

Projects Don't Begin With Requirements

Presented By

Kevin Forsberg, ESEP

The Center for Systems Management

Vienna, VA

kforsberg@csm.com

“Quality is Free!”

True, if ...

In 1979 Philip Crosby said: “Quality Is Free!”

- The definition of quality is ***conformance to requirements***
- The system of quality is prevention
- The performance standard is zero defects
- The measurement of quality is the price of nonconformance

History has proven that Crosby was and is still right,

- IF the project requirements are complete and correct.
- Adhering to an incomplete requirement set will seldom lead to user satisfaction.

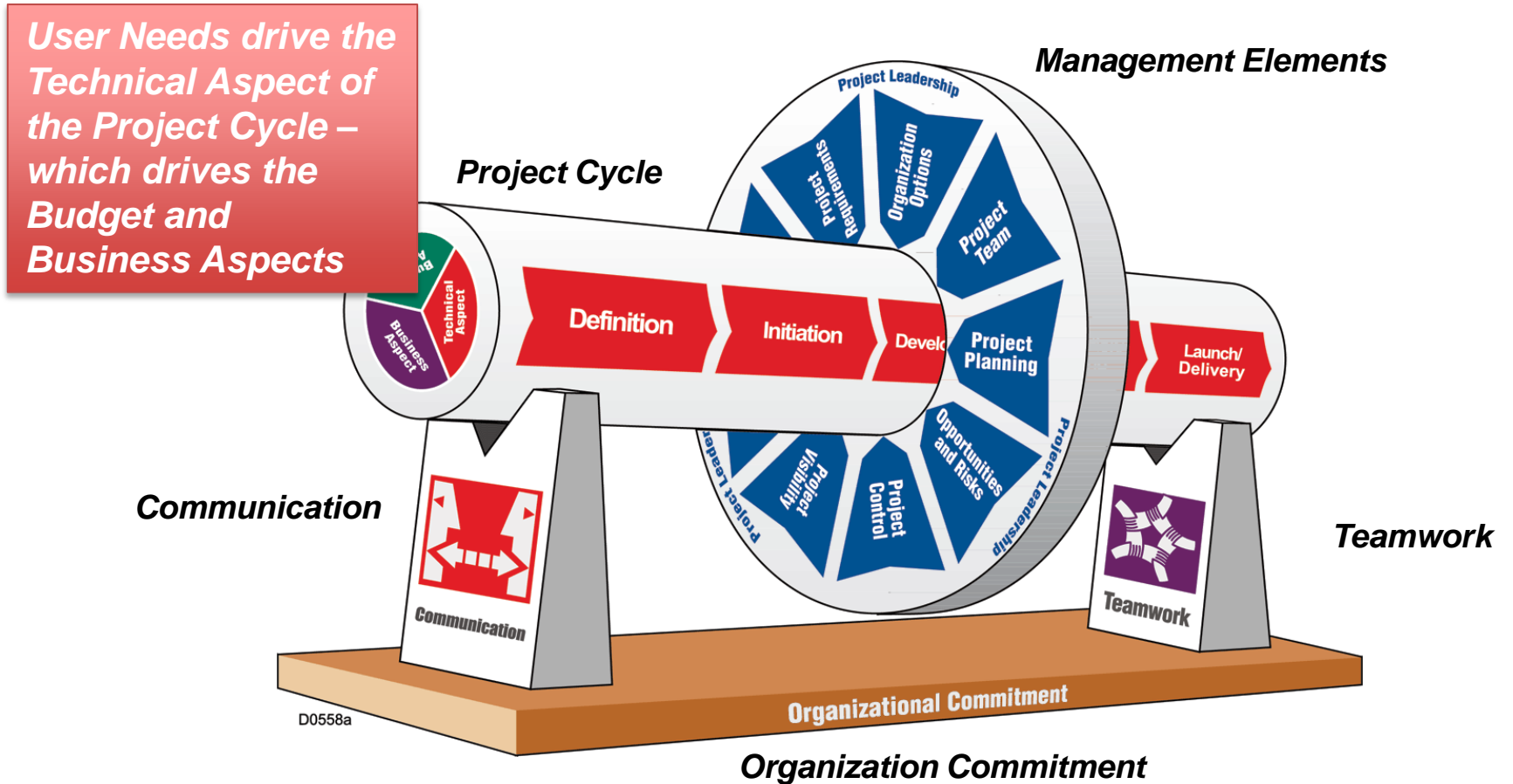
Projects Begin With User Needs

Possible starting points for new projects:

