33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

Architecture

Bringing Form to Function

Definitions

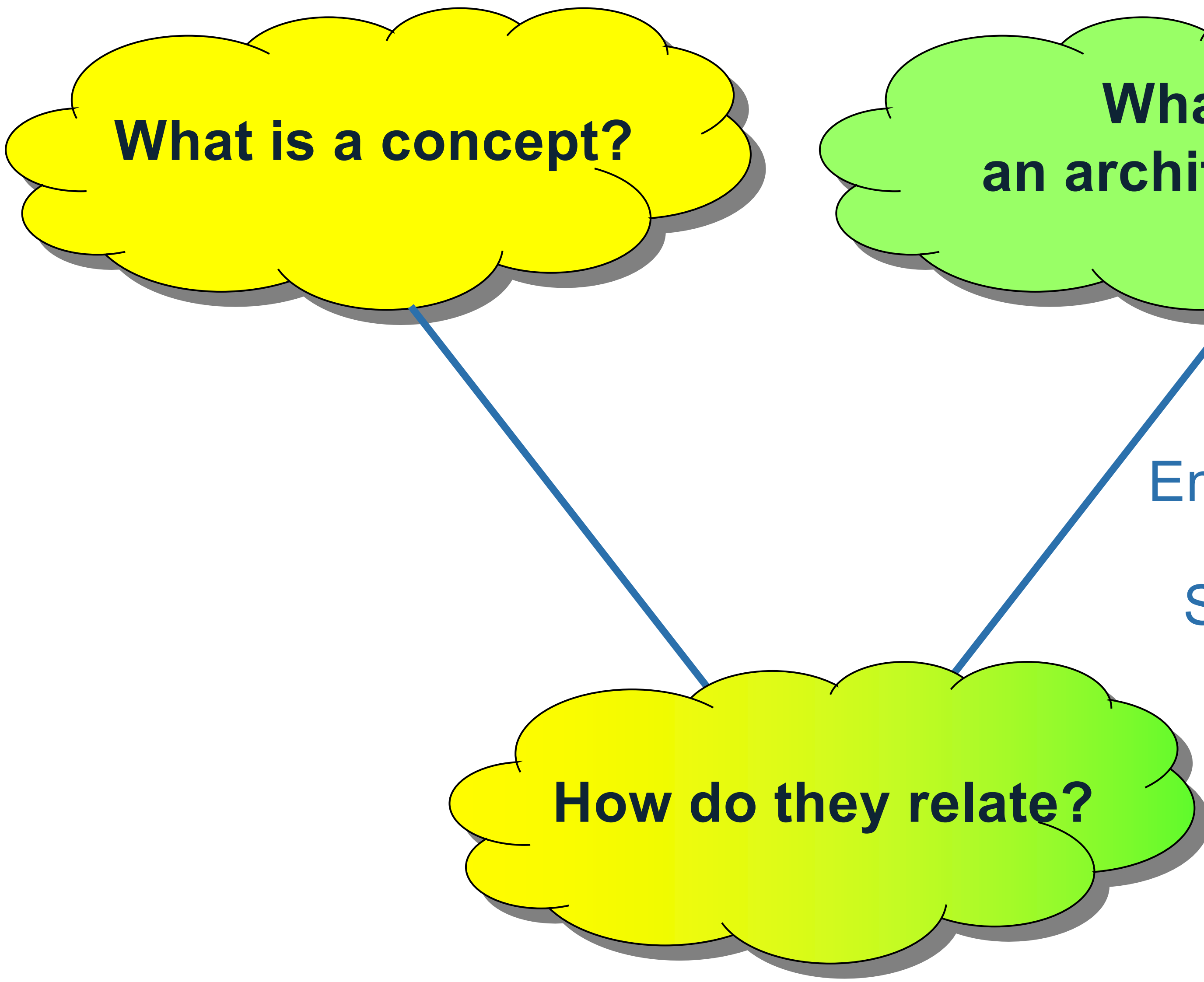
- **ΑΡΧΙΤΕΚΤΩΝ** ***(Greek)*** **= Master Builder**
  - Architéktōn is composed of two parts: ἀρχι (archi: to be the first, who commands) and τέκτων (tecton: mason, builder). That is, the boss who sends the masons.*

*https://www.archdaily.com/898648/etymology-in-architecture-tracing-the-language-of-design-to-its-roots

- **"The system architecture is the structure of the system."**
  - The essence of architecting is "structuring"
    - Bringing form to function
    - Converting the partially formed ideas into a workable conceptual model
    - Balancing needs and wants, fitting the interfaces, and compromising.

