

Design System Failure Away with Systems Engineering and Decision Analysis

The Engineering Design of Systems: Models and Methods

Dennis Buede

Department of Systems Engineering and Operations Research

George Mason University

INFORMS, Philadelphia

10 November 1999

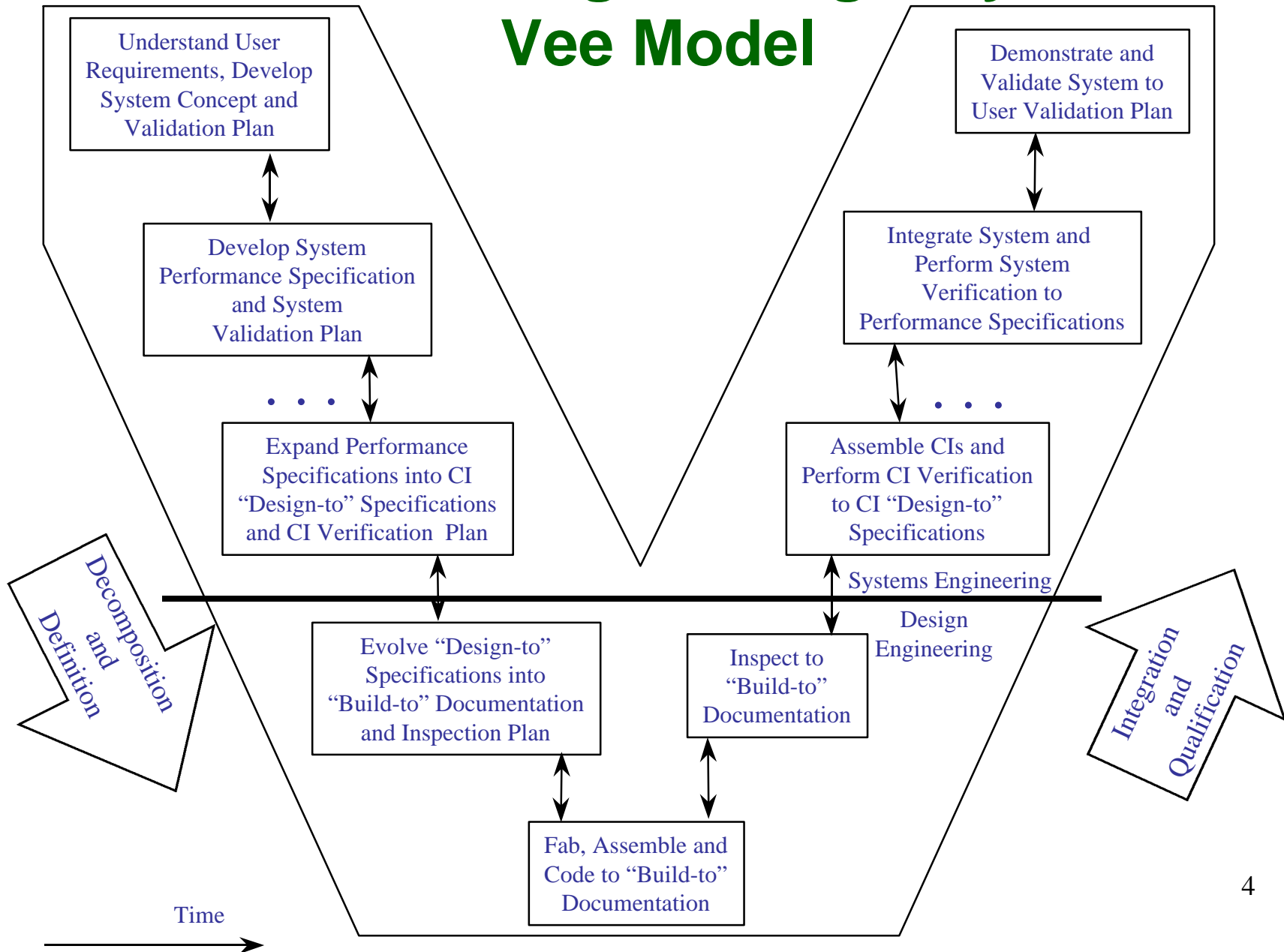
Agenda

- **Process for Engineering a System**
- **Engineering Design**
 - **Defining the Problem**
 - Defining the Trade Space
 - Defining the Value Space
 - **Solving the Problem**
 - Defining Solutions
 - Evaluating Solutions
- **Decision Analysis Issues**