- New opportunities
- New threat
- Emerging technology
- Paradigm shift
- Competitive pressure
- Political restructuring
- Performance shortfall for legacy systems
- Business case change for legacy systems
- Additional features required for legacy systems

Systems Management Process Model

The Five Essentials



Step 1 - Identify the Stakeholders

The Systems Engineer is responsible for ensuring all relevant Stakeholders are identified:

- Product users
- Service users
- Operational users
- Customers
- Management
- Associates
- Maintainers and logistics support
- Regulators
- Future users

Bus Fleet Stakeholder Analysis – Example

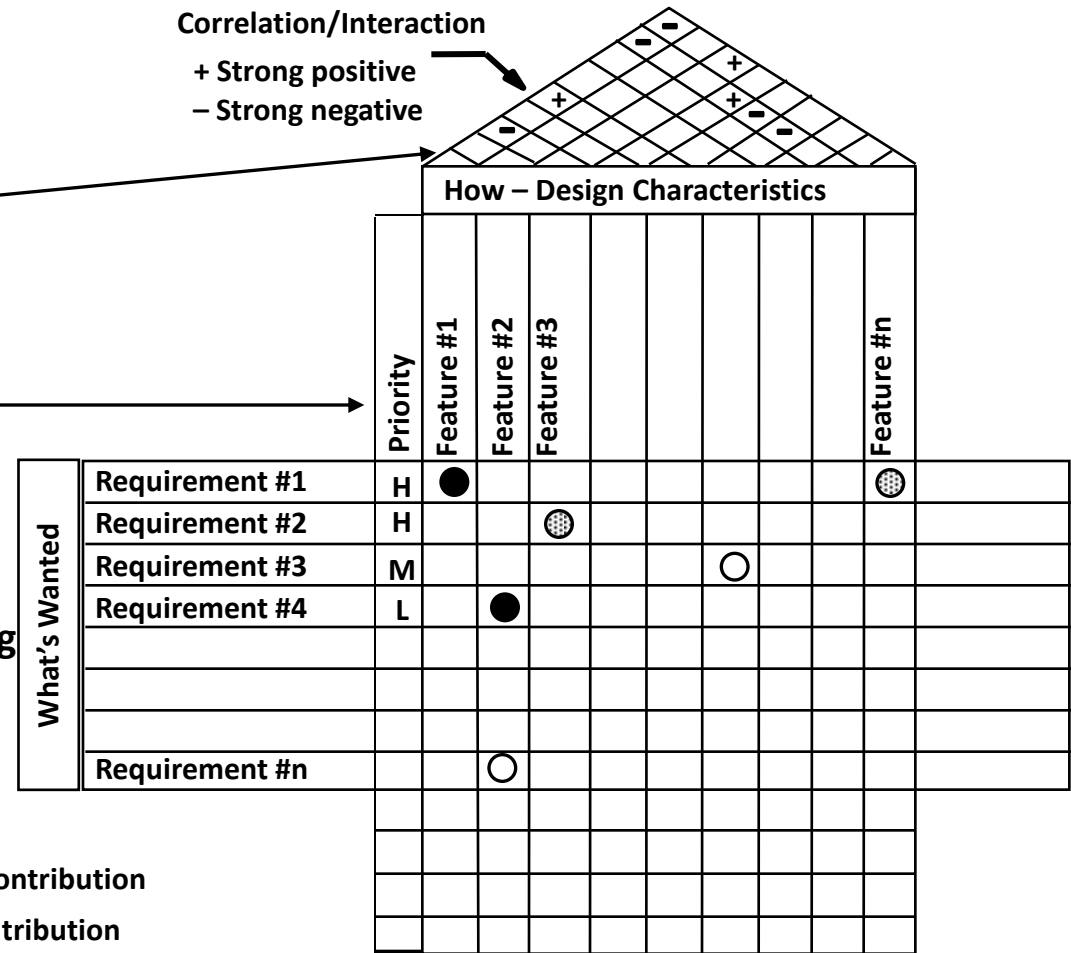
– Project – Provide a bus fleet

Stakeholder	Type	Expectation	Priority
Bus line Owner	Product User	<ul style="list-style-type: none"> • Low price • Low maintenance & operations costs • Resale value 	
