



***SI4000 Fall AY10
Systems Engineering Colloquium***

Thursday October 1st, 1500, SP321

***“Up and Down The Systems
Engineering “V””***

Dr. Gerald Mulvey

Northrop Grumman Aerospace Systems

Gerry.mulvey@ngc.com

Irish69th@msn.com



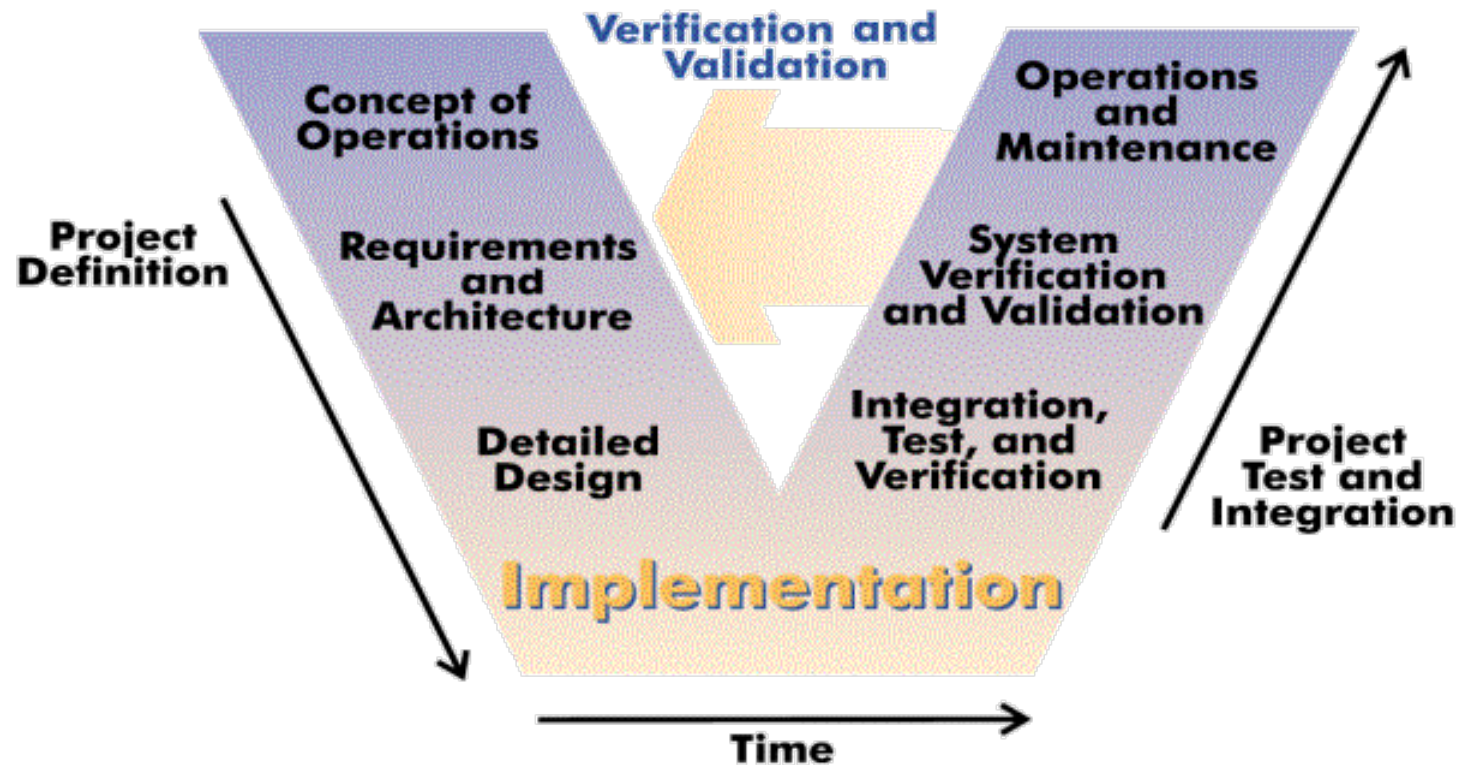
Outline



- **A series of case studies of system engineering practice successes and failures across the normal system engineering project tasks will be discussed.**
- **Each case will be presented in the following format**
 - **Project background**
 - **Systems Engineering challenge**
 - **Approach taken**
 - **Results**