Eberhardt Rechtin, *Systems Architecting*, 1991

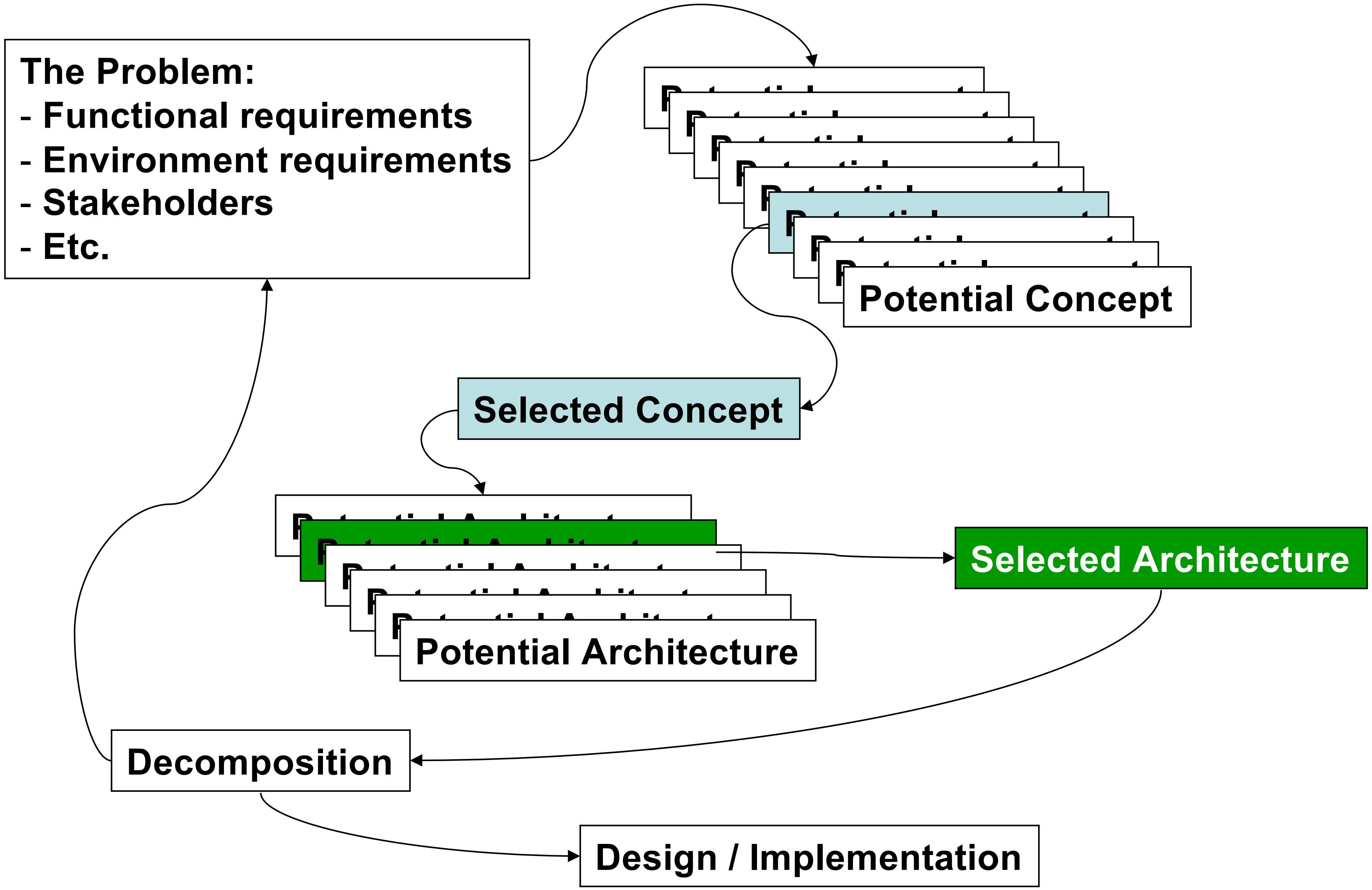
Discussion



Enterprise-level Architecture
or
System-level Architecture

StrategyBridge®

High-Level Process Flow



Architecture or Design?

- **"Architecture" is the high-level instantiation of the solution concept.**
  – Architecture decisions address strategy, purpose, and structure: how system elements and components will interact with each other and other systems.

- **"Design" is a specific solution implementation.**
  – Example: the systems engineering team on a large software project may make an architectural decision about whether to use a given technology, while the developers will make specific design decisions about how to implement data structures or algorithms consistent with that technology.

> *All architecture is design, but not all design is architecture. Architecture represents the significant design decisions that shape a system, where significant is measured in cost of change.* -- Grady Booch, Software Engineer

Architects and Systems Engineers

- Architecting is an essential part of Systems Engineering, especially for complex, unprecedented systems.
- The Architect's primary responsibility is to translate the vision of the user or operator to a system design.
- The Systems Engineer's role is to implement systems that conform to the architecture within programmatic constraints.