Environmental Protection Agency	Regulator	<ul style="list-style-type: none"> • Low emissions 	
State	Regulator	<ul style="list-style-type: none"> • Safety 	
Program Office Boss	Executive Management	<ul style="list-style-type: none"> • Low development cost 	
Passengers	Service User	<ul style="list-style-type: none"> • Handicapped access • Reclining seats • Package storage 	
Drivers	Operational User	<ul style="list-style-type: none"> • Comfortable driver's seat • User friendly controls • High MTBF 	
Recreational Vehicle Owner	Future User	<ul style="list-style-type: none"> • Ease of conversion • Attractive appearance 	

House of Quality – A Requirements Management Technique

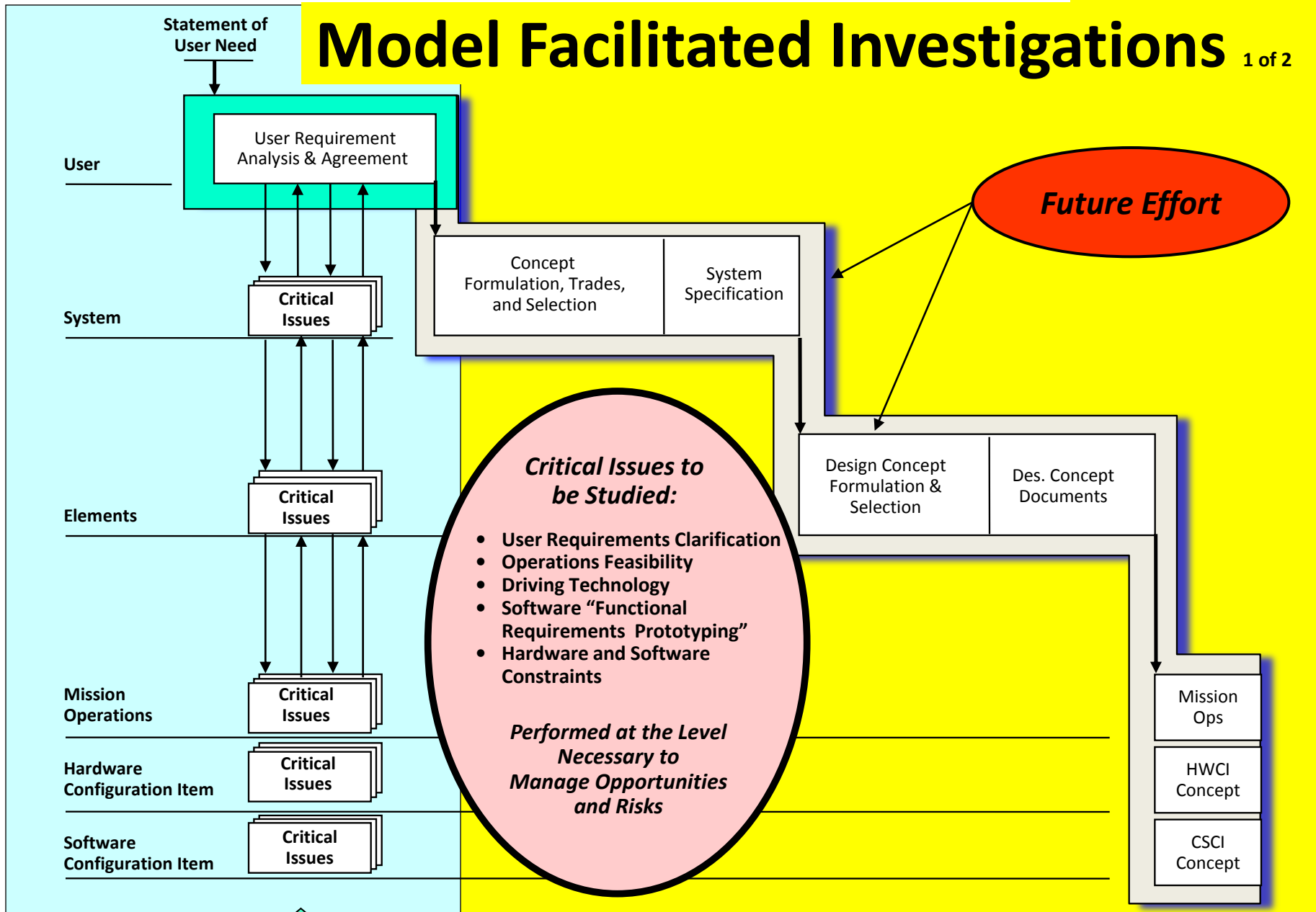
- Correlates requirements to requirements satisfaction locations:

- Roof illustrates compatible and conflicting situations e.g., Size and weight
- Priorities emphasize importance
- Supports comparison of design alternatives
- Family of matrices facilitates tracing satisfaction to components, parts, and processes



- Strong Contribution
- Mild Contribution
- Negative Contribution

Model Facilitated Investigations 1 of 2

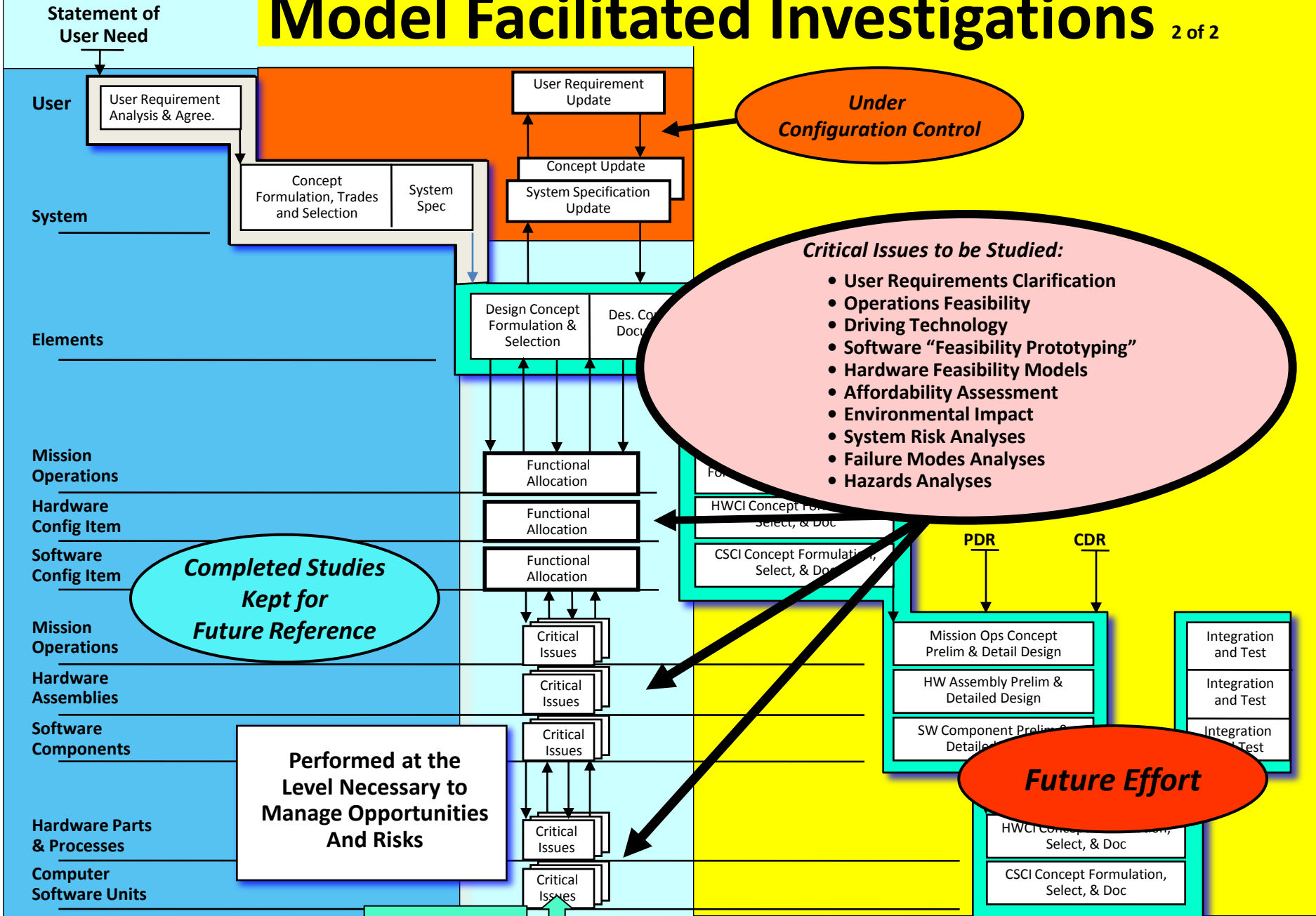


“Mars Mission Has Some Seeing Red”

