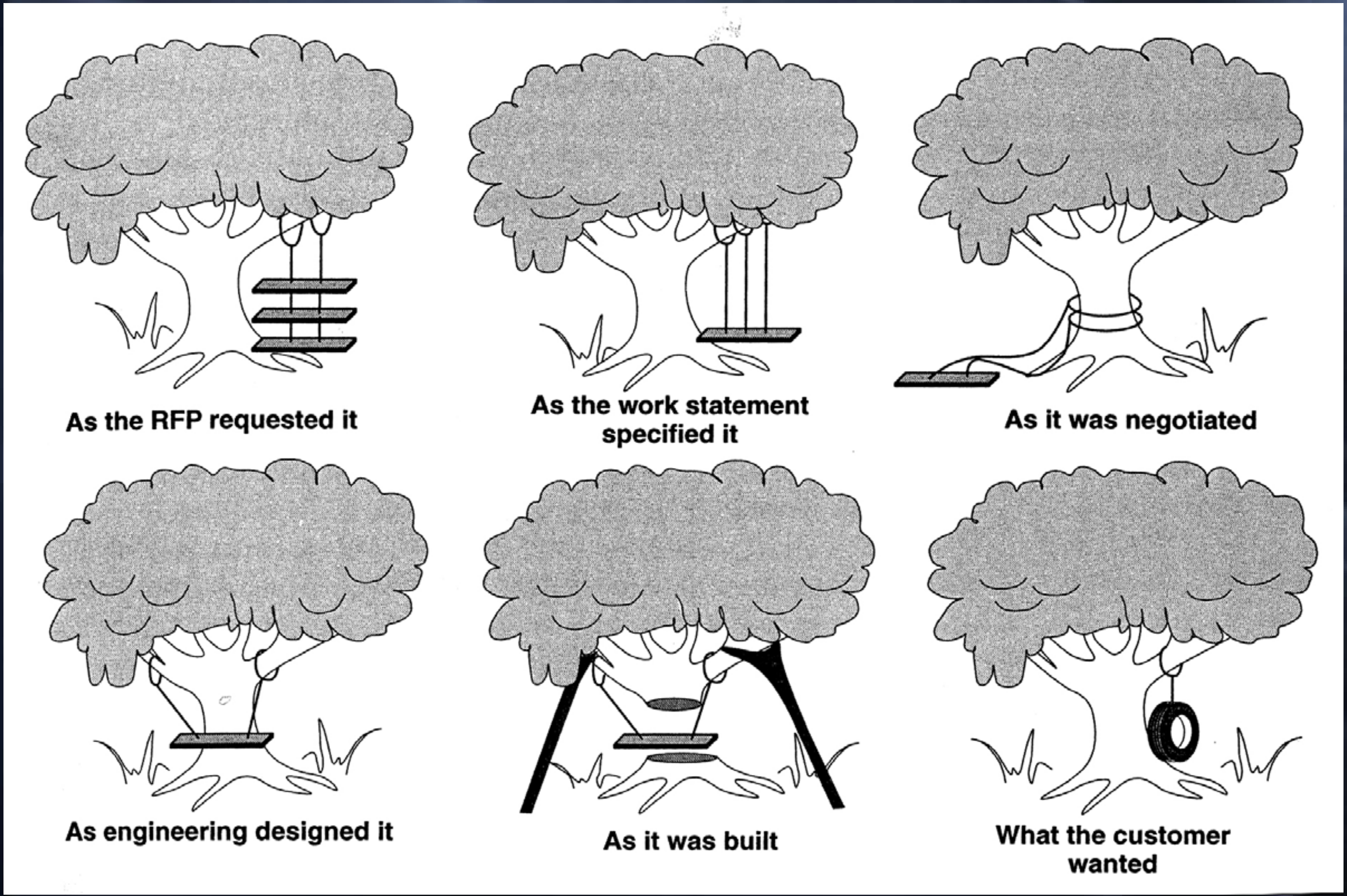
Validation

- Validation assures the design will meet mission objectives
  - *"Will the customer smile?"*
  - *Is the right system being built?*
- Validation begins at the start of the project cycle
  - *Validation plan set when the user requirements baseline is set*
  - *Confirmation at the control gates*
- Validation is a formal continuous confirmation that the product will meet the users needs.
  - *Requirements vs. needs*
  - *Specification vs. needs*
  - *Design vs. needs*
  - *Product vs. needs*
- Validation methods
  - *Focus on operational scenarios and how they are supported, I.e. the operations concept*
  - *Validate against architecture and design*