> *The architect, therefore, is not a "general engineer," but a specialist in reducing complexity, uncertainty, and ambiguity to workable concepts. The systems engineer, in contrast, is the master of making feasible concepts work.*
>
> E. Rechtin

Architectures Provide…

- A structure to address stakeholder requirements
- A set of rules and guidelines (constraints) for design consistency
- A partitioning concept compatible with the business model
- Guidance and priorities for trades during design
- A framework to allow design variants (e.g., different designs, same architecture)

Architecture is the structure — in terms of components, connections, and constraints — of a product, process or element.

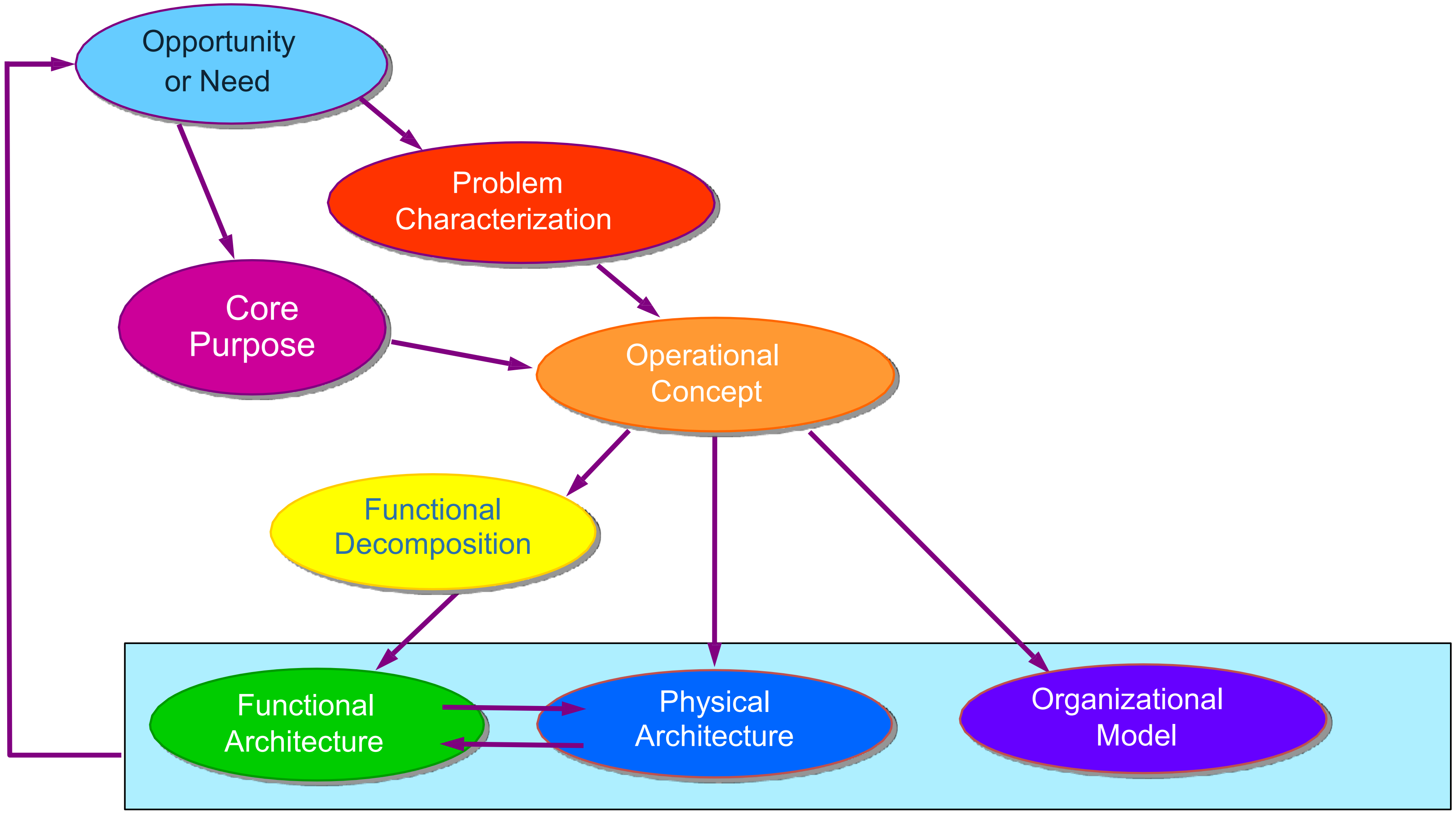
Maier & Rechtin, *The Art of Systems Architecting*, 2002

StrategyBridge®

Important Parts of an Architecture

- **The logical groupings of elements that make up the system and physical characteristics of those elements**
  - e.g., a car has a transmission, chassis, engine, entertainment, cooling system…
- **How the elements interact among themselves and externally**
  - e.g., The engine shall be mounted on the chassis …fuel moves from the gas tank to the engine through the gas line…
- **The rules and constraints that govern design**
  - e.g., no dashboard operation will require driver to use two hands

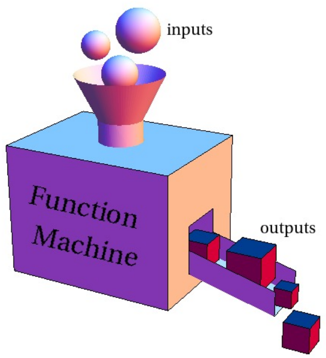
Architecting



Architecting provides a framework constituting a structure and a set of rules.
Designs represent specific implementations within this framework.