Washington Post, February 2009

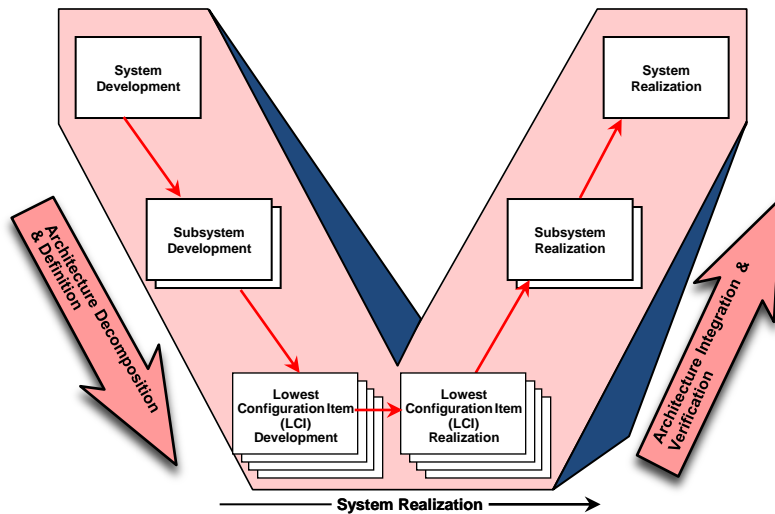
- The next NASA mission to Mars was to be the Mars Science Laboratory. At 1,875 pounds (845 Kg) it is twice as heavy as the largest previous Mars rover.
- Technical “glitches” in final checkout caused a delay and the team missed the launch window which caused a two year program delay.
- The program office said that as a result the program cost has increased by almost 30% (to \$2.2 billion).
- However program critics say the cost has really increased by 400% since the project was first dreamed up. They now want to kill the project, arguing that the new rover is too ambitious.

Model Facilitated Investigations 2 of 2



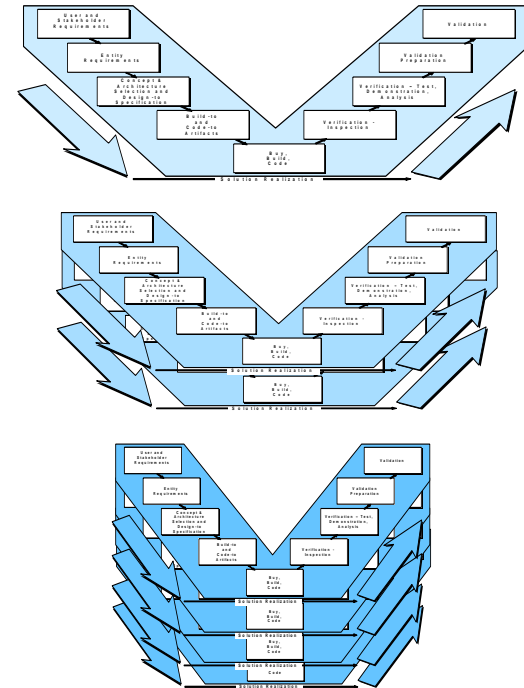
Dual Vee Model

Architecture Vee for architecture management



Depicts architecture baseline evolution. Vertical dimension is architecture decomposition. Horizontal dimension is system realization. Third dimension normal to the image is quantity of entities and their interfaces.

