Independent Verification and Validation (IV&V)

15th Annual Systems Engineering Conference

October 2012 - San Diego, CA

Al Florence
The MITRE Corporation

Agenda

- *Attendee Participation*
- Challenges
- Solutions
- IV&V Defined
  - » Verification
  - » Validation
  - » Independence
  - » IEEE IV&V - CMMI® V&V
- IV&V Relationships to other Disciplines (Test, Quality Assurance)
- IV&V; This Presentation
- Overview of Independent Verification & Validation (IV&V)
- Tailoring IV&V
- Examples of IV&V
- Questions/Comments
- References

Attendee Participation

- In relation to developing and acquiring systems and system products
  - Define Verification
  - Define Validation
  - Define Independence

"System" as Defined in this Context

A composite of items (e.g., hardware, software, facilities, personnel, material, services, and techniques) required to perform a complete operational role.

Agenda

- Attendee Participation
- *Challenges*
- Solutions
- IV&V Defined
  - » Verification
  - » Validation
  - » Independence
  - » IEEE IV&V - CMMI® V&V
- IV&V Relationships to other Disciplines (Test, Quality Assurance)
- IV&V; This Presentation
- Overview of Independent Verification & Validation (IV&V)
- Tailoring IV&V
- Examples of IV&V
- Questions/Comments
- References

Challenges to System Developing

- When developing, delivering, and acquiring systems and system products developers and acquirers face many challenges.
- Challenges can exist with many items and activities:
  - Cost
  - Schedule
  - Technical
  - Management
  - Programmatic
  - Process
  - Quality
  - Others?

Challenges

- Consequences may be numerous if challenges not mitigated
  - » Cost overruns
  - » Late deliveries
  - » Technically inadequate
  - » Mismanagement
  - » Programmatic difficulties
  - » Lack of sound process
  - » Irate customer
  - » Canceled project
  - » Others?

Agenda

- Attendee Participation
- Challenges
- *Solutions*
- IV&V Defined
  - » Verification
  - » Validation
  - » Independence
  - » IEEE IV&V - CMMI® V&V
- IV&V Relationships to other Disciplines (Test, Quality Assurance)
- IV&V; This Presentation
- Overview of Independent Verification & Validation (IV&V)
- Tailoring IV&V
- Examples of IV&V
- Questions/Comments
- References

Solutions

- There are many solutions for an organization to mitigate these challenges
  - Proper project planning
  - Adequate budgets
  - Adequate schedules
  - Proper requirements development and management
  - Proper project management
  - Program monitoring and control
  - Contract tracking and oversight
  - Product evaluation
  - Performance management
  - Risk management
  - Quality assurance
  - Configuration Managment
  - ***Independent Verification and Validation (IV&V)***
  - Others?

Solutions

- This presentation will focus on IV&V
  - » IV&V alone will not guarantee mitigation of all challenges
  - » But, can go a long way in solving many

Agenda

- Attendee Participation
- Challenges
- Solutions
- *IV&V Defined*
  - » Verification
  - » Validation
  - » Independence
  - » IEEE IV&V - CMMI® V&V
- IV&V Relationships to other Disciplines (Test, Quality Assurance)
- IV&V; This Presentation
- Overview of Independent Verification & Validation (IV&V)
- Tailoring IV&V
- Examples of IV&V
- Questions/Comments
- References

IV&V Defined

- Independent Verification and Validation (IV&V) is often misunderstood and misapplied
- In many cases a distinction is not made between verification and validation; V&V is treated as one activity
  - » Often is heard: “We are doing IV&V on a document”
  - » What was typically being conducted was an independent review of a document
- IV&V is an activity across the life cycle
  - » Not an isolated effort performed at random intervals
  - » Nor performed as necessary once on one or few components or items of interest

IV&V Defined

- From IEEE Standard 1012 2004, IEEE Standard for Software Verification and Validation; IEEE Computer Society
  - » **Verification: (A)** The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. **(B)** The process of providing objective evidence that the software and its associated products conform to requirements (e.g., for correctness, completeness, consistency, accuracy) for all life cycle activities during each life cycle process (acquisition, supply, development, operation, and maintenance); satisfy standards, practices, and conventions during life cycle processes; and successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities.
    - Answers the question: are we building the products right ?