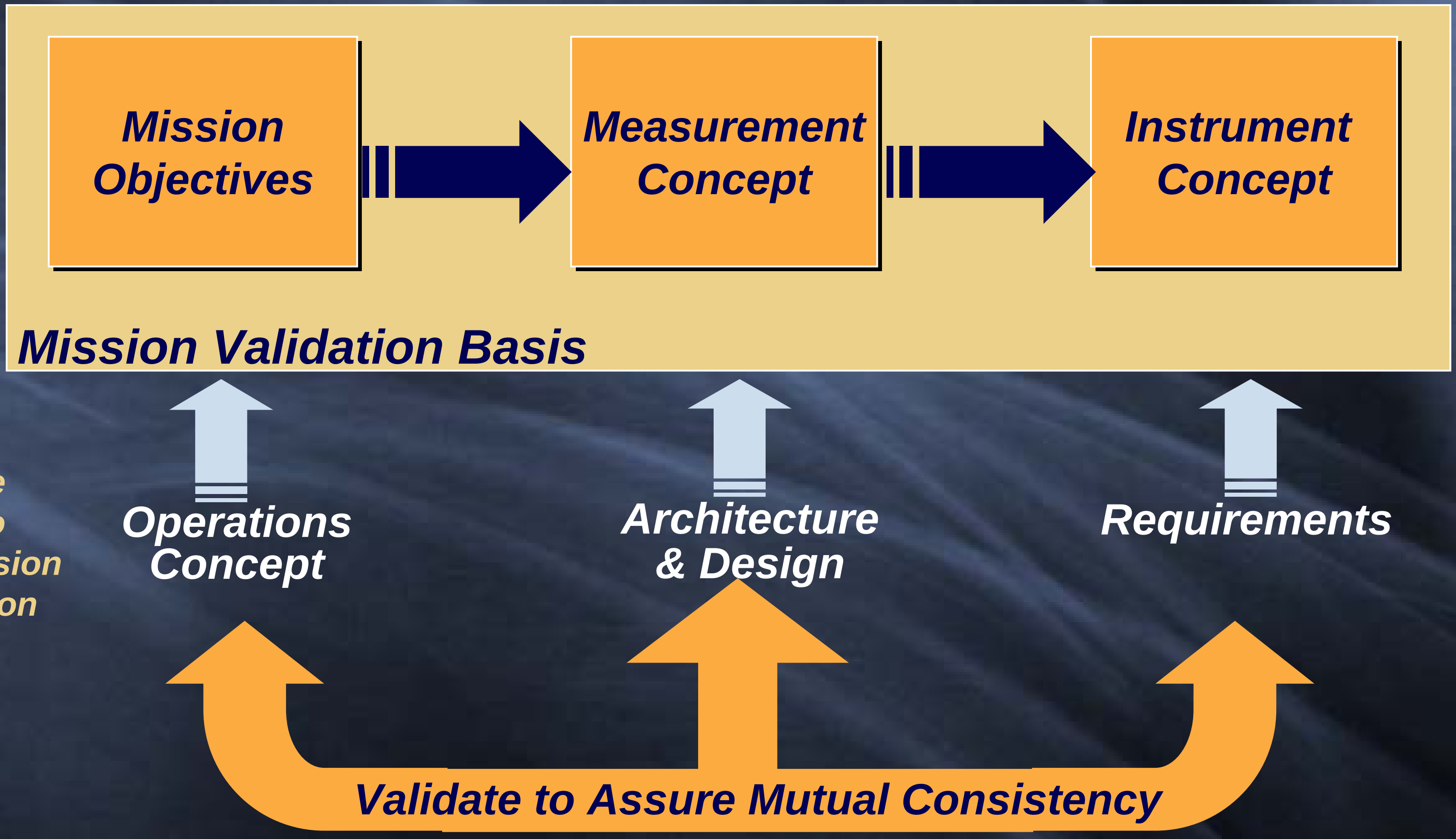
Validation

- Stakeholders interaction is critical
  - *Keep going back to stakeholders to validate what you are developing is what he/she wants*
- Verification program is validated to requirements
  - *Assure all requirements are verified*
  - *Assure the traceability of the parent and child requirements*
- End-to-end testing is the ultimate test for both verification and validation