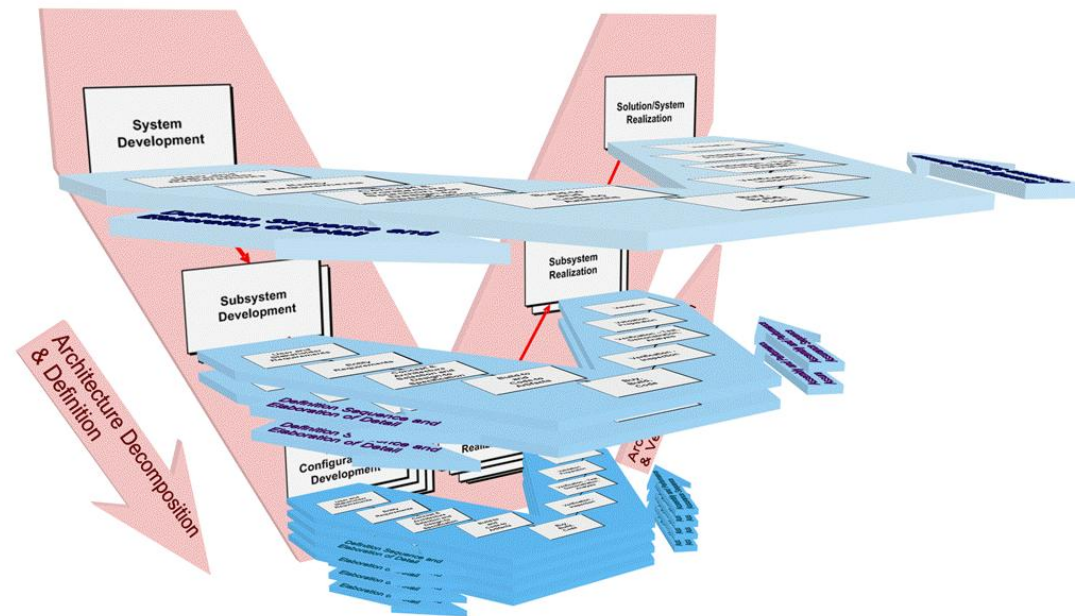
Entity Vee for entity management



An entity is any item of the architecture. The Vee depicts entity baseline elaboration. Vertical dimension is entity detail. Horizontal dimension is entity realization.

Concurrent Architecture and Entity Development

- Depicts concurrent architecture and entity development:
 - There is one architecture Vee
 - There is one entity Vee for each entity within the architecture
 - The architecture Vee depicts architecture baseline evolution
 - The entity Veeps depict baseline evolution for each entity