NOTE: Although not specifically mentioned this author interprets this at also investigating the processes that are used to develop and manage the components and products.

IV&V Defined

- **Validation: (A)** The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. **(B)** The process of providing evidence that the software and its associated products satisfy system requirements allocated to software at the end of each life cycle activity, solve the right problem (e.g., correctly model physical laws, implement business rules, use the proper system assumptions), and satisfy intended use and user needs.
  - Answers the question: are we building the right products?

NOTE: Although not specifically mentioned this author interprets this at also investigating the processes that are used to develop and manage the components and products.

IV&V Defined

- **Independent Verification and Validation**
  - Performed by an organization that is:
    - Technically
    - Managerially
    - Financially

    Independent of the development organization
  - Supports objectivity

IV&V Defined

- IV&V Objectives:
  - Assess software and system products and processes during life cycle
  - Facilitate early detection and correction of errors
  - Reduce effort to remove faults, via early detection
  - Demonstrate hardware, software, system requirements are complete, accurate, consistent, testable
  - Enhance management insight into process and product risk
  - Support the life cycle processes to ensure compliance with program performance, schedule, and cost requirements
  - Enhance operational correctness and product maintainability

IV&V Defined

- Verification ensures that the end products are developed correctly with a focus on the products, the process, and interim steps to achieve the end result and that the requirements are the right ones for the customers' needs.
- Validation, on the other hand, ensures that the correct products are developed with the focus on proving that the specified requirements are satisfied.
- Implementation of independence can be accomplished by one or more independent organizations.
- The IV&V organization can be supported/augmented by other independent entities such as an Federally Funded Research and Development Center (FFRDC) and/or the program office.

V&V Further Defined

V&V is an extension of program management and systems engineering that employs a rigorous methodology to identify objective data and conclusions to provide feedback about software/systems quality, performance, and schedule to the development organization.

This feedback consists of anomaly resolutions, performance improvements, and quality improvements not only for expected operating conditions, but also across the full spectrum of the system and its interfaces.

Early feedback results allow the development organization to modify the software/system products in a timely fashion and thereby reduce overall project and schedule impacts.

Without a proactive approach, anomalies and associated software/system changes are typically delayed to later in the program schedule, resulting in greater program costs and schedule delays.

IEEE Std 1012 2004

V&V Defined

Extracted from CMMI® - for Development, Version1.3

- **Verification**
  - The purpose of Verification (VER) is to ensure that selected work products meet their specified requirements. In other words, verification ensures that "you built it right."
- **Validation**
  - The purpose of Validation (VAL) is to demonstrate that a product or product component fulfills its intended use when placed in its intended environment. In other words, validation ensures that "you built the right thing."

NOTES:
- The IEEE and the CMMI® definitions are somewhat different but accomplish the same results
- For this presentation we will use the IEEE definitions

V&V CMMI®

- Extracted from CMMI® v1.3 - Validation
  - Specific Goal 1 - Prepare for Validation
    - SP 1.1 Select Products for Validation
    - SP 1.2 Establish the Validation Environment
    - SP 1.3 Establish Validation Procedures and Criteria
  - Specific Goal 2 - Validate Product or Product Components
    - SP 2.1 Perform Validation
    - SP 2.2 Analyze Validation Results

V&V CMMI®

- Extracted from CMMI® v1.3 - Verification
  - Specific Goal 1 - Prepare for Verification
    - SP 1.1 Select Work Products for Verification
    - SP 1.2 Establish the Verification Environment
    - SP 1.3 Establish Verification Procedures and Criteria
  - Specific Goal 2 - Perform Peer Reviews
    - SP 2.1 Prepare for Peer Reviews
    - SP 2.2 Conduct Peer Reviews
    - SP 2.3 Analyze Peer Review Data
  - Specific Goal 3 - Verify Selected Work Products
    - SP 3.1 Perform Verification
    - SP 3.2 Analyze Verification Results