Failure to Validate



Mission Validation Basis



Validate these to the Mission Validation Basis

Verification

- Make sure the team builds the system right
- Verify design and implementation against the requirements
  - *Proof of compliance with the specification*
- Verification process identifies the verification item, the method (analysis, inspection, test) and review of the verification results
- “Test as you fly and fly as you test
  - *Need to identify anything not tested in flight configuration and ascertain and mitigate risk*
- Test planning should include environment exposure as well as requirements for comprehensive, functional, aliveness tests, etc.
- End-to-end testing from the science input through the science data output is the best verification and validation test

Verification

- Object: Ensure all functional, performance and design requirements (from Level I through Level n) have been met
- Begins in Phase A, increases in Phase B with the refinement of requirements, cost, schedule. More detailed plans in Phase C (design). Phase D (development) includes the qualification and acceptance verification
- Verification methods and techniques:
  - *Test – Measured compliance with metrics*
  - *Analysis – Predicted compliance with history*
  - *Demonstration – Observed compliance without metrics*
  - *Inspection – Compliance with drawings, documentation*

NASA Systems Engineering Handbook

Requirements Verification Matrix

<u>Verification Types</u>

D - Development
Q - Qualification
A - Acceptance
J - Joint qual/accept
N/A - Not Applicable

<u>Verification Method</u>

1. Test
   a. Functional
   b. Environmental
2. Assessment
   a. Similarity
   b. Analysis
   c. Inspection
   d. Demonstration
   e. Validation of records

Performance Design Requirement (Include Spec. Reference Paragraph)		Verification Levels					Comments
				Assembled Levels			
		Comp.	Subsystem	End Item	Integ.	Post-Flight	
3.2.1.1.4.2.3	Velocity Control Pointing			1a/2b			Flight simulation test
3.2.1.1.4.2.4	Velocity Control Thruster Timing			1a/2b			Flight simulation test
3.2.1.1.4.2.5	Thruster Maneuver Response Time			1a/2b			Flight simulation test
3.2.1.1.4.2.6	Safe Hold Pointing			1a/2b			Flight simulation test
3.2.1.1.4.2.7	Safe Hold Response Time			1a/2b			Flight simulation test
3.2.1.1.4.2.8	Sun-Referenced Pointing			1a/2b			Flight simulation test
3.2.1.1.4.2.9	Sun-Referenced Pointing Response Time			1a/2b			Flight simulation test
3.2.1.1.4.2.10	Backup Orbit Maintenance			2b			
3.2.1.1.4.2.11	High Gain Antenna Pointing		2b	J1a/2d			HGA drive travel by test
3.2.1.1.4.2.12	Solar Array Pointing			J1a			
3.2.1.1.4.2.12.1	Solar Array Index Position			2b			