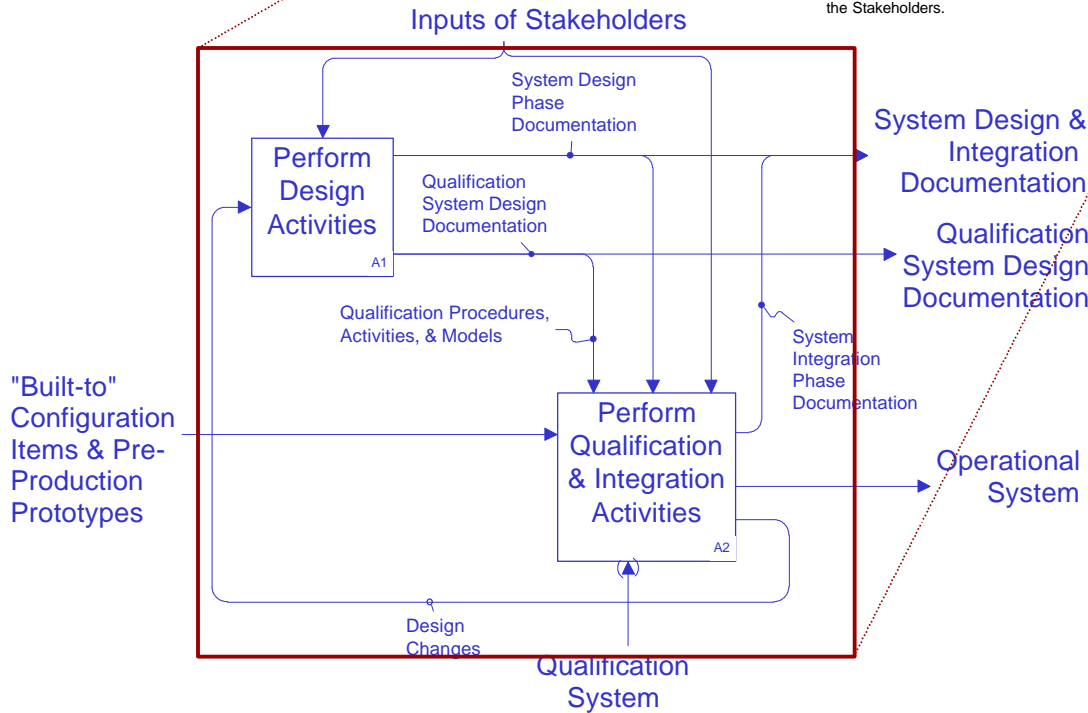
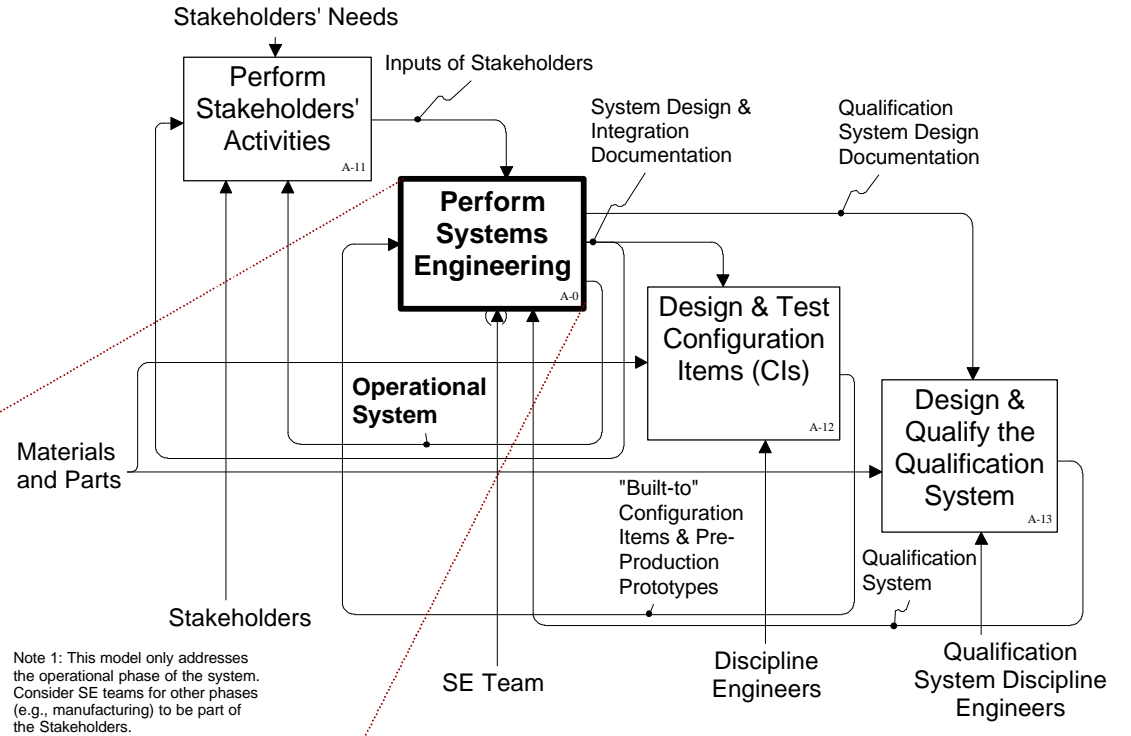
Problem Statement for Engineering Design of a System