Agenda

- Attendee Participation
- Challenges
- Solutions
- IV&V Defined
  - » Verification
  - » Validation
  - » Independence
  - » IEEE IV&V - CMMI® V&V
- *IV&V Relationships to other Disciplines (Test, Quality Assurance)*
- IV&V; This Presentation
- Overview of Independent Verification & Validation (IV&V)
- Tailoring IV&V
- Examples of IV&V
- Questions/Comments
- References

IV&V Relationships to Other Disciplines

- Although IV&V conducts some or similar activities as:
  - The Test Organization, and
  - The Quality Assurance Organization
- IV&V conducts them with a different filter

IV&V vs. Test

- Several types of tests can occur for product acquisition and product development
  - Component or Unit tests
  - Commercial-off-the-shelf (COTS) tests
  - Formal Tests
    - Functional Tests
    - Integration Tests
    - System Acceptance Tests
    - Operational Tests
    - Certification Tests

IV&V vs. Test

- Implementers conduct their own tests at the unit and component level to ensure that the units and components meet their documented and approved design
  - » Indirectly ensures that allocated requirements have been met
  - » Requirements' allocation should be reflected in the design
- Development test team, separate from the implementation team, conduct tests at the formal level to ensure that requirements have been met and that interfaces are correct and that products are ready for production
  - » Also called Validation

IV&V Relationships to Other Disciplines

IV&V vs. Test

- IV&V, independent of the development and test teams, objectively evaluate that:
  - Units and Components meet their documented design
  - Products meet requirements
  - Interfaces are correct
  - Products are ready for production
  - Again, IV&V is performed with a different filter completely separate and independent of the organization developing the system

IV&V vs. Quality Assurance

- Quality Assurance (QA) objectively assures:
  - Technical and programmatic products are developed according to standards, plans, procedures that govern their formats and contents
    - QA typically does not evaluate the technical adequacy of products
  - Technical and programmatic processes are executed according to standards, plans, procedures and process descriptions
  - QA provides objectivity by reporting to management above the project and/or program level

IV&V vs. Quality Assurance

- IV&V objectively evaluates
  - Technical and programmatic products are developed according to standards, plans, procedures and technical requirements that govern their formats, contents and functionality
    - IV&V evaluate the technical adequacy of products
  - Technical and programmatic processes are executed according to standards, plans, procedures and process descriptions
  - IV&V provides objectivity by reporting independently to an organization that is separate from the development organization

Agenda

- Attendee Participation
- Challenges
- Solutions
- IV&V Defined
  - » Verification
  - » Validation
  - » Independence
  - » IEEE IV&V - CMMI® V&V
- IV&V Relationships to other Disciplines (Test, Quality Assurance)
- *IV&V; This Presentation*
- Overview of Independent Verification & Validation (IV&V)
- Tailoring IV&V
- Examples of IV&V
- Questions/Comments
- References

IV&V; This Presentation

- Over the past several years this author has been involved in supporting DoD and civil agencies in defining, planning, and executing IV&V activities.
- Each engagement has had a different focus and varying amount of involvement.
- An overview of IV&V is first presented
- Next tips of tailoring IV&V
- Real project examples from some IV&V engagements are presented
  - » Some examples focus on verification
  - » Others on validation
  - » Others on both

Agenda

- Attendee Participation
- Challenges
- Solutions
- IV&V Defined
  - » Verification
  - » Validation
  - » Independence
  - » IEEE IV&V - CMMI® V&V
- IV&V Relationships to other Disciplines (Test, Quality Assurance)
- IV&V; This Presentation
- *Overview of Independent Verification & Validation (IV&V)*
- Tailoring IV&V
- Examples of IV&V
- Questions/Comments
- References

Independent Verification and Validation

- A client did not understand why it took so long and cost so much when its IV&V agent conducted IV&V of developed and delivered products
- The Program Office did not seem to understand what IV&V consisted of
- The Program Office asked this author to produce a white paper defining IV&V
  - » And to conduct a study on the IV&V and Certification activities and to make recommendations to correct problems, reduce cycle time and costs
    - > This will be presented later
- The following presentation summarizes the white paper
- This portion of the presentation is presented as per lifecycle phases
  - » This does not imply that IV&V should be a waterfall activity
  - » IV&V can be applied to any lifecycle paradigm appropriately tailored