Verification by Test

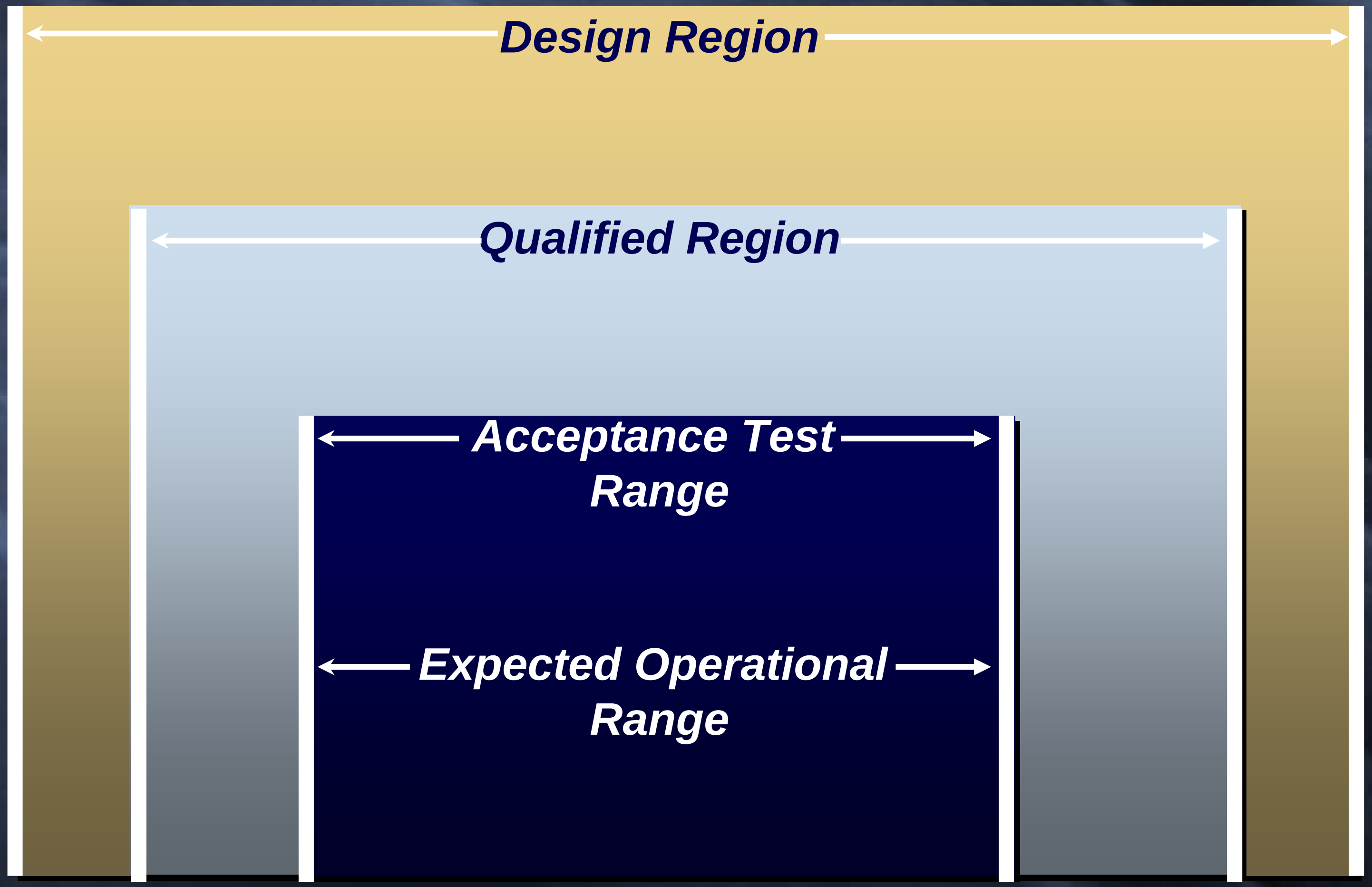
- Actual operation of equipment in ambient conditions or when subject to specified environments
- Functional testing – series of tests (elec./mech.) conducted on the hardware and/or software at conditions less than or equal to the design specification
  - *Comprehensive functional test*
  - *Does it perform satisfactorily?*
  - *Before and after each environmental test?*
- Environmental testing: series of tests to assure it will perform in the flight environment
  - *Vibration*
  - *Acoustic*
  - *Thermal vac*
  - *Etc.*

Verification Stages

- Development: Formulated and implemented up to the manufacturing of qualification or flight hardware
  - *E.g. Breadboard testing*
- Qualification Stage: Flight (protoflight) or flight-type hardware is verified to meet functional, performance, and design requirements
  - *More severe than acceptance conditions to establish the hardware will perform in flight with sufficient margin*
- Acceptance: Deliverable flight end item is show to meet functional, performance, and design requirements under flight conditions

NASA Systems Engineering Handbook

Systems Environment and Verification Philosophy



Summary

- Getting requirements right at the beginning is critical because they run through the whole program
  - *It is what you are putting all your effort into satisfying*
  - *Iteration with the stakeholders is critical*
- As you proceed through the program, they must be validated regularly with the stakeholders
  - *Control must be maintained through a configuration management process*
  - *Don't close your eyes to necessary changes*
- At the end, they must be verified and validated to assure mission objectives will be met
  - *"Is the system built right?"*
  - *"Has the right system been built?"*
  - *"Is the customer smiling?"*

Validation Example

The whole effort of the Project is directed toward satisfying the requirements. If done right, the Project will be successful!

Back-up

Requirements Management Process

- Derive requirements consistent with the Project Plan regarding technical content, cost, schedule, security and institutional requirements
- Perform project system engineering analysis to ensure cost effective requirements are specified
- Collect and allocate project requirements into implementation elements
- Document and maintain under configuration control project requirements, requirements verification, and end-item spec.