- **Two Level Discovery Problem**
 - What is the problem?
 - What is an acceptable solution?
- **Solution takes the form of**
 - **Hierarchical set of specs**
 - Hierarchical problem definition
 - Requirements for each component
 - **Interface control documents**
 - Define component connections
 - Critical to successful solution
- **Compounded by**
 - **Discover qualification system**
 - Verification
 - Validation
 - Acceptance
 - **Concurrent systems**
 - Development
 - Manufacturing
 - Deployment
 - Training
 - Refinement
 - Retirement

Process for Engineering a System: Vee Model



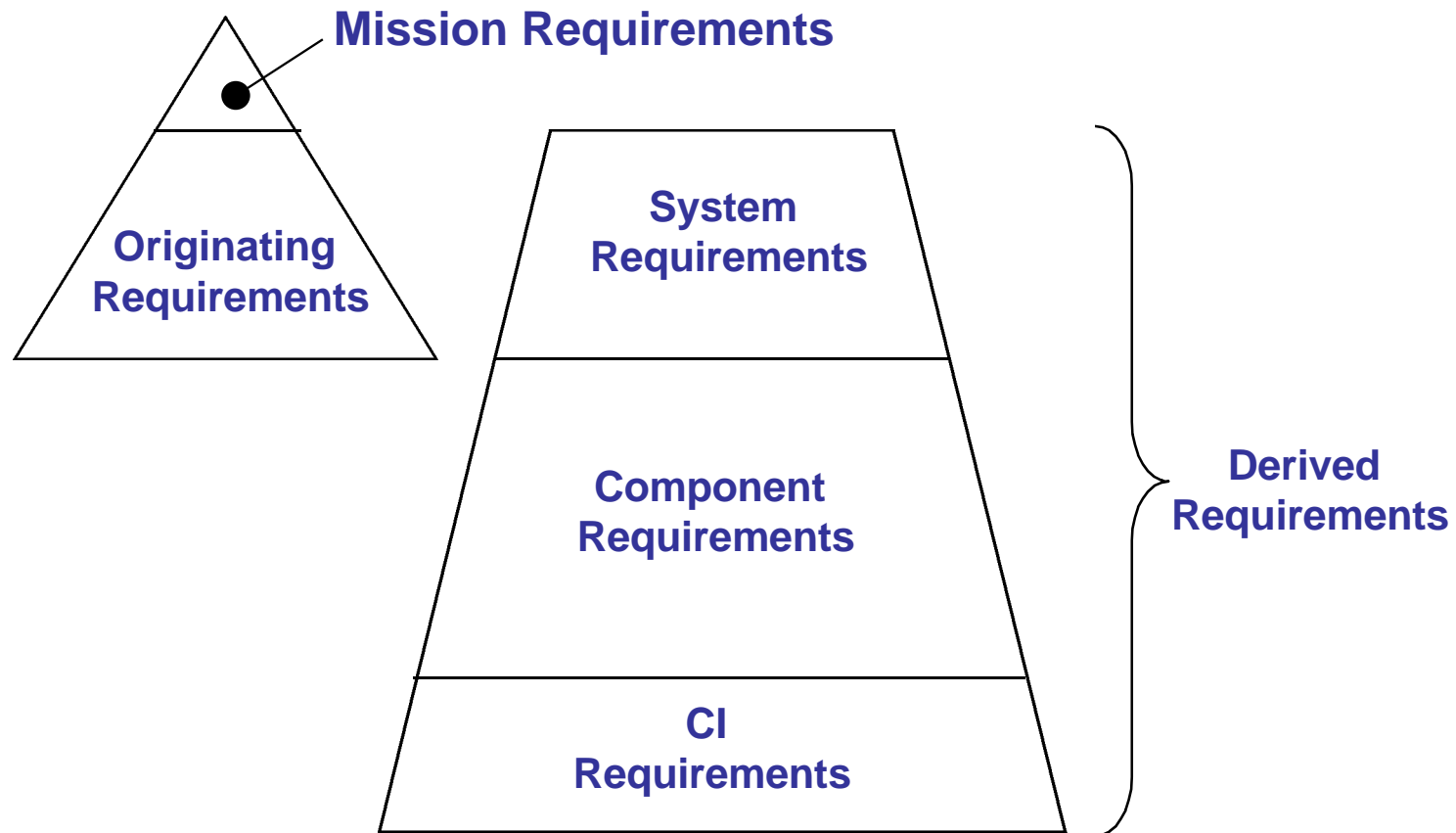
Process for Engineering a System: An IDEF0 Model



Defining the Problem

- **Top-level View of Requirements**
- **Requirements Development Process**
- **Requirements Taxonomy**
- **Integration of the Definition Process**