System Engineering “V”





The Case Studies*



- Afraid of Systems Engineering – How do we test
- We all know Systems Engineering – Spec is not important
- Systems Engineering Costs Too Much – Pay now or pay later
- The COTS Mixture – Design review success
- The Trust Balance – Control by ICD

* The names of the companies and the projects have been removed



Afraid of Systems Engineering

– But how do we test

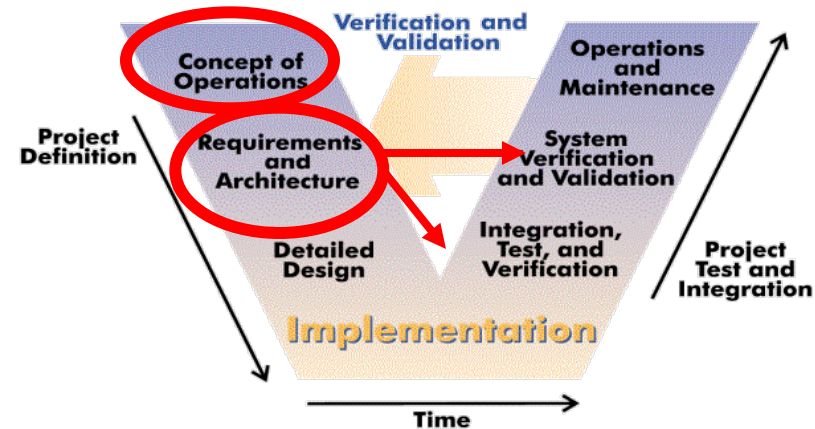


- **Project Background**

- S/W & H/S System
- \$10M
- Systems Engineering
 - Specialty Engineering
 - Top Level Architecture
 - System Integration

- **Systems Engineering Challenge**

- Develop Standard Documentation
 - Ops Con, Specs and ICDs
- Design Engineering Choose to Block Documentation Development





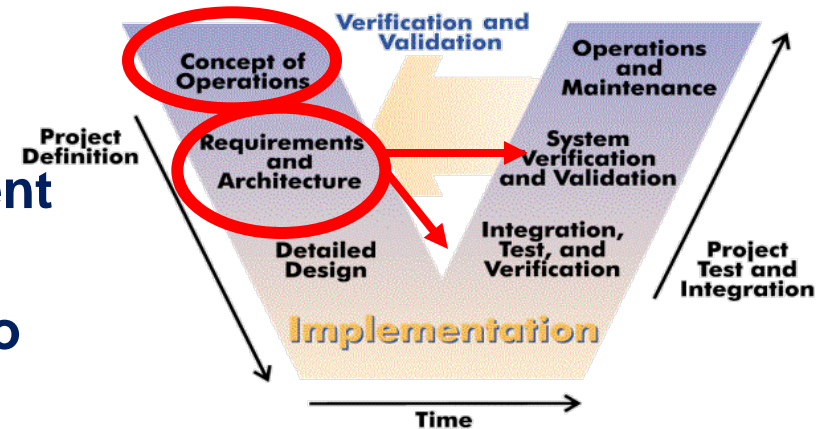
Afraid of Systems Engineering

– But how do we test



- **Approach Taken**

- **System Engineering Withdrew From Documentation Development Role**
- **Design Engineering Proceeded to Develop the System**



- **Results**

- **Design Engineering Did Not Know How to Test the System**
- **System Engineering Re-engaged Developed an Informal Opscon, Requirement Set, Testing Approach/plan and Test Data**
- **System Passed Tests**
- **Degree of Over-design Unknown,**
- **Cost of Over-design Unknown**