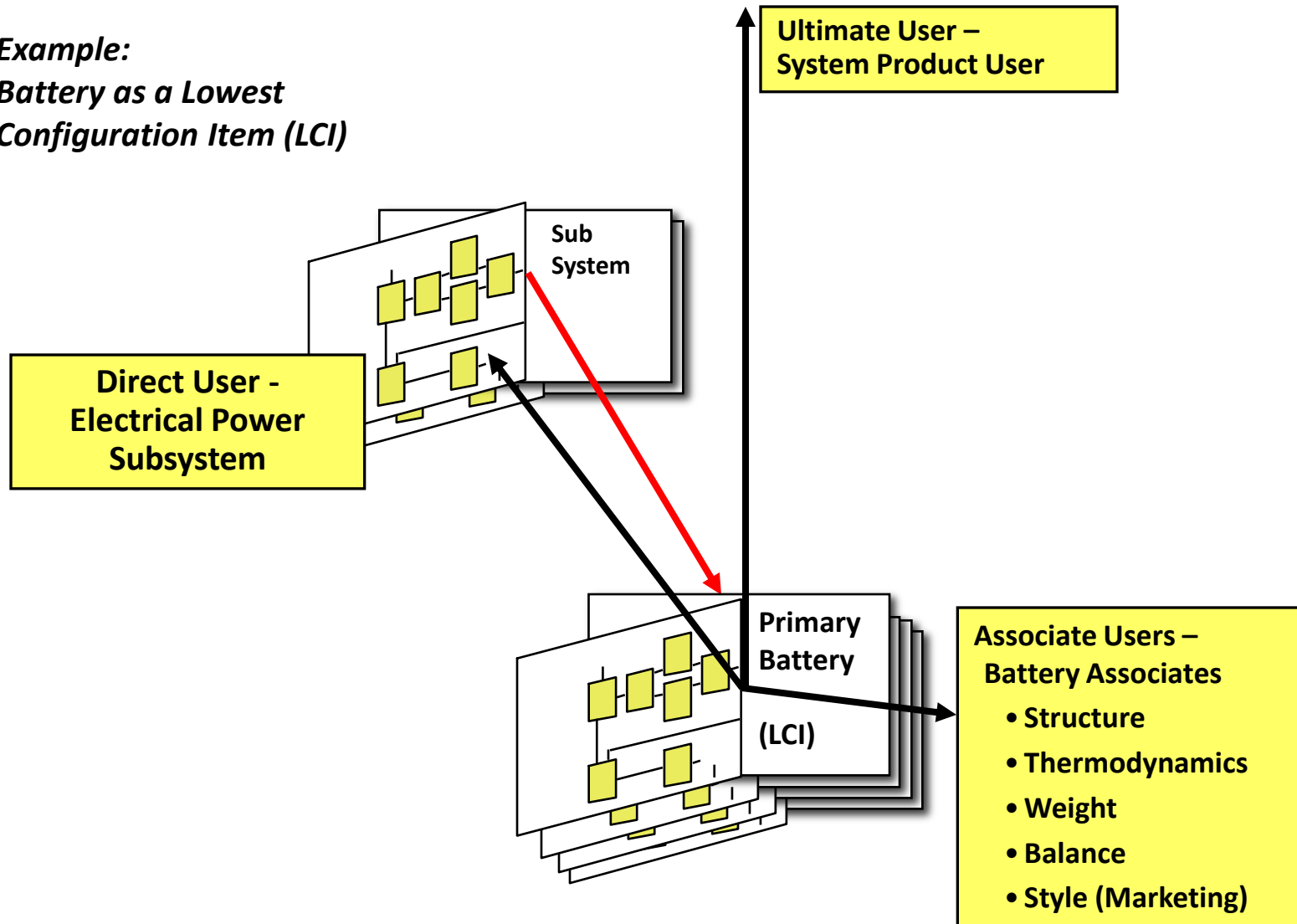


Ref: VPM p 350

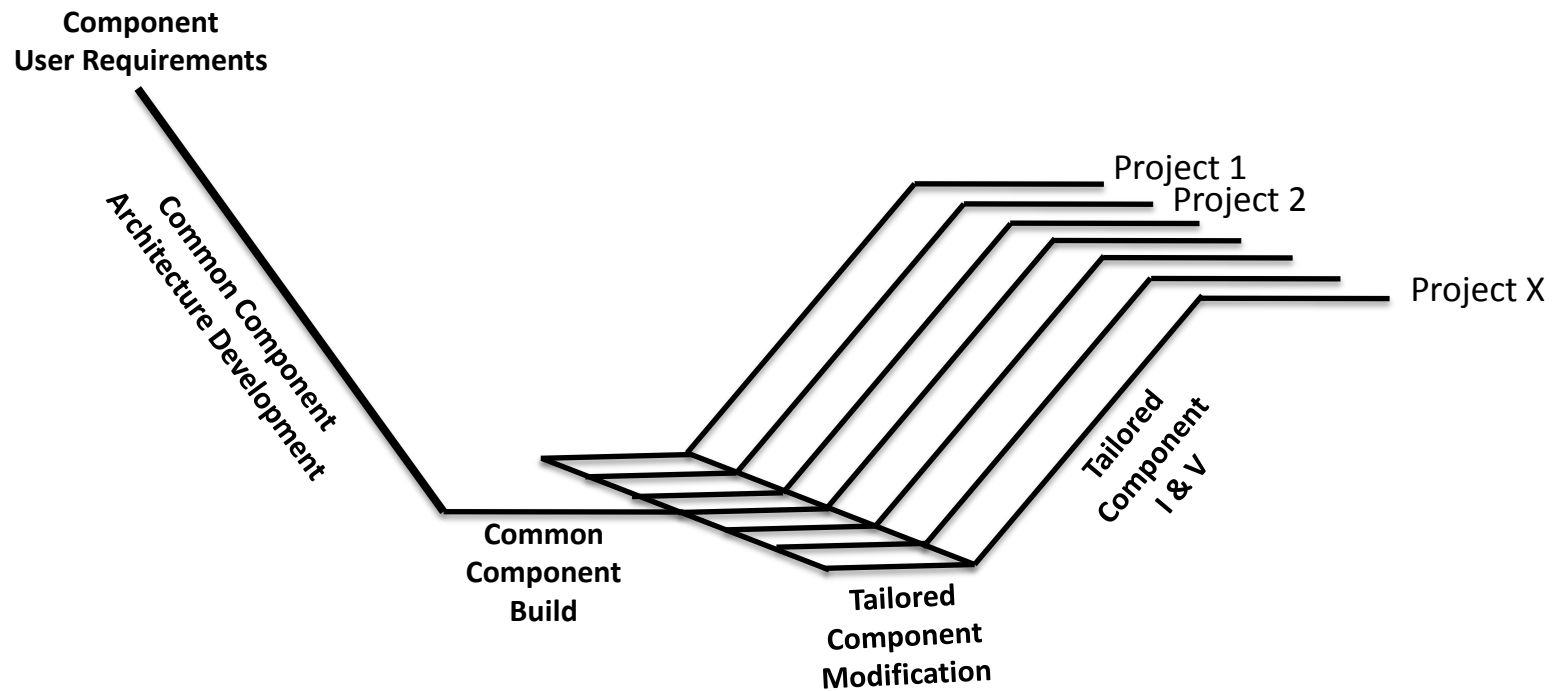
Three Categories Of Users Must Be Satisfied At Every Level – Example

