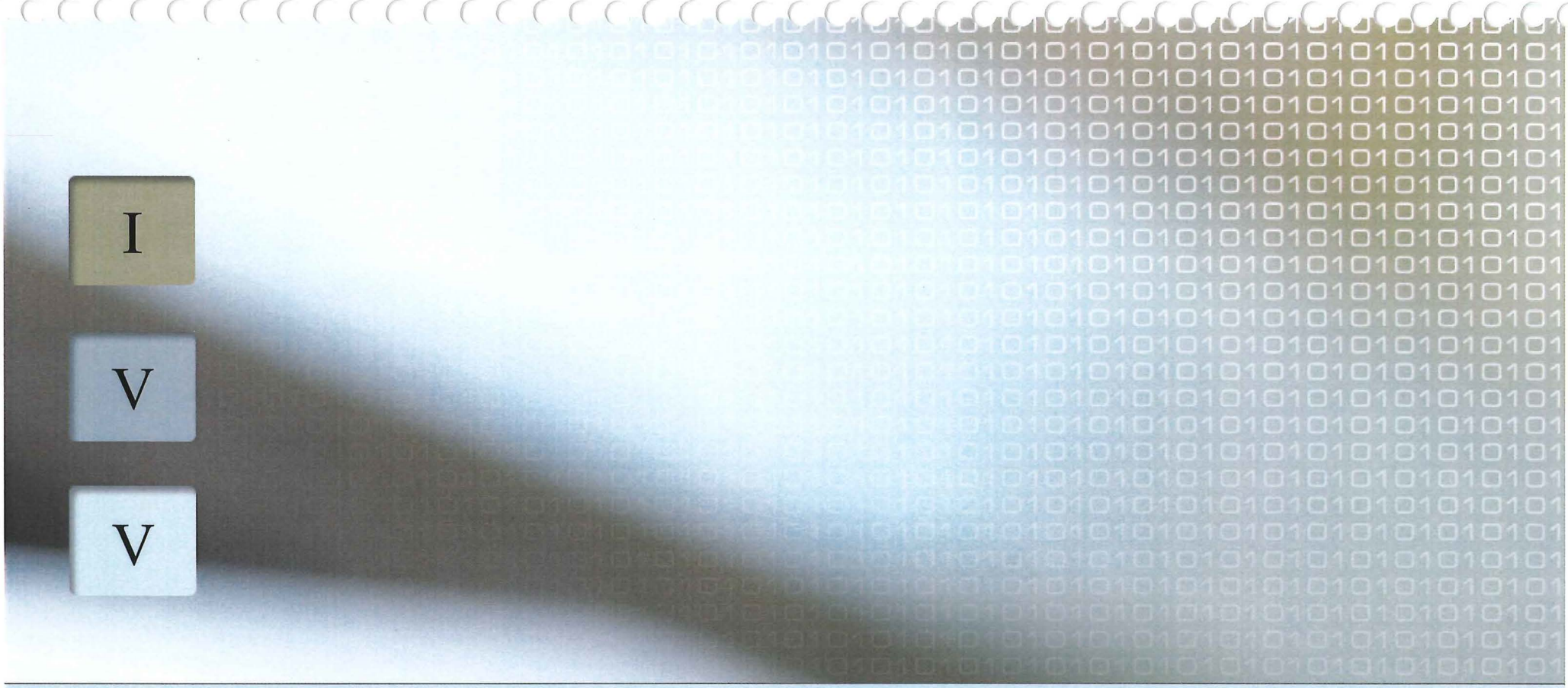

Systems Engineering Integrative Project:
A Minimum Essential IV&V Methodology

Jovanna F. Bunch

December 13, 2006

LMU | LA
Loyola Marymount University

Agenda

- Introduction
- Integrative Project
- Software IV&V
- Requirements
- Trade Studies
- Minimum Essential IV&V Methodology
- IV&V Costs
- Validation of Requirements
- Conclusion