We all know Systems Engineering

- The spec is not important

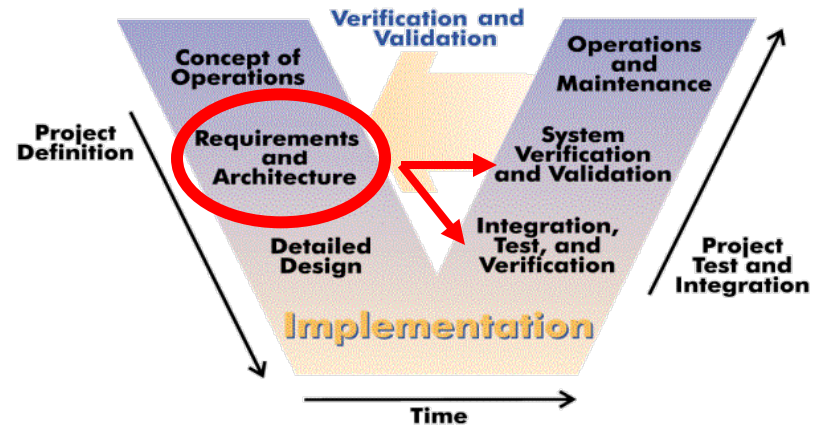


- **Project Background**

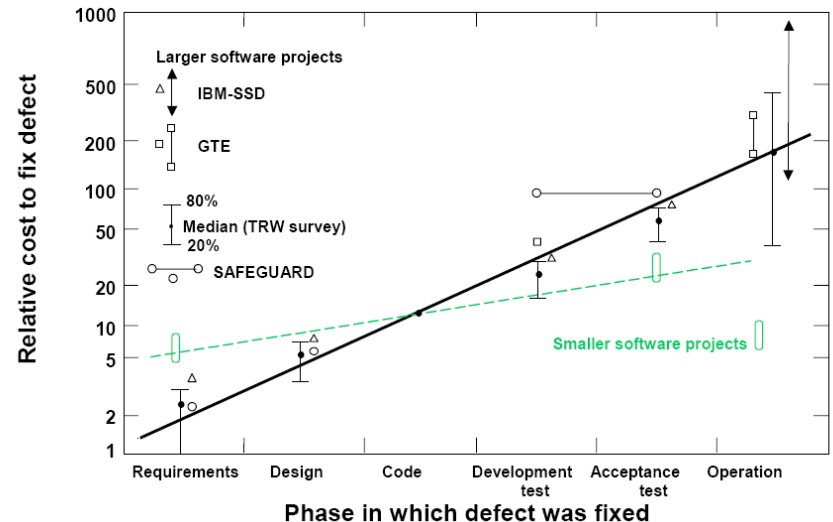
- H/W
- \$10M
- Systems Engineering
 - Performed by Design Engineering
 - Specification 25 Pages With Hand Written Annotations and Approximately 25 TBXs

- **System Engineering Challenge**

- Program Past CDR
- Specification Remains As It Was the Proposal
- Prepare for Testing and Sell-off



Cost to Fix an Error [ref 2 & 3]