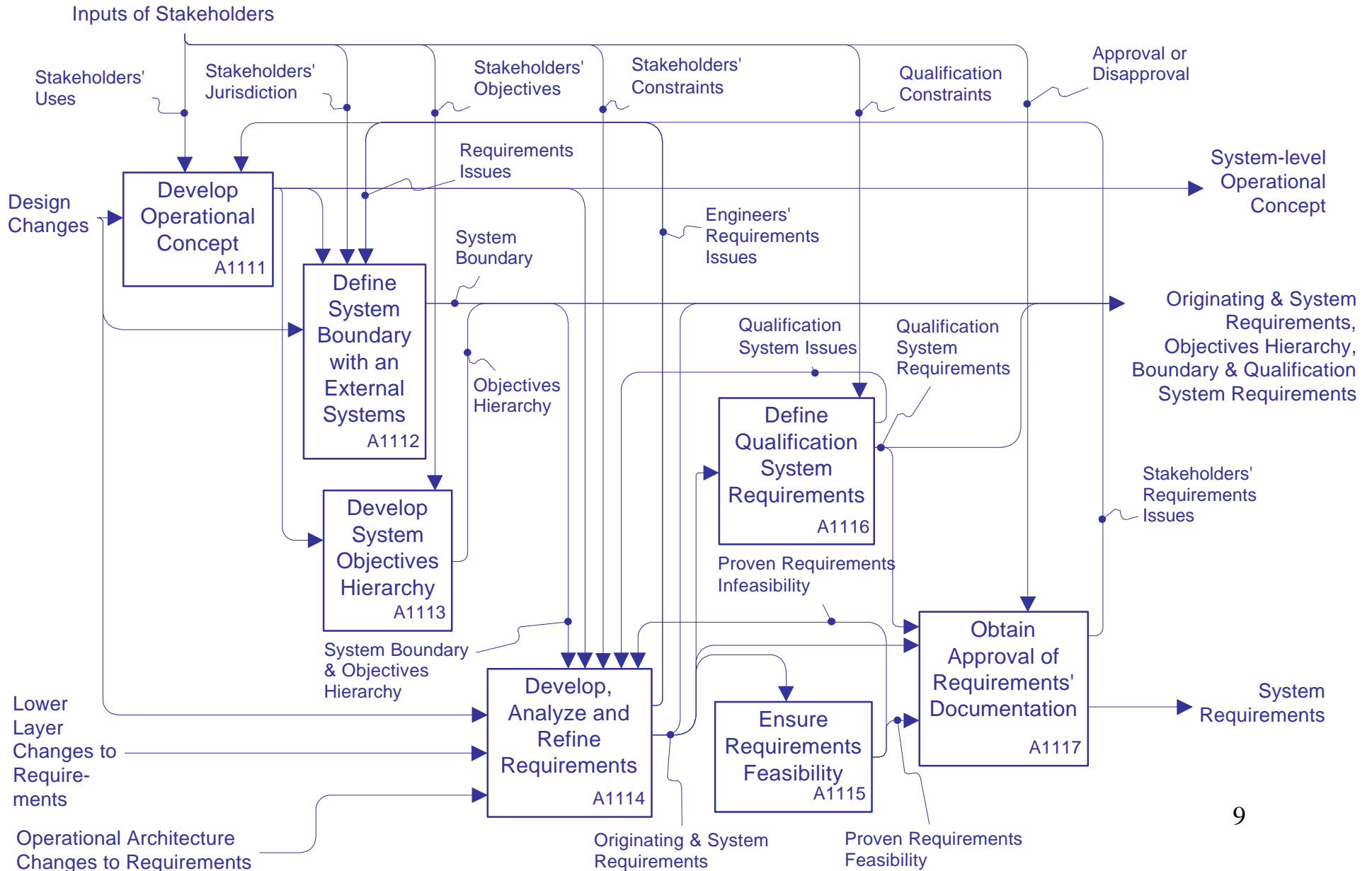
Top-level View of Requirements



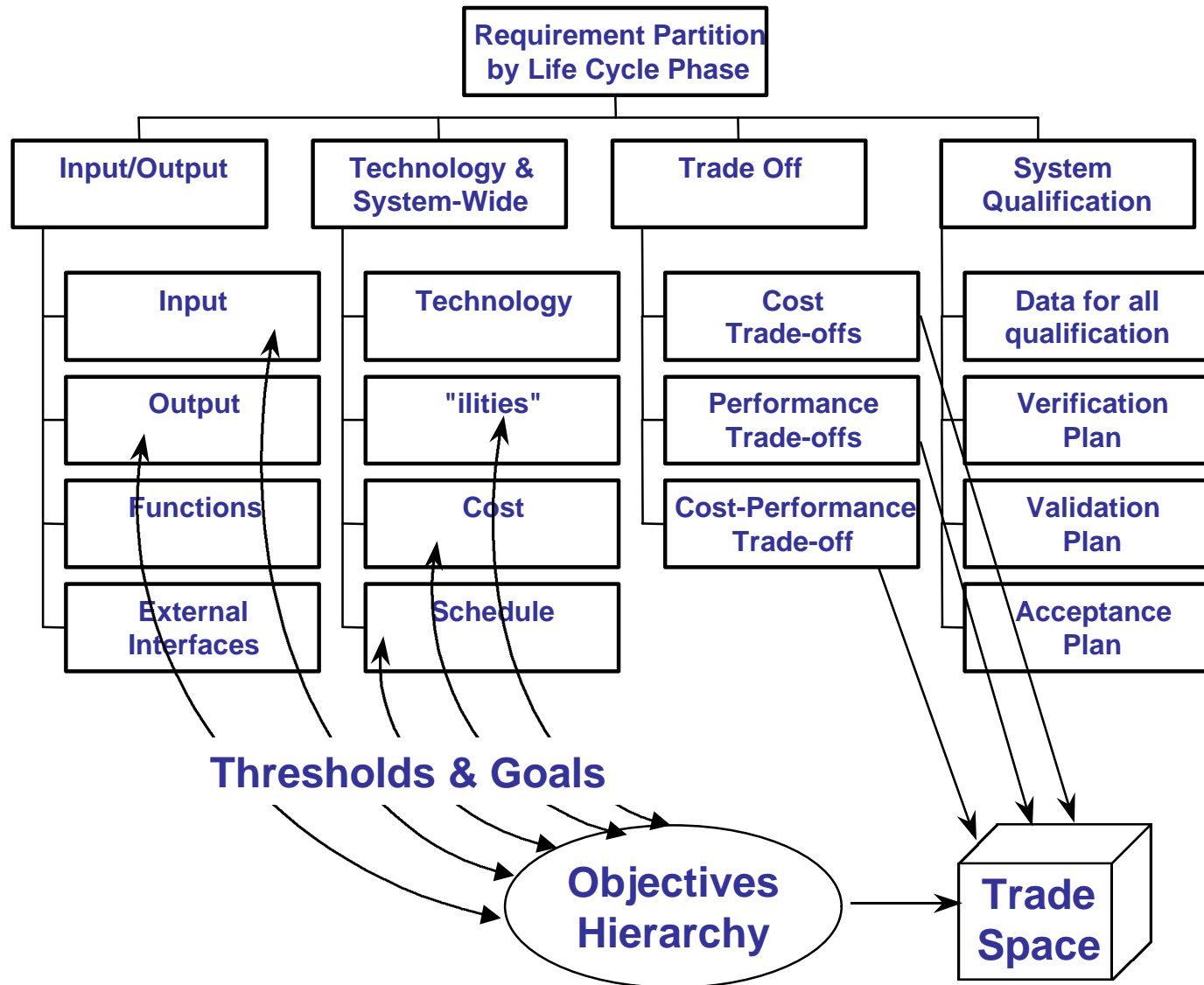
Engineering Design: Defining the Problem

- **Requirements**
 - Historical Approach in Systems Engineering
 - Many Meanings & Taxonomies
- **Proposed Approach**
 - Defines Taxonomy of Requirements
 - Addresses Entire Life Cycle
 - Separates Trade Space from Value Space