Planning Phase

- During the planning phase IV&V activities are planned
  - Identify key IV&V stakeholders
  - Identify IV&V lifecycle activities
    - Balance with other project activities
  - Identify and acquire IV&V resources
  - Identify and acquire IV&V budget
  - Tailor IV&V to the scope of the project
    - More on this later
  - Identify and select IV&V staff
  - Develop IV&V Plan
  - Review and approve IV&V Plan

Independent Verification

- Independent Verification is conducted throughout all life cycle phases: requirements, design, implementation, integration, test, and production
- Activities of verification may include:
  - Reviews and audits
    - Product reviews
    - Peer reviews
    - Process audits
  - Analysis
  - Prototypes
  - Simulations

Planning Phase

- During the planning phase some Independent Verification is conducted against Program Planning
  - » Development Schedules
  - » Project Management Plans
  - » Quality Assurance Plans
  - » Configuration Management Plans
  - » Risk Management Plans
  - » Software Development Plans
  - » System Engineering Plans
  - » Hardware Development/Acquisition Plans
  - » COTS Plans

Requirements Phase(s)

- The process of evaluating a system or its components to determine whether the products of the requirements phase satisfy the conditions imposed at the start of that phase with the main focus on requirements.
- Products and their related activities may include:
  - » Requirements Specifications
  - » Interface Specifications
  - » Test Plans
  - » Updated Plans and schedules form the Planning Phase

Independent Verification

Design Phase

- The process of evaluating a system or its components to determine whether the products of the design phase satisfy the conditions imposed at the start of that phase with the main focus on design.
- Products and their related activities may include:
  - » Design Documents
  - » Interface Design
  - » Updated
    - > Requirements Specifications
    - > Interface Specifications
    - > Development Schedules
    - > Test Plans
    - > Plans from the Planning Phase

Implementation Phase

- The process of evaluating a system or its components to determine whether the products of the implementation phase satisfy the conditions imposed at the start of that phase with the main focus on implementation.
- Products and their related activities may include:
  - » Code / HW development / COTS procurement
  - » Unit Test Plans
  - » Unit Test Procedures
  - » Unit Test Reports
  - » Updated
    - > Design Documents
    - > Interface Design
    - > Requirements Specifications
    - > Interface Specifications
    - > Development Schedules
    - > Test Plans
    - > Plans from the Planning Phase

Independent Verification

Integration Phase

- The process of evaluating a system or its components to determine whether the products of the implementation phase satisfy the conditions imposed at the start of that phase with the main focus on integration.
- Products and their related activities may include:
  - » Integration Plans
  - » Integration Procedures
  - » Integration Witness
  - » Integration Reports
  - » Updated
    - > Hardware Components
    - > Code
    - > Unit Test Plans
    - > Unit Test Procedures
    - > Unit Test Reports
    - > Design Documents
    - > Interface Design
    - > Interface Design
    - > Requirements Specifications
    - > Interface Specifications
    - > Development Schedules
    - > Test Plans
    - > Plans from the Planning Phase

Formal Test Phases

- The process of evaluating a system or its components to determine whether the products of the test phases satisfy the conditions imposed at the start of those phases with the main focus on test.
- Formal test phases may include:
  - » Functional Tests
  - » System Integration Test
  - » Acceptance Test
  - » Operational Tests
  - » Certification Tests

Independent Verification

Formal Test Phase

- Products and their related activities may include: :
  - » Test Plans
  - » Test Descriptions
  - » Test Procedures
  - » Test Witness
  - » Test Reports
  - » Updated
    - > Integration Plans
    - > Integration Procedures
    - > Integration Reports
    - > Hardware Components
    - > Code
    - > Unit Test Plans
    - > Unit Test Procedures
    - > Unit Test Reports
    - > Design Documents
    - > Interface Design
    - > Requirements Specifications
    - > Interface Specifications
    - > Development Schedules
    - > Test Plans
    - > Plans from the Planning Phase