We all know Systems Engineering

- The spec is not important

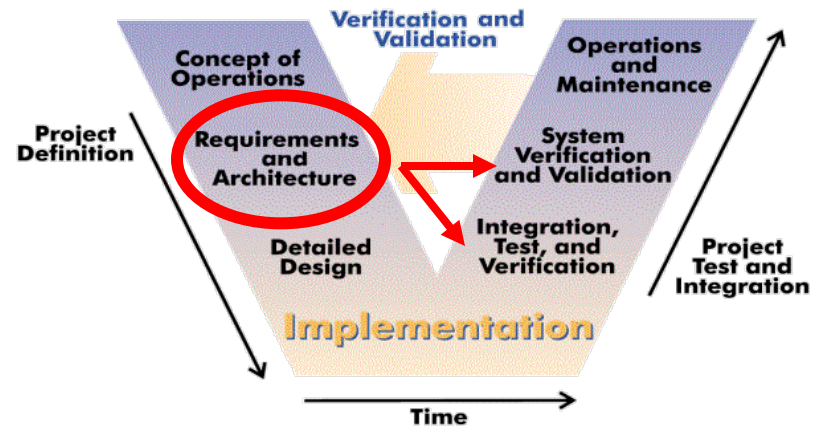


• Approach Taken

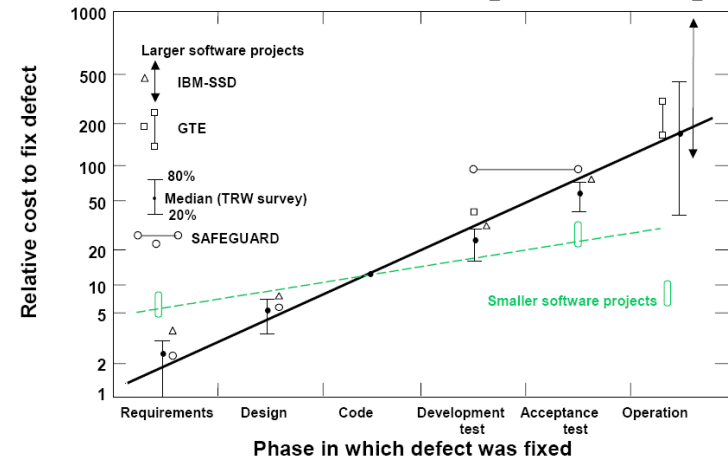
- Update Specification
- Control Requirement Creep
- Develop VCRM

• Results

- Customer Continued to Change Specification Without Agreement
- Customer Refused to Sign Spec
- Customer Refused to Agree on Required Testing
- Customer Refused Product Delivery
- Contract Went to Arbitration
- Contractor Retained Product Incur \$90M Cost Overrun



Cost to Fix an Error [ref 2 & 3]



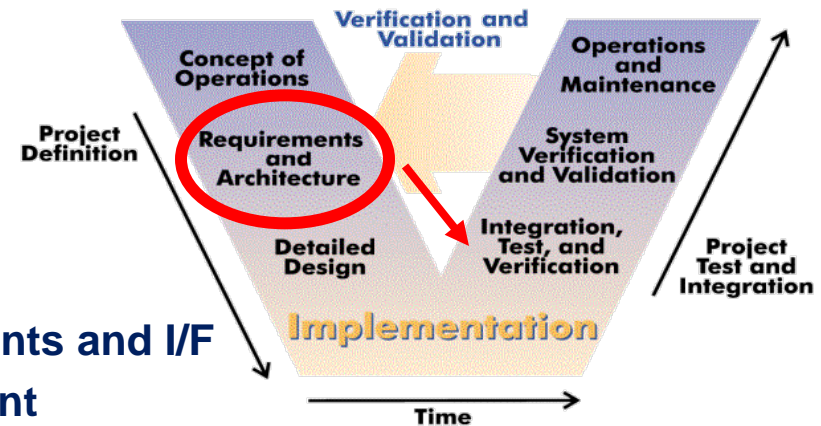


Systems Engineering costs too much – pay now or pay later



- **Project Background**

- H/W & S/W
- \$250M
- **Systems Engineering**
 - Develop and Control Requirements and I/F
 - Plan for and Conduct Deployment
 - Plan for and Execute Verification
 - Plan & Conduct Sell-off



- **Systems Engineering Challenge**

- **Management Cut SE Budget to Meet Their “Expectation”**
(Top Down Allocation, Price to Win, Cut Fat, Lean and Mean, Provide Challenge)
- **Plan for Success**
- **Do the Job for Less**
- **Hold Schedule**

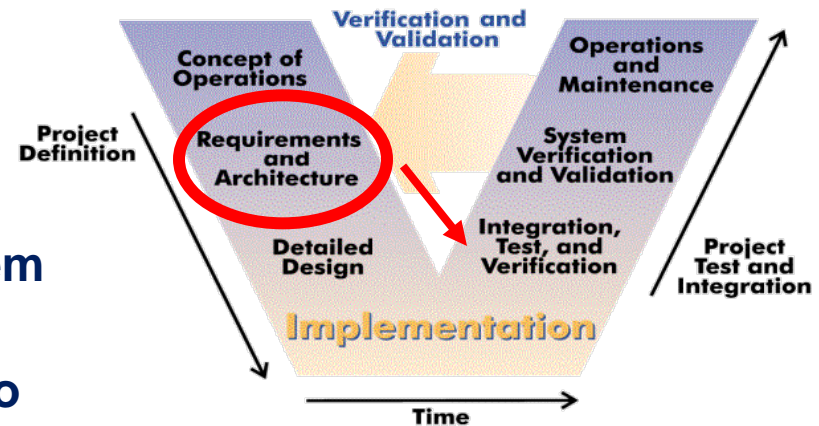


Systems Engineering costs too much – pay now or pay later



- **Approach Taken**

- Used Requirement Management Tool Still Under Development (Cheap)
- Advise Management of Budget Problem Before PDR, and Before CDR
- Plan Based on Management Refusal to Reconsider
- Management Responded by Adding Another Layer of Direct Management