 - Provides Focus for Stakeholders' Concerns
 - Does Not Begin Solution Process
 - Addresses Qualification Decisions as well
 - Provides a Process for Developing Requirements (No Miracle Required)

Requirements Development Process: IDEF0 Model



Requirements Taxonomy

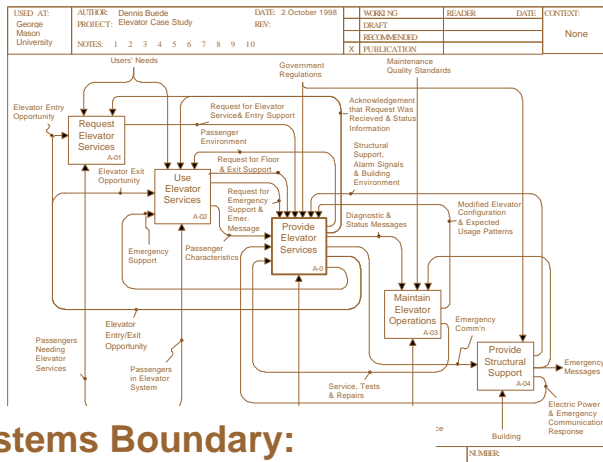
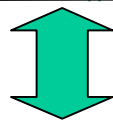
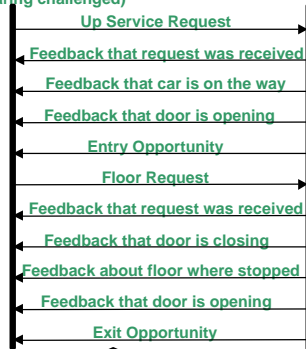