StrategyBridge®

Functional Architecture



- Definition
  - The set of functions and their sub-functions that defines the transformations of input flows into output flows performed by the system to achieve its mission.
- Purpose
  - Describe system requirements in (technology independent) functional terms
  - Basis for the physical architecture
- Represented in any of several notations:
  - Indented lists, IDEF notation, structure charts, flow diagrams, organization formats, class and object structure, etc.

Discussion: what are some top-level functions for your home entertainment system?

Functional Breakdown Example

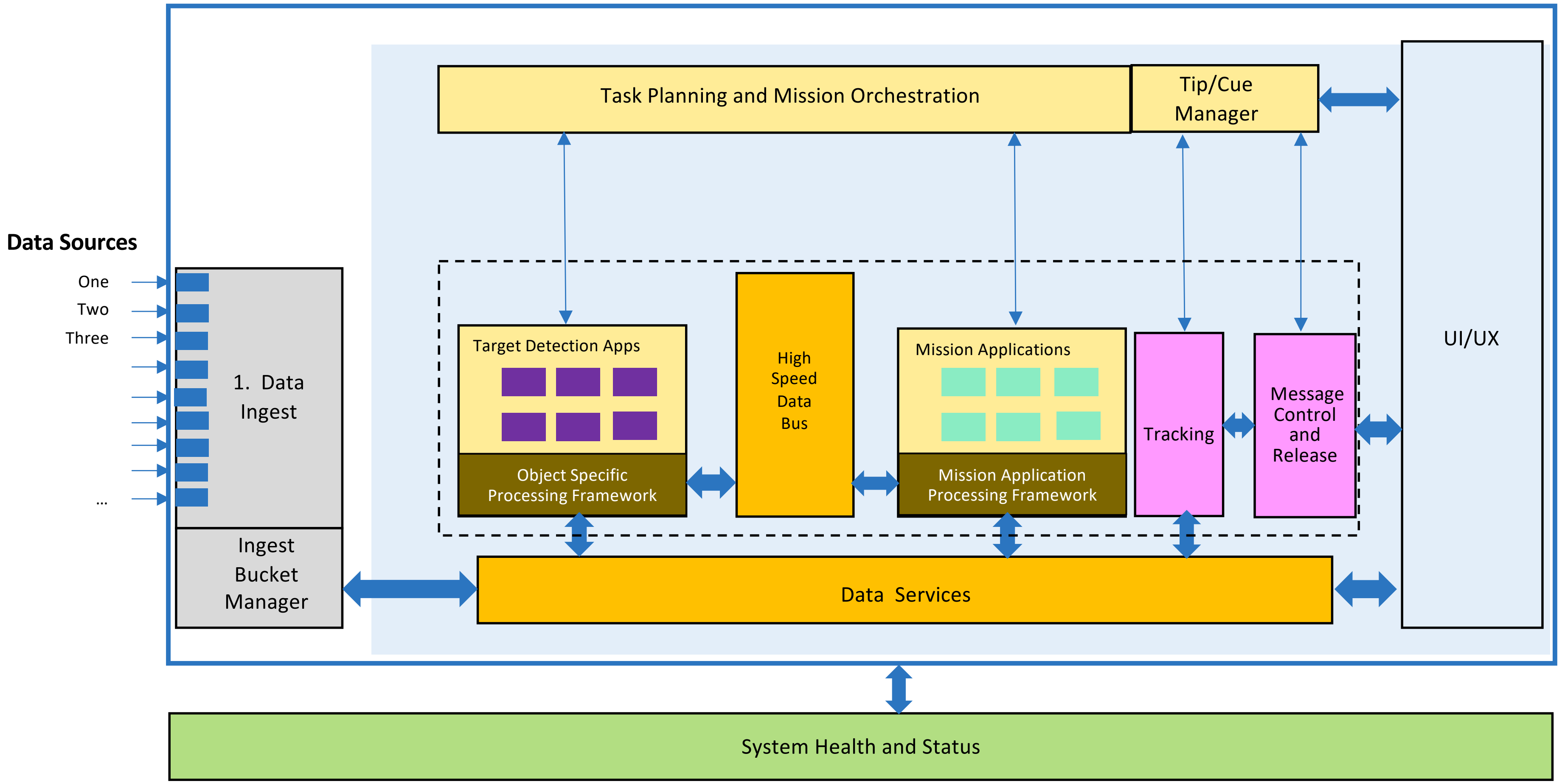
Pass another vehicle

Accelerate
- Increase fuel flow
- Increase RPM
- Downshift

Pass
- Signal
- Change lane
- Clear other vehicle
- Change lane

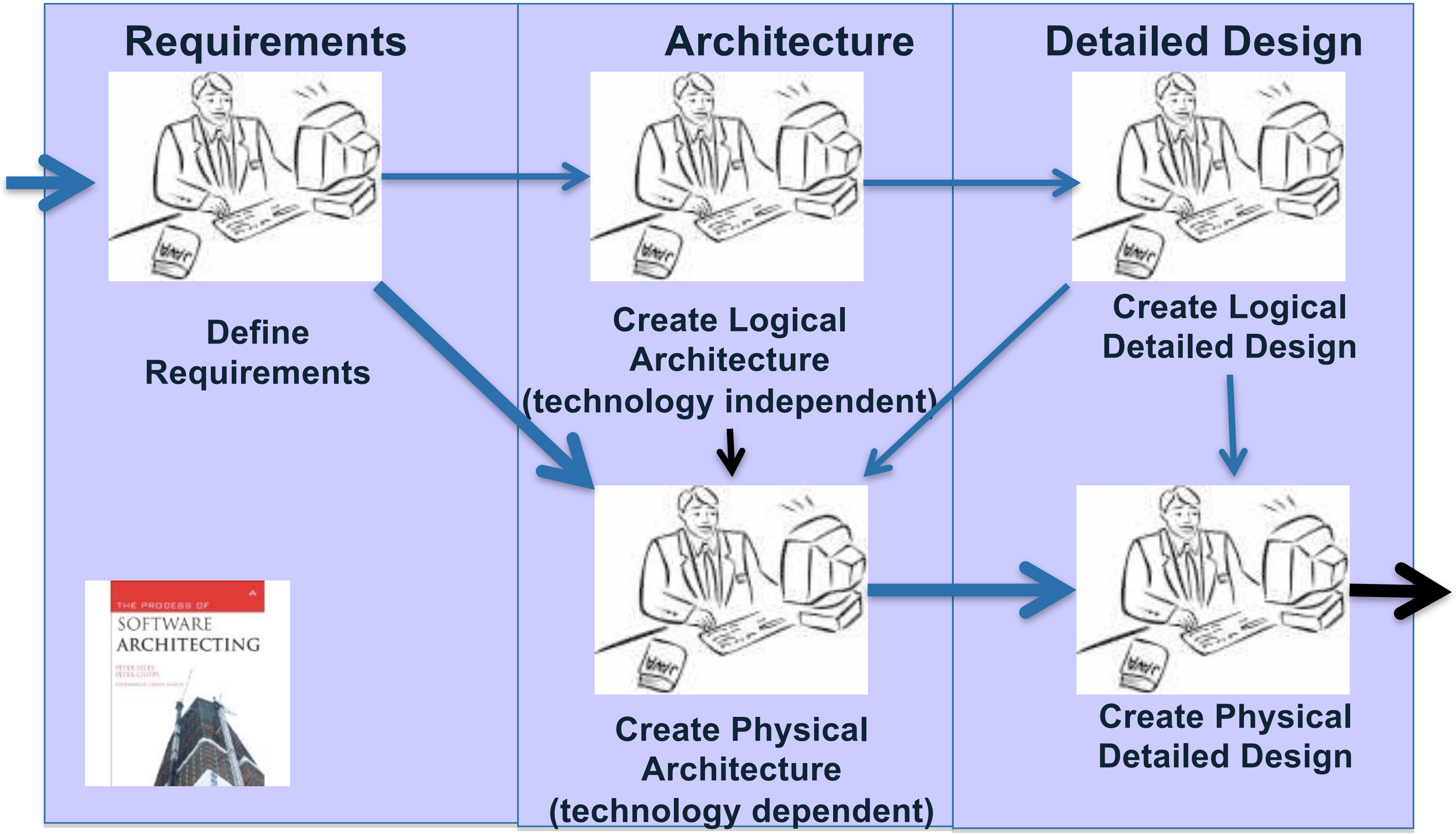
Decelerate
- Reduce fuel flow
- Decrease RPM
- Upshift