- **Results**

- Do What Was Necessary to Complete PDR
- Downsize Team After PDR (Technical & Lower Level Leadership)
- Management Reconsidered and Increased Budget
- System Completed and Deployed



The COTS Mixture

– Design review success

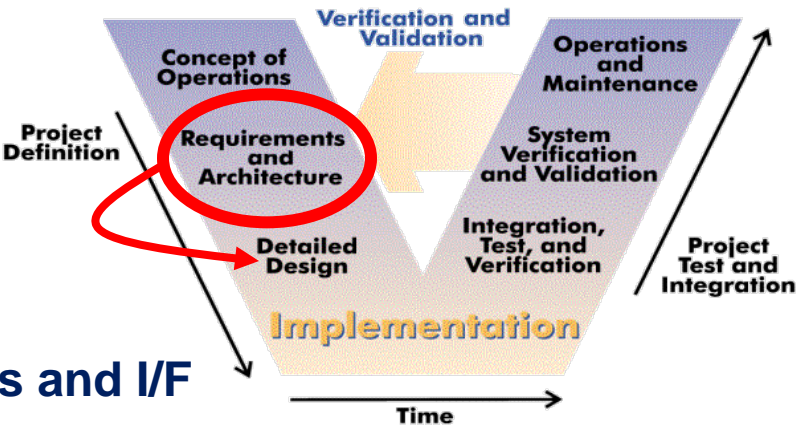


- **Project Background**

- H/S & S/W Large Amount of COTS
- ~ \$500M
- **Systems Engineering**
 - Develop and Control Requirements and I/F
 - Plan for and Conduct Deployment
 - Plan for and Execute Verification
 - Plan & Conduct Sell-off

- **Systems Engineering Challenge**

- Control Strong Subs
- Forceful Customer
- Changing Financial Budgetary Environment





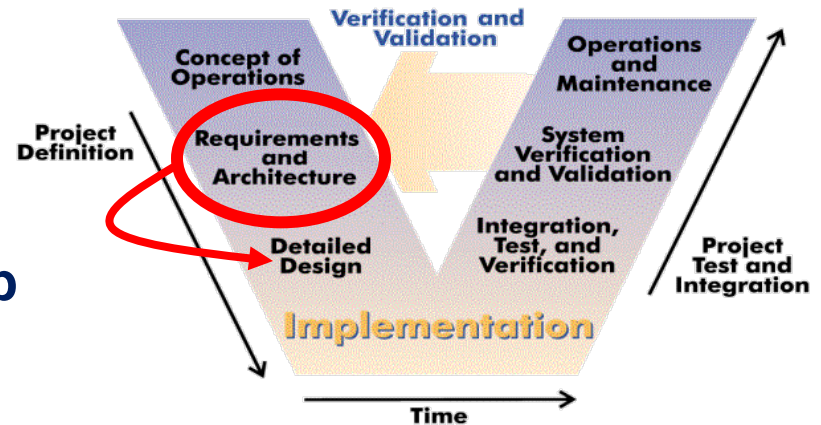
The COTS Mixture

– Design review success



- **Approach Taken**

- **Employ Strong Systems Engineering Process to Develop and Retain Functional Interconnections**
- **Use Unique Approaches to PDR Format That Utilized Strong Subs to Highlight COTS Role**



- **Results**

- **Customer Satisfied**
- **Project Downsized Due to Customer Budget Cuts**
- **Reduced System Deployed Successfully**