Integration of Definition Process

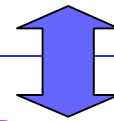
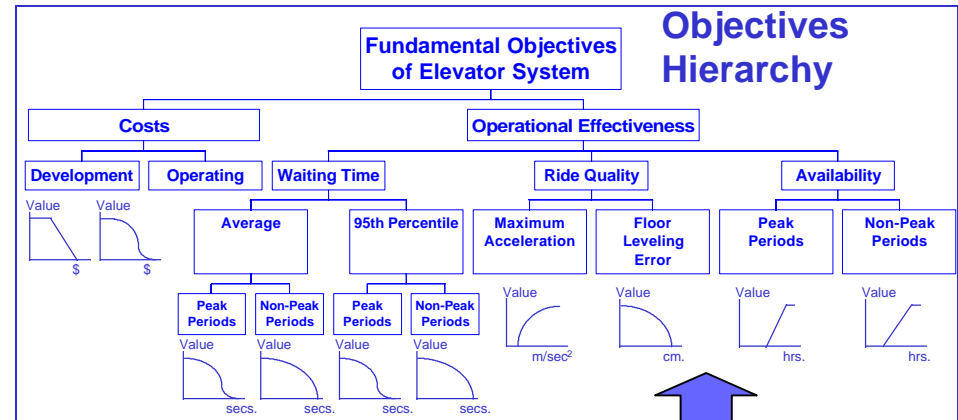
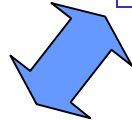
Operational Concept: Vision Scenarios

Passenger (including mobility, visually & hearing challenged)

Elevator

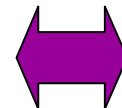


**Systems Boundary:
External Systems Diagram**



Originating Requirements Document (ORD)

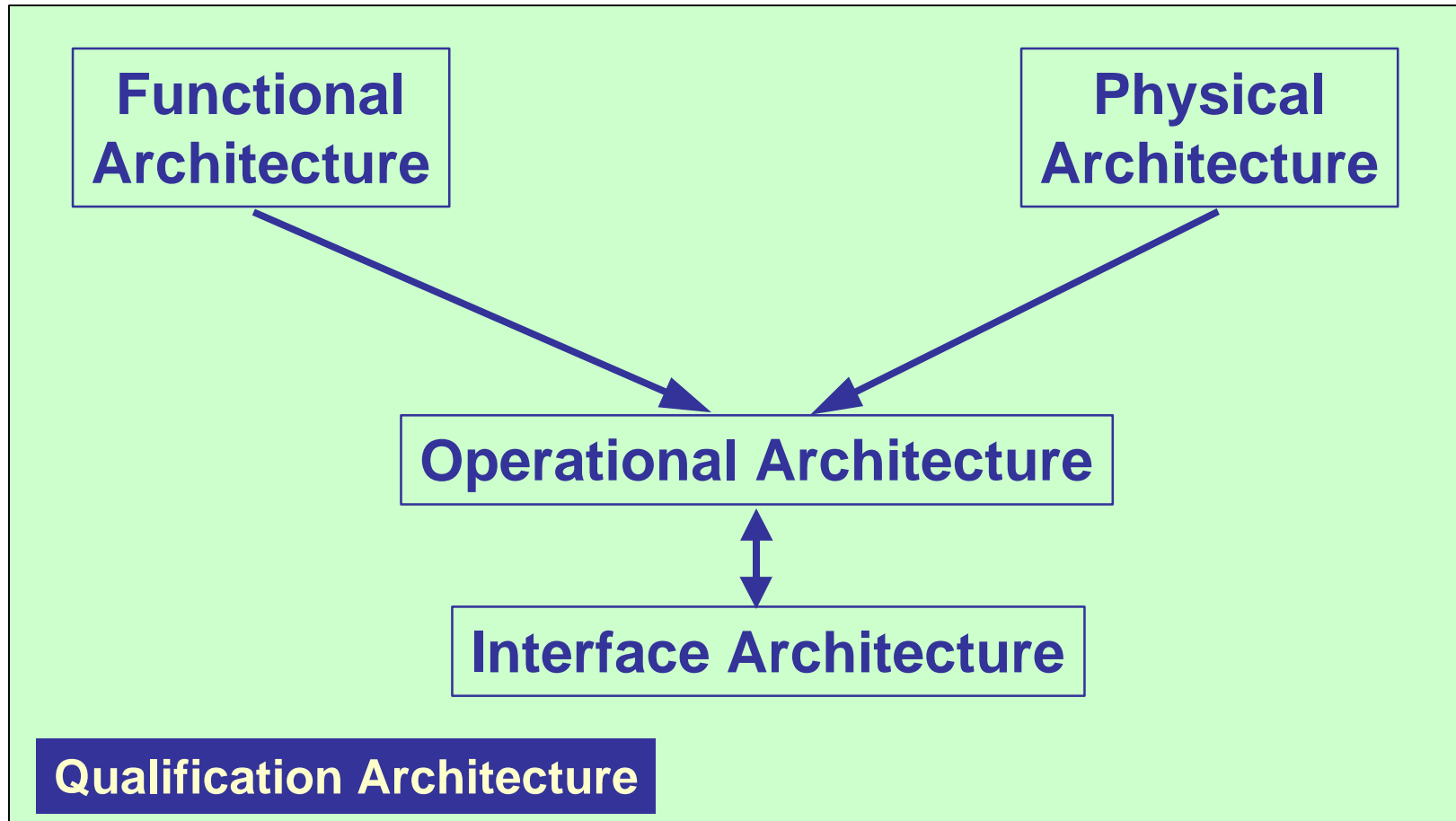
- 1.0 System Overview
- 2.0 Applicable Documents
- 3.0 Requirements
 - 3.1 Development Phase (Programmatic) Requirements
 - ...
 - 3.2 Manufacturing Phase Requirements
 - ...
 - 3.3 Deployment Phase Requirements
 - ...
 - 3.4 Training Phase (if present) Requirements
 - ...
 - 3.5 Operational Phase Requirements
 - 3.5.1 Input/Output Requirements for Operations
 - 3.5.2 System-wide/Technology Requirements for Operations
 - 3.5.3 Trade-off Requirement for Operations
 - 3.5.4 Qualification Requirement for Operations
 - 3.6 System Improvement/Upgrade Phase Requirements
 - ...
 - 3.7 Retirement Phase Requirements
 - ...
 - 3.8 Overall Trade-Off Requirement