Note that most requirements will be derived from higher level requirements

Requirements Accountability

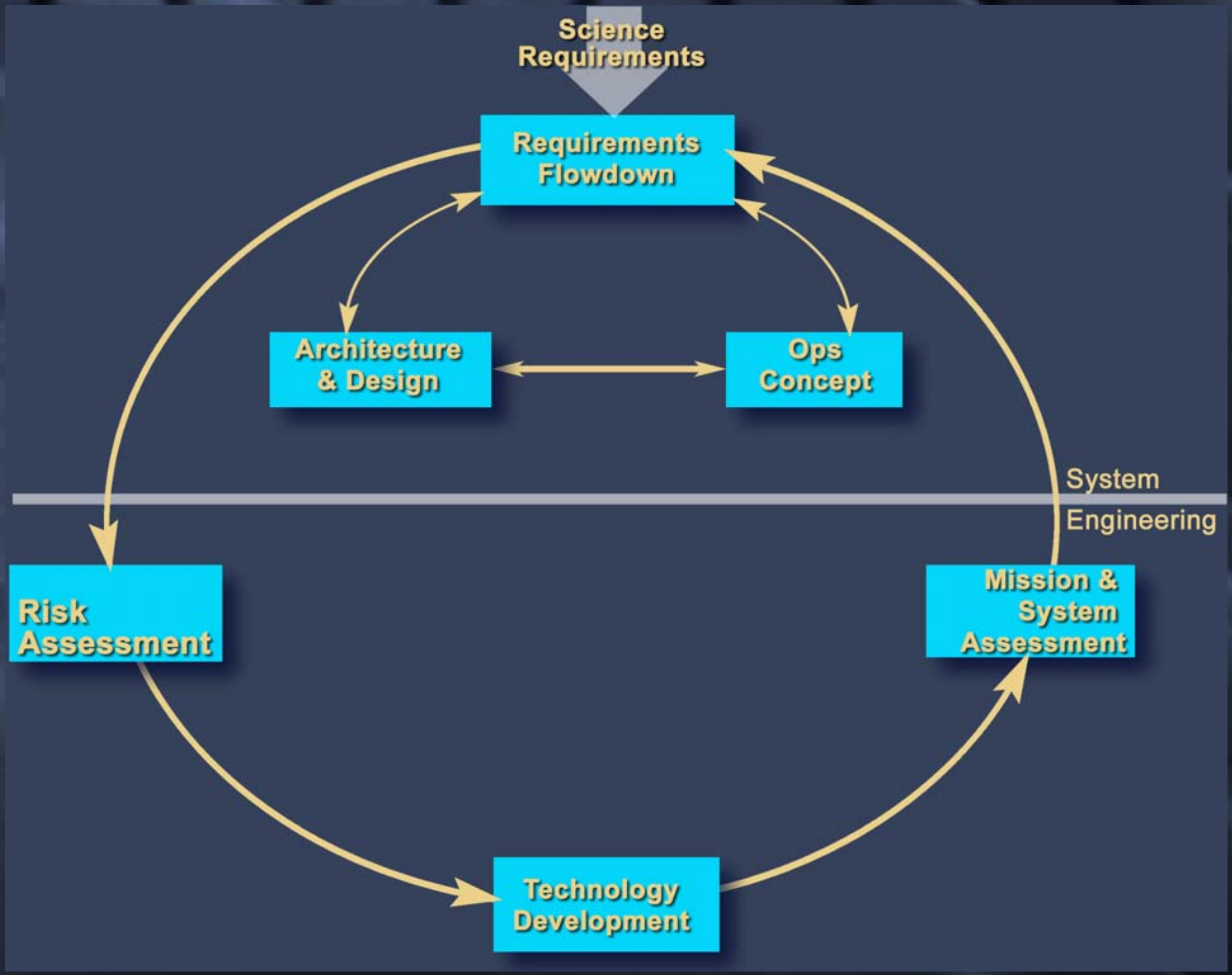
- The purpose of requirements accountability is to ensure :
  - *That all requirements have been responded to, and;*
  - *Have been verified by test, inspection, demonstration and analysis*
- Systems Engineering is responsible for auditing the verification results and certifying that the evidence demonstrates requirements have been achieved.
- The accountability extends from the beginning of the project to the end