Functional Architecture Example



StrategyBridge®

Architecting is rarely linear…



StrategyBridge®

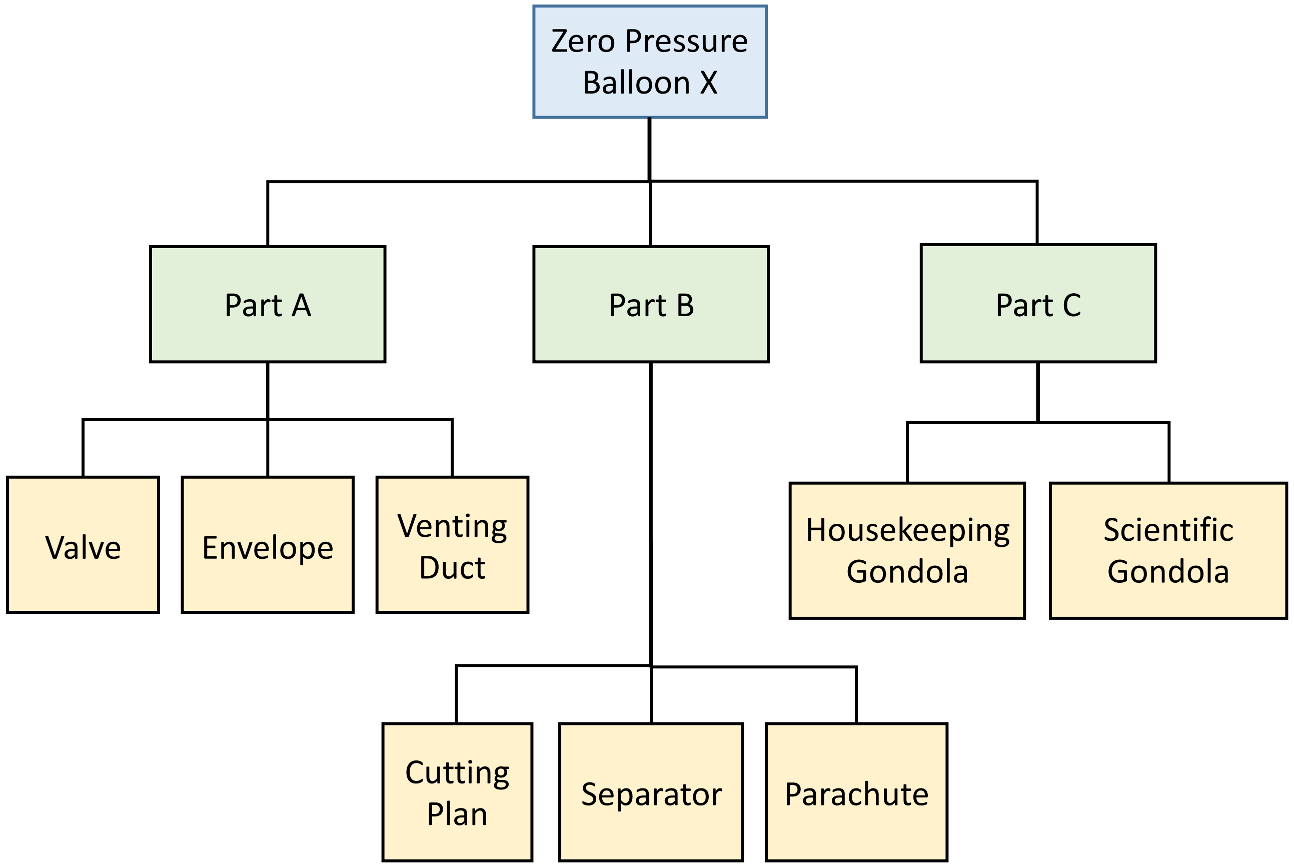
The Process of Software Architecting by Peter Eeles (Author), Peter Cripps (Contributor)

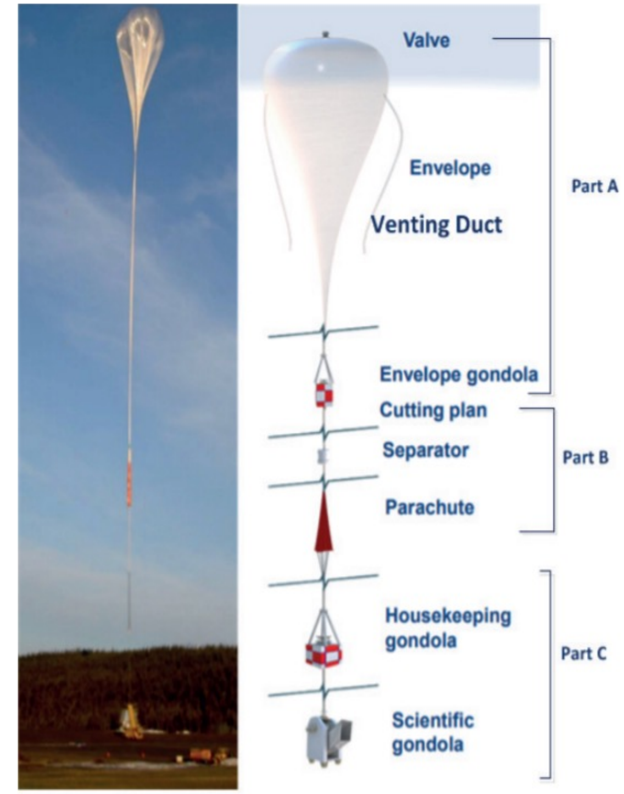
Physical Architecture

- Definition
  - An arrangement of physical elements (system elements and physical interfaces) that provides the design solution
    - Individual entities can consist of hardware, software, personnel, facilities, data, and procedures
    - All the system entities must be defined at each level in the system architecture so that all requirements can be allocated
    - Lower levels in the system hierarchy are defined *relative to* the design decisions imposed by the architecture at higher levels

Discussion: What alternative architectures could you consider to instantiate the video display function for your home entertainment system (TV)?