I

V

V

Introduction

- Jovanna Bunch
  - 9.5 years of experience in aerospace industry
    - Northrop Grumman and Smiths Aerospace
  - The Aerospace Corporation
    - Software Assurance Section
  - BS in CS & CE degree
    - Software concentration
  - MS in Systems Engineering degree
    - In process
- Systems Engineering Integrative Project
  - Last step to complete MS degree
  - Designed to demonstrate student's knowledge of systems engineering principles
  - Project based on student's current work experience

I

V

V

Integrative Project Overview

- Software Assurance Section tasked with performing IV&V on contractor and internally-developed software
- Task begins when software is in code and development test phase
  - Code Cost-Constrained Assurance Methodology
- Case study on IV&V ROI
  - Software has fewer defects with full life cycle IV&V rather than with IV&V beginning in code and development test phase
    - Start IV&V sooner rather than later

I

V

V

Integrative Project Goals

- Develop a "minimum essential" IV&V methodology
  - Build off of and improve upon the Code Cost-Constrained Assurance Methodology
    - Extend to cover earlier life cycle phases
  - Focus on complex areas
- Show that the methodology is cost-effective

I

V

V

Systems Engineering Applied

- Demonstrate understanding of the Systems Engineering process
  - Define the methodology beginning with requirements definition and flow-down
  - Show complete process through development to the requirements
  - Conclude with a process for performing IV&V on software in a cost-effective manner

I

V

V

Software IV&V Defined

- A systems engineering process employing rigorous methodologies for evaluating the correctness and quality of the software product throughout the software life cycle
- Focus is on mission and safety critical software and/or extremely costly systems
- Products are independently reviewed, verified, and validated in parallel with development
  - IV&V organization is technically, financially, and managerially independent

I

V

V

Purpose of IV&V

- Evaluate the correctness and quality of a software product
- Early exposure of deficiencies
  - Requirement inconsistencies
  - Design and coding errors
- Build quality into the software as the software life cycle progresses
- Contractor's responsibility for mission success and system safety is never alleviated

I

V

V

Requirements Definition

- Elicitation
  - Interviews with past IV&V participants
  - Extraction from industry standards and documentation
  - Observation of existing methodology
- Assumptions
  - Focus is on the software
    - System and software requirements, software design, and software product
    - Requirements, design, code development, and test phases
  - IV&V Team has access to all information and data necessary to perform IV&V

I

V

V

Requirements

- Integrative Project-specific
  - The Minimum Essential IV&V Methodology shall:
    - Cost less than The Aerospace Corporation "Rule of Thumb" to perform
    - Consist of least amount of work needed to conduct a competent analysis
    - Be an improvement upon the Code Cost-Constrained Assurance Methodology

I

V

V

Requirements cont'd.

- Software IV&V-specific
  - The Minimum Essential IV&V Methodology shall:
    - Provide an objective assessment of the software product throughout the software life cycle
    - Facilitate early detection and correction of software errors
    - Enhance customer insight into the software process and product risks

I

V

V

Trade Studies

- Methods used:
  - Extensive research
    - Standards, journals, articles, organization processes
  - Documents developed
    - Methodology Comparison Spreadsheet
    - Excerpts and Conclusions Table
  - Discussions with Subject Matter Experts
  - Interviews with IV&V Team members
  - Feedback from my superiors

I

V

V

Methodology Comparison Spreadsheet

- Compares software verification and validation activities of:
  - The Aerospace Corporation
  - IEEE Standard 1012
  - IEEE/EIA Standard 12207.0
  - NIST 500-234
  - RTCA/DO-178B
  - NASA
  - United States Navy
- Organized according to development life cycle phases

I

V

V

Methodology Comparison

Aero. Corp.	IEEE Std 1012	IEEE/EIA Std 12207.0	NIST 500-234	RTCA/DO -178B	NASA	US Navy
Conceptual Phase						
	■Concept documentation evaluation ■Criticality Analysis ■HW/SW/User Req'ts Allocation Analysis ■Traceability Analysis ■Hazard Analysis ■Risk Analysis	■Contract Verification ■Project Process V&V	■Concept documentation evaluation		■Proposed Architectural Schema Assessment ■System Req'ts Analysis	■Evaluation of robustness of developers' processes & methodologies

Methodology Comparison cont'd.