"Visualizing Project Management"
Forsberg, Moog, Cotterman

Requirements Levels

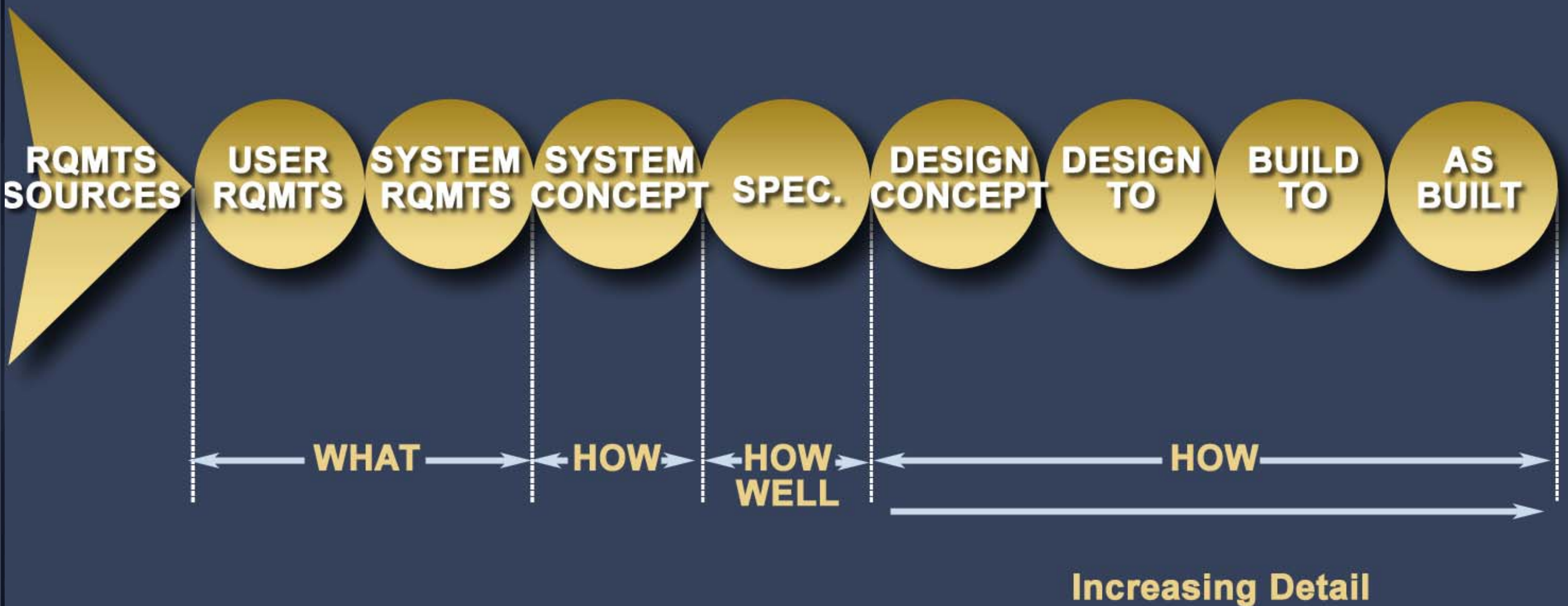
- Science Requirements
  - *Based on science goals, e.g., determine the role of massive black holes in galaxy evolution*
  - *Stated in terms of various parameters*
  - *Flow to science instrument requirements in terms of measuring these parameters*
- Level I requirements
  - *Sets top level system requirements based on the science instrument requirements*
  - *Brief document, often in the Project Plan, controlled by the Enterprise*
  - *Sets derived mission level requirements*
- System specification
  - *Defines what it must do*
  - *Defines how well it must do it*

Requirements Process



Technology as a major component affecting requirements

Requirements Development



Requirements and Document Chain

"Visualizing Project Management"
Forsberg, Moog, Cotterman

Control Gates

- All Control Gates must answer two questions at each level of decomposition
  - *Are we building the right solution?*
  - *Are we building the solution right?*
- To answer these, the case for each level must be the current one flowed down with accompanying criticality, risk, cost and schedule
- Must look up one level to assure you are building the right entity.