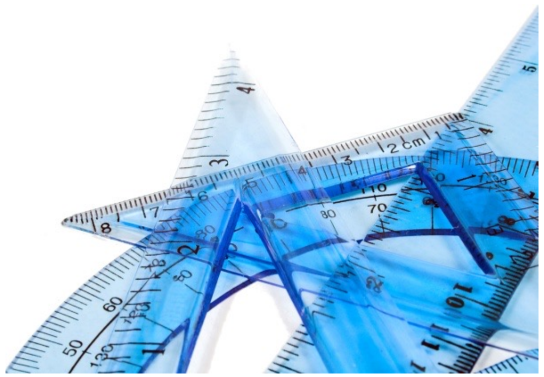
Physical Architecture Example





How to Decompose the Physical Architecture?

- You are finished with decomposition when…
  - You reach an entity that exists or can be purchased or contracted for, or
  - Someone or a small team can build the entity



- Conditions:
  - The requirements are clearly understood by the provider
  - We have confidence the entity will meet higher level requirements in the system decomposition

StrategyBridge®

Integration Required!

- Integration is the process of incorporating the lower-level system elements into a higher-level system element in the physical architecture
  - Integration will be required anywhere you decomposed an entity into two or more sub-elements: creates an interface.
  - The plan or strategy for the integration process, including the assembly sequence, may impose constraints on the design solution.
  - An assembled system element, also developed with the technical and technical management processes, may include fixtures for hardware or compilers for software.
- Integration also refers to the incorporation of the final system into its operational environment and defined external interfaces
- The integration process may require the development of enabling systems such as: test-beds, fixtures, simulation environments, or stubs

Source: Defense Acquisition Guidebook - Chapter 4.2.4.5

Integration Design

- Integration is an important driver for the system architecture
  - The integration approach that you will ultimately implement must be incorporated into the system architecture
  - Consider integration design early in solution definition
    - Identify as specific tasks in WBS
  - Interfaces are often high risk
    - Always a possible point of failure
  - Document interface and integration decisions
    - e.g., Interface Control Document

A Tale of Two Watches

- Bios and Mekhos both made fine, expensive watches
- The watches each consisted of one thousand parts
  - Mekhos assembled his watches bit by bit -- like making a mosaic
    - Each time he was disturbed in his work and had to put down the partly assembled watch, it fell to pieces and he had to start over
  - Bios made watches by constructing subassemblies of about ten components, each of which held together as an independent unit
    - Ten of the subassemblies could then be assembled together in a subsystem, and ten of those constituted the whole watch
    - If an interruption caused Bios to put down or even drop the watch he was working on, he had merely to reassemble that particular subassembly

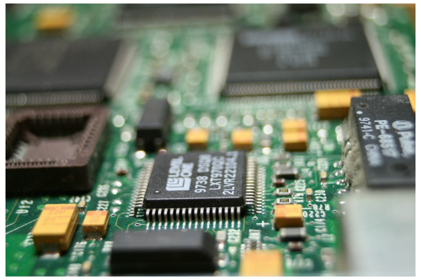
Adapted from: Meyer , M.H. and Lehnerd, A.P. (1997). *The Power of Product Platforms*. The Free Press, NY.

What kind of watch would you make?

What is an Interface?

- **Interface**
  - A common boundary between two or more systems or system elements (e.g., functions, physical components, organizations)
  - An interface is a point at which entities interact; it is also the interaction itself
- **Types of Interfaces**
  - Internal and external
  - Physical and functional
- **Initiated by:**
  - An output from a function
  - Need for an input to a function
  - A physical connection need
- **Completed by:**
  - Output identified as input to a function or external system
  - A physical connection termination

Interfaces are always an issue!



- **Every interface must be identified, documented and managed**
  - Internal – between the components, elements and subsystems
  - External – everything the system connects to or exchanges data with

Interfaces are defined during the architecting process

StrategyBridge®

Interface Management

- Establishes practices and procedures to ensure proper interface definitions, documentation and compliance throughout the system life cycle
- Develops interface control requirements that govern the development effort
  - Enables integration of system and subsystems
  - Facilitates contractor/supplier bidding
  - Supports maintenance and future enhancements/upgrades

Who manages interfaces on your projects?