The Trust balance

– Control by ICD

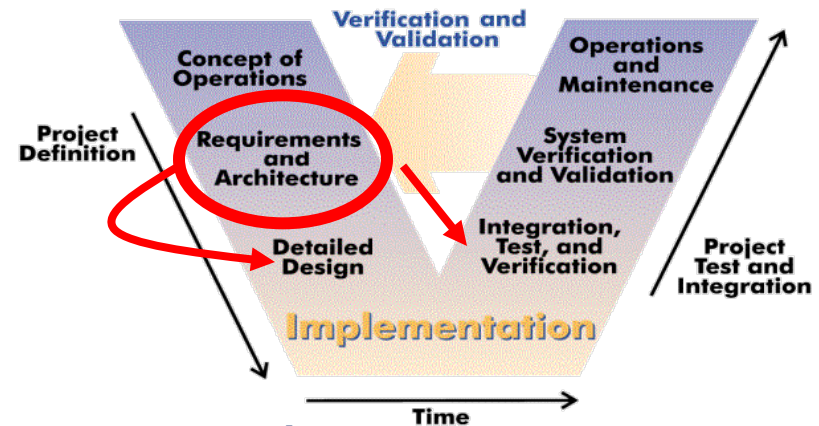


- **Project Background**

- H/S & S/W
- \$500M +
- Systems Engineering
 - Develop and Control Requirements and I/F
 - Plan for and Conduct Deployment
 - Plan for and Execute Verification
 - Plan & Conduct Sell-off

- **Systems Engineering Challenge**

- Requirements De-scoped to Meet Budget Profile
- Resulting Requirements Not Adequate to Design System





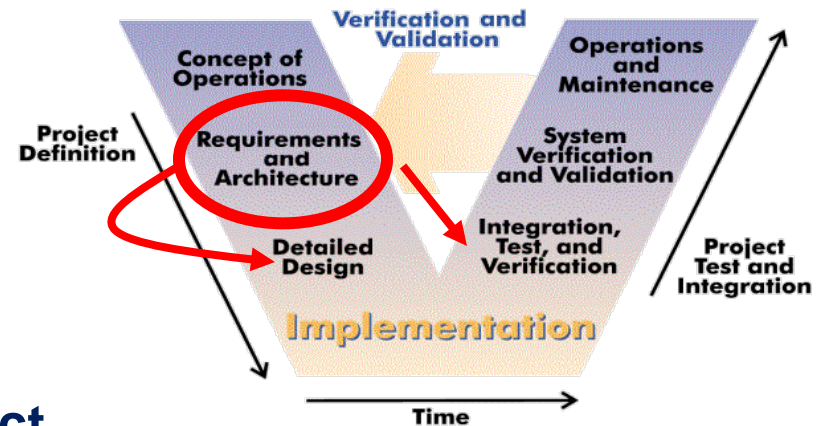
The Trust balance

– Control by ICD



- **Approach Taken**

- Carefully Work Interface Definitions to Contain Requirements
- New I/F Costs Were Allowed Since Not Developed at Contract Start



- **Results**

- Customer and Contractor Controlled Growth and Technical Requirements
- Costs Did Climb, but Under Tight Systems Engineering Control (Contractor and Customer)
- System Completed and Deployed



Summary



- Systems Engineering Process Implementation Varies Across Programs
- The Most Successful Programs Have Knowledgeable Systems Engineers (or Members Who Recognize the Value of Systems Engineering) in Both the Contractor and Customer Teams
- Most of the Successful Programs Employ a Balanced Amount of Systems Engineering

If time permits, the smallest case \$120K



Parting Thoughts

“Systems engineering culture is essential. All the companies agree that there must be a culture of systems engineering and that it must pervade every program, no matter how large or small. ... The prevailing view is that systems engineering is not a phrase, a bumper ticker, an organization, or a job code — systems engineering is a discipline. It is not something that one can have a nodding acquaintance with; nor is it something that one can just be familiar with. It is something one has to own and believe in.” [ref 3]



References

1. Software Engineering Economics, Boehm, Barry W., Englewood Cliff, Prentice Hall, 1981
2. Why systems engineering on telescope?, G. Swart, J. Meiring, Proceedings of SPIE Vol. 4837 Large Ground-based Telescopes, p 166-177, 2003
3. Pre-Milestone A and Early-Phase Systems Engineering: A Retrospective Review and Benefits for Future Air Force Systems Acquisition, National Academy of Science, The National Academies Press 2008