"Visualizing Project Management"
Forsberg, Moog, Cotterman

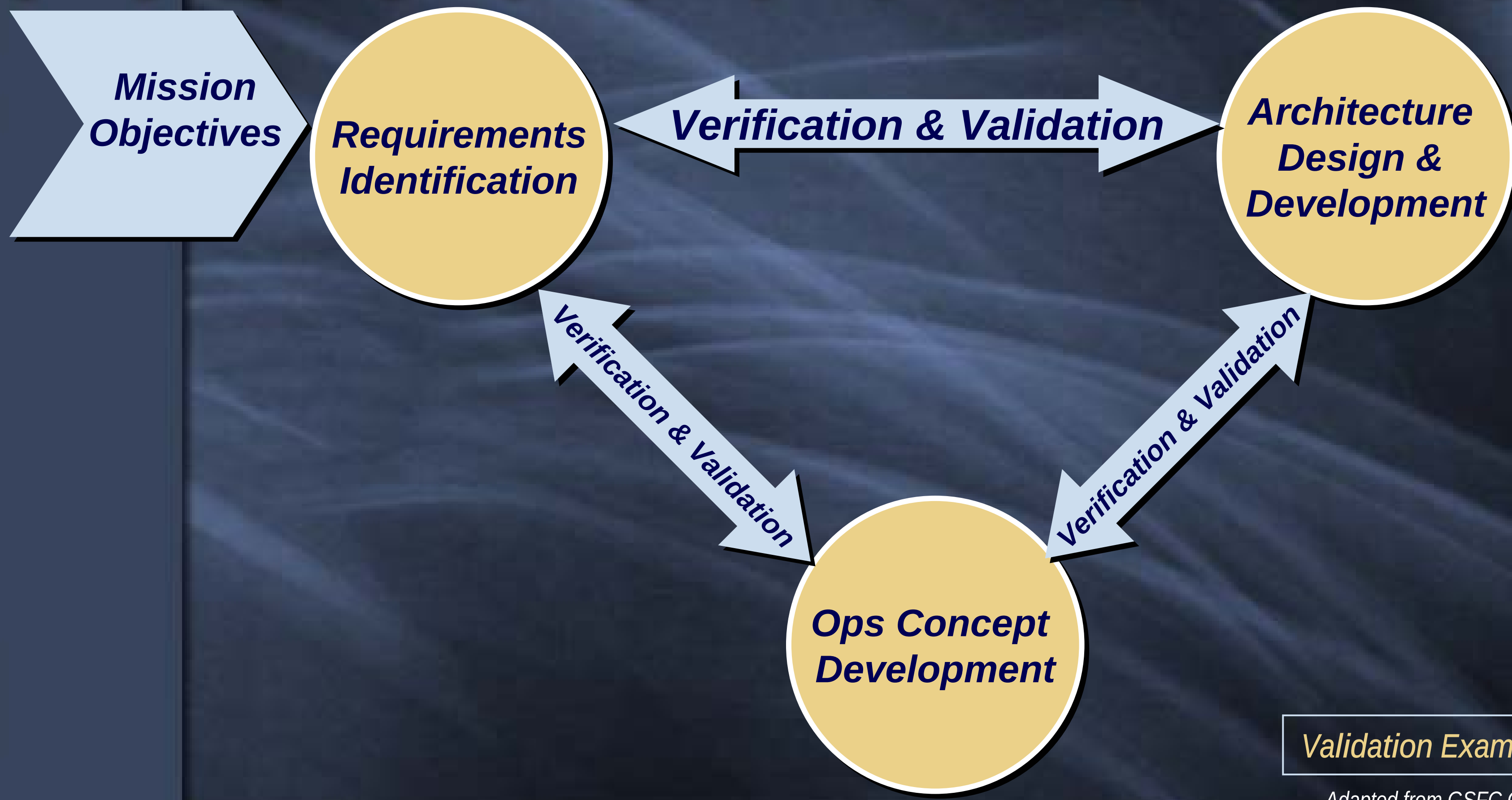
Validation and Verification

Verify – "To prove the truth of …"

Verification – "Evidence that established or confirms the accuracy or truth of …"

Validate – " … substantiate, confirm; to give official sanction, confirmation or approval to …"

Verification and Validation



Validation Example

Adapted from GSFC GPG on Systems Engineering

Mutual validation of Requirements, Architecture, and Operations