Solving the Problem

- **Architectures for Solving the Problem**
- **Functional Architecture**
- **Physical Architecture**
- **Operational Architecture**
- **Interface Architecture**
- **Qualification Architecture**

Architectures for Solving the Design Problem



Functional Architecture

- **Decomposition of the system's top level function**
- **Model defining how the system transforms its inputs into its outputs**
- **Tracing of input/output requirements to specific functions and items**

How the System Meets Its Input/Output Requirements

Physical Architecture

- **Generic physical architecture**
 - **Set of resource buckets**
 - **Devices**
 - **Hardware**
 - **Software**
 - **Facilities**
 - **People**
- **Instantiated physical architecture - a generic physical architecture to which a complete definition of the performance characteristics of the resources have been added**

What the System Is

Operational Architecture

- Allocation of Functions to Components
- Tracing of System-wide Requirements to System
- Derivation of Component-Wide Requirements
- Definition and Analysis of Functional Activation and Control Structure
- Key Design Decisions

How the System Performs

Interface Architecture

- **Control Documents**
 - External Interfaces
 - Internal Interfaces
- **Physical and Functional Elements**

How the System Connects

Qualification System Architecture

- **Qualification Issues**
 - **Verification – “Built System Right”**
 - **Early Validation – “Right Problem”**
 - **Late Validation – “Solution Is Right”**
 - **Validation – “Built Right System”**
 - **Acceptance – “Stakeholders Agree”**
- **Qualification Methods**
 - **Instrumented Test**
 - **Analysis and Simulation**
 - **Inspection**
 - **Demonstration (Field Test)**

How the System Solution Is Proven

Decision Analysis Issues

- **Organizing Design Decisions for Concurrent & Qualification Systems**
 - **Concept Definition: Evaluation of Competing Concepts**
 - **Risk Analyses: Evaluation of Competing Operational Architectures**
 - **Performance Analyses: Understanding Performance**
 - **Trade Studies: Detailed Fine Tuning of Performance Requirements**
- **Deriving Trade-off Requirements**
 - **Deriving Objectives Hierarchies**
 - **Deriving Thresholds and Design Goals and Associated Weights**
 - **Deriving Value Curves**

Decisions

- What are decisions?
- What does it mean to make a good decision?
- Elements of all decisions
 - Values
 - Alternatives (and Options)
 - Facts and Experiences (and Interpretations thereof)

Engineering Design Process Acknowledges Key Elements

Organizing Design Decisions

- **Guiding Principles**
 - **Define Good Performance Requirements**
 - Wide Solution Space
 - Captures Key Risk Areas
 - **Define Stakeholder Preferences in Requirements**
 - Value Functions for Performance within Objective
 - “Swing” Weights for Combining across Objectives
 - **Build Highly Aggregated Decision Models Early**
 - Address both Design and Qualification
 - Permit Updating as more Information Is Obtained
 - **Refine Decision Models as Architectures Are Refined**
 - Include Links to Peer Models
 - Include Links to Higher Level Models

Define a Decision Architecture

Derivation of Trade-off Requirements: The Problem

- **System-level Requirements Partitioned for Component-level Specifications**
 - **Input/output Requirements**
 - Requires system-level process model & matching component-level models
 - Some complex synthesis possible
 - **System-wide and Technology Requirements**
 - Use equivalence, allocation and synthesis principles
 - **Trade-off Requirements**
 - Divide value structure into segments
 - Requires that coherent design decisions can be made asynchronously and concurrently
 - **Qualification Requirements**