Characteristics of “Good” Architecting

- **Context Sensitivity**
  - Understanding context of system implementation
  - Understanding of next level up functionality (external interfaces and communication protocols)
- **Time Sensitivity**
  - Understanding history of product family, product, and platform evolution
  - Understanding expected/anticipated future evolution
- **Stakeholder Sensitivity**
  - Technology Stakeholders
    - Linkage to technology opportunities
  - Business Stakeholders
    - Linkage to business model / market opportunities and trends
  - User Stakeholders
    - Compatibility with organizational structure, roles, charters

What is an example of “good” system architecting you interacted with today?

Poor Architectures may...

- **Be infeasible**
  - Cannot be implemented within triple constraints
- **Verify but not validate**
  - Does the architecture meet the user needs (CONOPS)?
- **Introduce unnecessary complexity**
  - Complexity affects integration and test; often leads to significant learning curves for developers, users, and maintainers
- **Become fragile over time**
  - Difficult or too expensive to update technology, change a component, or add new features

A Practical Illustration:
How does architecture fit with the SE process?

- Need Statement
  - My neighbor needs to light his garage so he can safely and effectively work on his classic car at night

CONOPS

- Work occurs evenings year round performed by a single individual ~60 years old
- The work area has a garage door opening that faces north and has heat but no cooling
- Work occurs without consideration of weather with the door open or closed as appropriate based on weather, temperature, and mosquitos
- While the work typically focuses on a specific element, the access to tools and work bench require lighting over a broad area

StrategyBridge®

Domain-Specific Knowledge

- Amount of light measured as footcandles or lumens
- Light quality measured using:
  - Contrast – ability to distinguish items
  - Flicker – pulsing in the light from electrical source
  - Glare – reflectance from surfaces from bright lights
  - Color – affects ability to distinguish colors
  - Efficacy – lumens per watt
  - Heat – tied directly to watts used

Requirements Analysis

- Footcandles as required to see well for detailed work
- Contrast – working on a black car
- Flicker – not noticeable
- Glare – minimal
- Color – natural
- Efficacy – as efficient as possible
- Heat – Limited for summer operation

Requirements

- Footcandles – at least 50 and less than 100
- Contrast – 3 to 1 between adjacent areas
- Flicker – None observed
- Glare – See contrast
- Color – kelvin temperature > 5000
- Efficacy – > 40 lumens per watt
- Heat –
  - < 10 degree heat conductive heat gain at 1 foot
  - < 20 degrees radiant heat gain at 1 yard

System Concepts and Architecture

- Possible Concepts:
  - Many small lights ceiling mount
  - A few large lights ceiling mount
  - Combined ceiling and local work lights
- Architecture Decision
  - 12 small ceiling lights (4 across by 3 front to back)
    - 3 sets of lights, 4 lights per set

Allocated Requirements

- Each Set of 4 lights
  - Lumens - 5000
  - Flicker – None observed
  - Color – kelvin temperature > 5000
  - Efficacy – > 40 lumens per watt
  - Radiant Heat – less than 5 degrees at 3 feet
  - Conductive heat
    - < 5 degree heat conductive heat gain at 1 foot
    - < 5 degrees radiant heat gain at 1 yard

Possible Light Bulb Concepts

- Incandescent
- Halogen
- Sodium Halide
- Fluorescent Tube
- Compact Fluorescent
- Light Emitting Diode

StrategyBridge®

Selected Light Bulb Design

- For each light (12 lights total):
  - LED floodlight bulbs (downward pointing)
  - Downward reflector with diffuser
  - 1500 lumens
    - Meets requirement (5000 lumens per set of 4)?

Unit Testing

- Verify performance of each light against the requirements using lab equipment to measure performance
- Resolve any deficiencies to cause and correct to achieve required performance

StrategyBridge®

System Testing and Validation

- Verify performance against system requirements using laboratory equipment
- Validate total lighting by installation in the target environment and perform an inspection of usability of the resulting light against the original need statement
  - Solicit feedback on level of user satisfaction

StrategyBridge®

33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

What are your questions?

StrategyBridge®

Additional Resources

- INCOSE Systems Engineering Handbook, Section 4.4, Architecture Definition Process
- *The Art of Systems Architecting*, Eberhard Rechtin and Mark Maier, CRC Press, 1997
- *Systems Architecting: Creating & Building Complex Systems*, Eberhard Rechtin, PTR Prentice-Hall, 1991
- *To Engineer is Human: The Role of Failure in Successful Design*, Henry Petroski, Vintage Books, 1992
- *The Design of Everyday Things*, Donald A. Norman, Doubleday, 1990
- *The Power of Product Platforms,* Meyer, M.H. and Lehnerd, A.P., The Free Press, NY, 1997
- *The Tall Office Building Artistically Considered*, Louis H. Sullivan, Lippencott's Monthly Magazine, March 1896
- *Why Buildings Stand Up: The Strength of Architecture*, Mario Salvadori, W W Norton and Company, 1990

StrategyBridge®

33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023