Aero. Corp.	IEEE Std 1012	IEEE/EIA Std 12207.0	NIST 500-234	RTCA/DO-178B	NASA	US Navy
Requirements Phase						
	■Traceability Analysis ■SW Req'ts Evaluation ■Interface Analysis ■Criticality Analysis ■System V&V Test Plan ■Acceptance V&V Test Plan ■Configuration Management Assessment ■Hazard Analysis ■Risk Analysis	■System Req'ts Evaluation ■System Req'ts Allocation Evaluation ■Software Req'ts Evaluation	■Software Traceability Analysis ■Software Req'ts Evaluation ■Software Interface Analysis ■Begin test planning	■Verify system req'ts allocated to SW have been developed into SW high-level req'ts that satisfy those system req'ts	■Verify system & SW req'ts ■Verify sufficiency of test plans & acceptance criteria ■Verify sufficiency of testing methods ■Verify correct SW processes are in place	■Analysis of all specs & req'ts ■Verify SW req'ts are entered into a DB ■Req'ts allocation flow-down & verification ■Verification that all code is supported by a req't

Methodology Comparison cont'd.

Aero. Corp.	IEEE Std 1012	IEEE/EIA Std 12207.0	NIST 500-234	RTCA/DO-178B	NASA	US Navy
Design Phase						
	■Traceability Analysis ■SW Design Evaluation ■Interface Analysis ■Criticality Analysis ■Component V&V Test Plan ■Integration V&V Test Plan ■V&V Test Design ■Hazard Analysis ■Risk Analysis	■Verify correctness & traceability of design ■Verify proper design implementation ■Verify design can be derived from req'ts ■Verify design implements safety, security, & other critical req'ts	■SW Design Traceability Analysis ■SW Design Evaluation ■SW Design Interface Analysis ■Verify SW req'ts for required system algorithm & integrity checks ■Coordinate w/SW integration test planning	■Verify high-level req'ts are developed into SW architecture & low-level req'ts that satisfy high-level req'ts	■Verify design satisfies req'ts ■Verify sufficiency of test plans & environments ■Verify design doesn't have characteristics that will cause failure under operational scenarios	

Methodology Comparison cont'd.

Aero. Corp.	IEEE Std 1012	IEEE/EIA Std 12207.0	NIST 500-234	RTCA/DO-178B	NASA	US Navy
Code/Implementation Phase						
■SW & domain familiarization ■Key module identification ■Code quality determination ■Code exec. Traceability analysis ■Produce standard metrics ■Produce analysis artifacts ■Key module peer review	■Traceability Analysis ■Source code & document evaluation ■Interface analysis ■Criticality analysis ■V&V Test Cases & Procedures ■Component V&V test & verification ■Hazard analysis ■Risk analysis	■Code traceability analysis ■Verify proper code implementation ■Verify code is derived from design or req'ts ■Verify code implements critical req'ts	■Source code traceability analysis	■Verify SW architecture & low-level req'ts are developed in source code that satisfies the low-level req'ts & SW architecture ■Verify exec. Object code satisfies SW req'ts	■Verify code reflects the design ■Verify code is correct ■Verify test cases cover SW req'ts & oper. needs ■Verify test cases, expected results, & criteria meet test objective ■Analyze selected code unit test plans & results	■Perform module-level tests using target processor or a simulator

Methodology Comparison cont'd.

Aero. Corp.	IEEE Std 1012	IEEE/EIA Std 12207.0	NIST 500-234	RTCA/DO -178B	NASA	US Navy
Test Phase						
■Test key modules ■Operations-based testing ■Institute configuration control	■Traceability analysis ■Acceptance V&V test procedures ■Integration V&V Test ■System V&V Test ■Acceptance V&V Test ■Hazard analysis ■Risk analysis	■Verify proper integration of SW components & units ■Verify proper integration of HW, SW, & ops. ■Verify proper integration test ■Prepare for test result analysis ■Test stress, boundary, & singular inputs ■Test ability to isolate errors ■User Test ■Validate SW ■Test in selected areas of target env.	■Plan Unit Test, SW Integ. Test, SW System Test ■Trace test design, cases, procedures, & exec. Results ■Confirm anomalies in SW ■Generate test cases & procedures ■Perform test ■Document test results	■Req'ts-based coverage analysis ■Structural coverage analysis ■Low-level testing ■SW integration testing ■HW/SW integration testing	■Verify correct disposition of SW test anomalies ■Validate SW test results vs. acceptance criteria ■Verify tracing & completion of all SW test objectives	■Develop a test for each SW req't & record results ■Use same DB for all SW test activities ■Verify SW req'ts against system specs. ■Verify external interfaces vs. req'ts & HW ■Verify design vs. req'ts ■Verify code vs. design ■Validate build of exec. Code ■Validate integ. Code vs. system specs. & req'ts