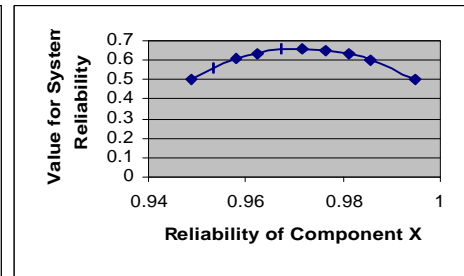
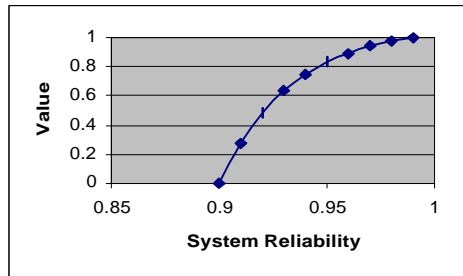
Consider Reliability Requirement

- **System Requirement**
 - Minimum Acceptable Threshold: 0.9
 - Design Goal: 0.99
 - Trade-off: Decreasing returns to scale
 - Assume exponential value function
 - $(1 - e^{-\alpha(r - r_{\min})}) / (1 - e^{-\alpha(r_{\max} - r_{\min})})$
- **Two Components**
 - Equal thresholds: 0.95
 - Equal design goals: 0.995
- **Problem:**
 - Can trade-off requirements for each component be found that are consistent with decreasing returns to scale for the system?

Derivation of Trade-off Requirements

Coefficient
of Exponential
Value Function

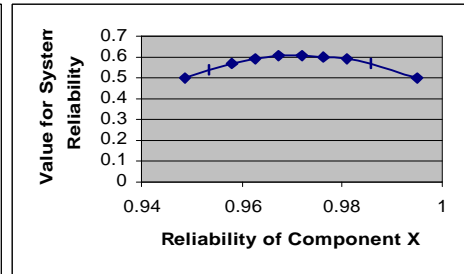
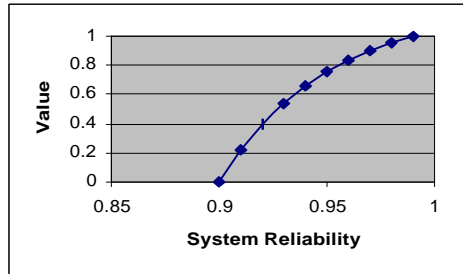
30



Value for System
Reliability of
0.9439

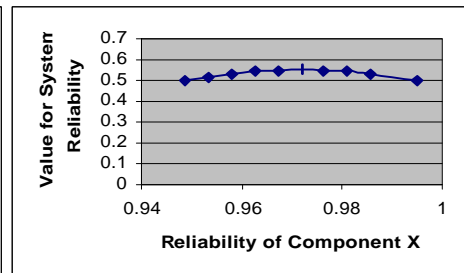
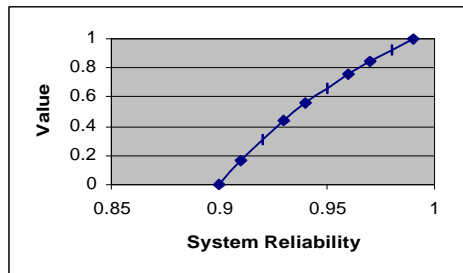
0.78

20



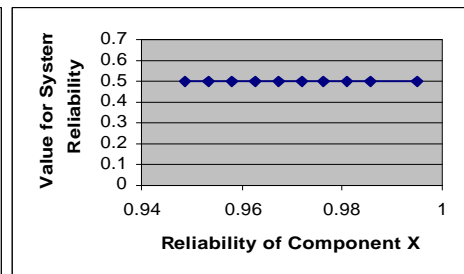
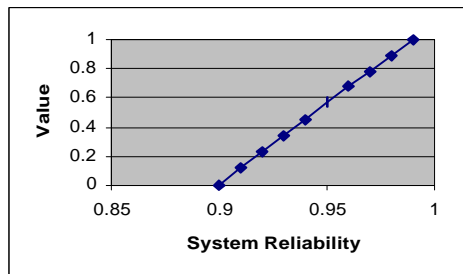
0.70

10



0.60

1



0.50

Summary

- **Engineering Design**
 - **Can Be Approached Systematically**
 - **Decision Analysis Can Be Used Throughout**
 - **Is Very Complex**
 - **Concurrent Systems/Problems**
 - **Trade Offs & Decisions Cross System Boundaries**
- **Searching for**
 - **Principles to Guide Decision Architecture**
 - **Derivation of Trade-off Requirements**
Permitting Decreasing Returns to Scale