Example:

Battery as a Lowest Configuration Item (LCI)



Component Level Development



Mobile Phones and Their Batteries

Table 1: Number of Current (2010) Nokia Mobile Phones and Related Batteries

Number of different mobile phone models	108	Average number of phones supported per battery design	2.4
Number of unique batteries	45	Number of batteries that support three or more phones	9

Source: <http://www.1800mobiles.com/nokia.html>

Case Studies (1 of 2)

Examples of projects that missed key User Needs:

- **LockAlloy – Research Product: New material with No Users**
 - Beryllium: Stiffer and stronger than steel; lighter than Aluminum – but very brittle
 - Aluminum: Ductile, but less stiff and weaker than Beryllium
 - LockAlloy: Stronger and stiffer than Aluminum, but ductile
 - Development team failed to sell it to internal projects
- **LI-1500 Fibrous Rigid Insulation – Research Product: No user**
 - Lockheed Patent Attorney said “No user, No patent”
 - It sat for 3 years on shelf
 - By pure chance Lockheed Shuttle Program Manager found the material
- **“Lunch Bucket” – Internal users (manufacturing, logistics) ignored**
 - Product is a ten-ounce serving (soup, stew, chili, etc.) in microwavable container
 - All internal users (manufacturing, logistics, etc.) were identified in serial fashion
 - It took EIGHT years to get to market – and market window was lost
- **Nokia mobile phone batteries – Logistics ignored**
 - Since 1992 Nokia has held major market share
 - In 2010 they have 108 different models currently available
 - These 108 models use 45 different batteries
 - Missed chance to improve commonality of batteries over multiple phone models

Case Studies (2 of 2)

Examples of projects that missed key User Needs (Continued):

- **“AutoCon” Contract generation – Missed upper management as user**
 - The “operational COTS product” the team bought turned out to be a Windows NT alpha-version; a 6-month delay ensued
 - What got the PM fired is that the team missed the management reporting requirements
- **Aquila Remotely Piloted Vehicle – Operational users involved too late**
 - The user command and the contractor decided to build a “Cheap, throwaway RPV” for battlefield tactical surveillance (< \$20 K)
 - The delivered product met both their expectations
 - The operational commands were then shown the Aquila, which they loved – except the display resolution was poor;
 - New electronics worked beautifully – but cost \$500 K
 - Contract required world-wide deployment, which meant cold start at minus 50° but when user in Panama wanted the vehicle the CO said “no;” The engine won’t start at minus 50°
 - Throw-away concept bit the dust, and after a lengthy struggle, so did Aquila
- **The U-2, the TR-1A, and SOTAS –**
 - US Army helps out in justification of US Air Force program ... by cancelling their SOTAS
 - After the program was underway, cost overrun causes program de-scope, and the Army loses (just as happened in Shuttle for NASA and DoD payloads)
- **Hubble Space Telescope –**
 - Ultimate users ignored (which led to the book: *“Hubble Wars, where Astrophysics Meets Astropolitics”*)

Conclusions

- Input from the user community is essential *before starting*
- **Continuous** input from the user community is essential throughout the project
- It is essential to have a **broad interpretation** of who users are
- The **systems engineer is responsible** for managing stakeholder expectations to stakeholder satisfaction