Methodology Comparison cont'd.

Aero. Corp.	IEEE Std 1012	IEEE/EIA Std 12207.0	NIST 500-234	RTCA/DO -178B	NASA	US Navy
Operation/Maintenance Phase						
■Fix defects found ■Peer review of fixes ■Regression testing ■Generate final IV&V report	■Installation Configuration Audit ■Installation Checkout ■Hazard Analysis ■Risk Analysis ■Generate final IV&V report		■Installation Configuration Audit ■Installation Test ■Anomaly resolution ■Proposed Change Assessment ■Generate IV&V report		■Verify sufficiency of regression tests	

Methodology Comparison cont'd.

Aero. Corp.	IEEE Std 1012	IEEE/EIA Std 12207.0	NIST 500-234	RTCA/DO-178B	NASA	US Navy
Other						
		■Verify that documentation preparation is timely ■Verify configuration management of documents follows procedure				■Ensure a quality system is in place ■Ensure ability to record & track problems ■Allow for continuous process improvement ■Use CM system to control all versions of documents, DBs, code, & test results ■Ensure ability to auto-report any req't or flow-down test that was not satisfied completely

Excerpts and Conclusions Table

- Excerpts from research materials
- Thoughts concluded
- Demonstrate ability to think through a problem and come to a defensible conclusion
- Helps to provide rationale for activities chosen

I

V

V

	EXCERPT	CONCLUSION
1	A substantial number of the critical faults reported stem from the detection and recording of ambiguous or unclear statements in the requirements specifications and design documents.	With the Code Cost-Constrained Assurance Methodology, these documents are not looked at until the code is already being developed or is developed and going through development test. These documents should be peer reviewed prior to code development.
2	As programs become complex, there can be too many branches to test. The test data should then be selected by examining scenarios of the expected system use and by considering potential failure modes.	Gear the testing portion of the Minimum Essential IV&V Methodology towards scenario-based testing and perhaps include tests that may uncover operational use errors.
3	Lewis states that "the greatest cost-benefit ratio in IV&V comes from requirements verification, wherein defects in requirements can be caught before they begin to ripple forward.	In several of the systems engineering courses that I took at LMU, it was stated repeatedly that it is much cheaper to fix something early in the life cycle than it is to fix it later. My proposed methodology really needs to focus on the requirements.
4	Lewis estimates that an IV&V effort starting at the requirements phase and continuing through deployment would increase the development costs approximately 10 to 18 percent.	Find out what The Aerospace Corporation uses to estimate IV&V costs. Use this scale since the minimum methodology is mainly being developed to improve on the Code Cost-Constrained Assurance Methodology. Develop the proposed methodology so that it can be performed at a lower cost than a full-scale IV&V effort.
5	The Carnegie Mellon Software Engineering Institute (SEI) reports that at least 42-50 percent of software defects originate in the requirements phase.	Further reinforcement to start an IV&V effort at the requirements phase.
6	Code Reviews are seen as unproductive, typically doubling the time it takes to 'craft' the code in the first instance.	I have heard from several members of past IV&V teams that they do not always see the value in line-by-line reviews of the code. It's a time consuming manual process. Remove this activity from the proposed methodology or develop a way for it to be performed in a more productive way.

Minimum Essential IV&V Methodology

- Tailors the intensity of full-scale IV&V
- Mission critical functions are identified and prioritized
  - Resources are focused on these high-risk areas
- Customer provided with a report at conclusion of each step
- Traceability emphasized throughout

I

V

V

6-Step Minimum Essential IV&V Methodology

1	Criticality Analysis	■Conduct analysis of mission critical software functions ■Provide customer with IV&V plan ■Assemble & inform IV&V Team
2	Requirements Analysis	■Verify traceability of system requirements to/from software requirements ■Conduct Requirements Peer Review ■Provide customer with report of findings/recommendations
3	Design Analysis	■Verify traceability of software requirements to/from software design ■Conduct Design Peer Review ■Provide customer with report of findings/recommendations

6-Step Minimum Essential IV&V Methodology cont'd.

4	Testing	■Verify traceability of software requirements to test cases ■Conduct peer review of test plans & procedures ■Perform minimum set of scenario-based test cases ■Test critical functions & requirements violations ■Provide customer with report of findings/recommendations
5	Code Analysis	■Verify traceability of software design to/from software code ■Can be done prior to Step 4 ■Perform static analysis to identify levels of complexity ■Conduct Code Peer Review ■"Red" modules & bugs found in Step 4 ■Provide customer with report of findings and recommendations

6-Step Minimum Essential IV&V Methodology cont'd.

6	IV&V Project Wrap-up	■Provide customer final summary report that includes: ■Activities ■Products, outputs ■Problem reports ■Conclusions ■Recommendations ■Technical Operating Report (TOR)

IV&V Costs

- Whether or not to perform IV&V is based on the cost of IV&V vs. the cost of risks incurred without IV&V
- The Aerospace Corporation "Rules of Thumb"
- SEER-SEM Activity Report

I

V

V

The Aerospace Corporation Rules of Thumb

- Developed in late 80's/early 90's
- Came from cost estimating runs using the Price Systems PRICE-S commercial software model
- Depending on complexity of software
  - 'Low' end projects came in at ~15%
  - 'High' end projects came in at ~35%

Hence the Rule of Thumb

I

V

V

Rules of Thumb

Typical Software Cost Per LOC
(LOC = Equivalent Lines of New Development, Delivered Source Code)

		Average Cost Per LOC (FY06$) (Does not include G&A and Fee)			
		Ground Software		Flight Software	
Cost Range	Software Application	Support	Mission	Spacecraft	Payload
Low	More than ½ Program is: – Data Storage & Retrieval – String Manipulation – Mathematical Operations – Routine Diagnostics	$137± 26	$195 ± 42	$392 ± 53	$448 ± 79
Nominal	Neither Low nor High Characteristics Apply	$216±53	$261 ± 63	$498 ± 79	$594 ± 105
High	More than ½ Program is: – Operating Systems – Interactive Operations – Real time Command & Control – Tight Timing Constraints	$253 ±63	$295 ± 74	$615 ± 105	$658 ± 132

COST FACTORS/RULES OF THUMB

1. **SW Independent Verification and Validation (IV&V): 25% ± 10% of development cost**
2. **Equivalent Lines of Code (ELOC/ESLOC) Conversions:**
   - Reuse = .4 (% re-design) + .25 (% re-code) + .35 (% re-test)
   - Modified = .7 (extensively modified) + .5 (modified) + .3 (minor mods/rehost)
3. **SW Maintenance:**
   - Life Cycle: Maintenance/Development cost ratio (1.0 to 1.5:1) for typical 10 year life cycle.
   - Annual Change Traffic: An estimate of the number of source instructions that undergo change during a typical year, through either addition or modification. Typically, 5 – 15% of development cost per annum.
   - Bug Correction: 3 – 4.5 defects/1 KSLOC for ground software. PRICE-S and SEER-SEM estimates defect density as part of each model's output.
   - Lines Maintained:
     - Military (ground) : 50,000 lines per person year
     - Military (flight) : 25,000 lines per person year

I

V

V

SEER-SEM Activity Report

- Sample represents a generic project with an 'unmanned space environment'
- Breaks down costs per software life cycle phase
- Air Force and NRO use Galorath Inc. SEER-SEM
  - Major customers
- IV&V Rule of Thumb values added

I

V

V

SEER-SEM Estimates

SEER-SEM (TM) Software Schedule, Cost & Risk Estimation Version 7.1.40

Project : Generic
Program : 1.1:

Activity Report

Activity	Schedule Months	Person Months	Person Hours	Cost	IV&V Rule of Thumb 15%	25%	35%
System Requirements Design	2.05	1.68	255	33,931	5,090	8,483	11,876
Cumulative 12/06/06	2.05	1.68	255	33,931			
S/W Requirements Analysis	2.42	4.93	749	99,577	14,937	24,894	34,852
Cumulative 2/18/07	4.47	6.61	1,005	133,507			
Preliminary Design	4.34	15.03	2,285	303,668	45,550	75,917	106,284
Cumulative 6/29/07	8.81	21.64	3,290	437,175			
Detailed Design	4.99	26.33	4,002	531,796	79,769	132,949	186,129
Cumulative 11/29/07	13.80	47.97	7,291	968,970			
Code & Unit Test	1.63	10.58	1,608	213,730	32,060	53,433	74,806
Cumulative 1/17/08	15.43	58.55	8,900	1,182,700			
Component Integrate & Test	4.21	30.30	4,605	612,027	91,804	153,007	214,209
Cumulative 5/24/08	19.64	88.85	13,505	1,794,727			
Program Test	0.47	3.48	529	70,365	10,555	17,591	24,628
Cumulative 6/07/08	20.11	92.33	14,034	1,865,092			
System Integrate Thru OT&E	2.18	25.93	3,942	523,869	78,580	130,967	183,354
Cumulative 8/13/08	22.29	118.27	17,976	2,388,961			
Maintenance	0.00	0.00	0	0			
Cumulative 8/13/08	22.29	118.27	17,976	2,388,961	**358,344**	**597,240**	**836,136**

Validation of Requirements

- The Minimum Essential IV&V Methodology shall:
  - *Cost less than The Aerospace Corporation "Rule of Thumb" to perform*
    - Full-scale IV&V cost estimates shown
    - Proposed methodology is minimized version of full-scale IV&V
    - Can conclude that cost to perform proposed methodology is less
  - *Consist of least amount of work needed to conduct a competent analysis*
    - Consulted with several informed sources
    - Presented methodology to superiors
    - Concurrence received that this is best minimal, yet competent, methodology given the project timeframe

I

V

V

Validation of Requirements cont'd.

- The Minimum Essential IV&V Methodology shall:
  - *Be an improvement upon the Code Cost-Constrained Assurance Methodology (CCAM)*
    - CCAM begins at code and development test phase
    - Sources report that 50% of software defects originate in the requirements phase
    - Proposed methodology begins earlier in the software life cycle than CCAM
      - Errors can be detected as early as in the requirements phase

I

V

V

Validation of Requirements cont'd.

- The Minimum Essential IV&V Methodology shall:
  - *Provide an objective assessment of the software product throughout the software life cycle*
    - The technical, managerial, and financial independence in V&V:
      - Provides an objective assessment
      - Adds a new analytical perspective
      - Brings its own set of tools and techniques
      - Introduces intermediate users who serve as beta testers
      - Enhances testing and discovery of design flaws and coding errors

I

V

V

Validation of Requirements cont'd.

- The Minimum Essential IV&V Methodology shall:
  - *Facilitate early detection and correction of software errors*
    - Determination and categorization of undetected software errors
    - Methodology begins at point of requirements creation
    - Analysis performed in parallel with development
    - Errors reported to customer and flowed down to contractor for correction
  - *Enhance customer insight into the software process and product risks*
    - Reports provided to the customer at conclusion of each of 6 steps
    - Informed early of potential software risks to the program

I

V

V

Lessons Learned

- Difficult to provide a quantitative argument for this methodology
  - Cost-effectiveness of activities is subjective
- True amount of assurance provided is likely to be based on the amount of funds available

I

V

V

Future Work

- Develop "out-of-the-box", innovative solutions
  - Further streamline of the IV&V process
- Innovation Grants
  - Research & Program Development Office
  - IR&D funds for quick-reaction projects
  - Create an ability to rapidly address emerging or current technologies

I

V

V

Conclusion

- Demonstrate knowledge of systems engineering principles
- Ability to apply systems engineering principles to a technical project
- Show ways to improve how IV&V is performed
- Further exploration of this topic

I

V

V


Thank you!