Independent Validation

- The process of evaluating a system or its components during, or at the end of the development process, to determine whether they satisfy their specified requirements.
- Planning for Independent Validation activities starts at the requirements phase and continues throughout other phases.
  - This includes validating that requirements are traceable and testable
- Products and their related activities may include:
  - Validation Plans
  - Validation Descriptions
  - Validation Procedures
  - Code
  - Validation Conduct
  - Validation Reports

Independent Validation may or may not include the actual execution of tests, more on this later

Independent Verification & Validation

Operational Phase

- The same activities that are executed, from a lifecycle phase perspective, may need to be conducted during operation
  - Correcting defects
  - Correcting requirements
  - Implementing changes
    - Enhancements
    - New functionality
    - Modernization
    - Legacy upgrades
    - Integration into system-of-systems
    - Etc.
- These may require repeating some or all development lifecycle phases or activities
  - May require the execution of IV&V as appropriate

Agenda

- Attendee Participation
- Challenges
- Solutions
- IV&V Defined
  - » Verification
  - » Validation
  - » Independence
  - » IEEE IV&V - CMMI® V&V
- IV&V Relationships to other Disciplines (Test, Quality Assurance)
- IV&V; This Presentation
- Overview of Independent Verification & Validation (IV&V)
- *Tailoring IV&V*
- Examples of IV&V
- Questions/Comments
- References

Tailoring IV&V

- The scope of the IV&V effort is dependent on several factors such as:
  - Cost
  - Size
  - Schedule
  - Technical Complexity
  - Criticality
  - Security
  - Safety
  - Risk
  - Team complexity
  - Organizational complexity
  - Requirements maturity
  - Level of integration
- IV&V can be very costly, in some cases costs can be as much as those of the development effort for complex and critical systems.
- IV&V factors need to be analyzed resulting in a cost effective IV&V effort that is appropriately tailored to the scope of the application.

Tailoring IV&V

- By conducting an analysis of the scope of the application in relation to: cost, size, schedule, complexity, criticality, security, safety, risk, team complexity, organizational complexity, requirements maturity, and level of integration an IV&V effort can be appropriately tailored to the scope of the application.
  - » Full scale IV&V would include all the activities described.
  - » Tailoring may include the conduct of all or some of the activities excluding the Independent Validation.
  - » It could involve only Independent Validation.
  - » Any variation of the 2 above.
  - » IV&V may be tailored by only including items that are:
    - > Time critical and/or
    - > Safety critical and/or
    - > Security critical

Tailoring IV&V

- Validation could include full scale independent testing
- Validation could include oversight of contractor testing
  - No independent testing
- Verification could include items that are:
  - Identified as high risk
  - Selected by Management
  - Randomly selected
  - A percent of all items such as 10%, 20%, 30%, etc.
  - Only deliverables
- IV&V could include only technical requirements
- IV&V could be independent oversight of some or all contractor's V&V activities

Risk Based IV&V

Integrity Levels

Description	Levels
♦ Software element must execute correctly or grave consequences (loss of life, loss of system, economic or social loss) will occur. » No mitigation is possible – High level of IV&V	4
♦ Software element must execute correctly or the intended use (mission) of the system/software will not be realized, causing serious consequences (permanent injury, major system degradation, economic or social impact) » Partial to complete mitigation is possible – Medium level of IV&V	3
♦ Software element must execute correctly or an intended function will not be realized, causing minor consequences. » Complete mitigation possible – Moderate level of IV&V	2
♦ Software element must execute correctly or intended function will not be realized, causing negligible consequences. » Mitigation not required – IV&V not needed	1

IEEE Standard 1012 2004

Agenda

- Attendee Participation
- Challenges
- Solutions
- IV&V Defined
  - » Verification
  - » Validation
  - » Independence
  - » IEEE IV&V - CMMI® V&V
- IV&V Relationships to other Disciplines (Test, Quality Assurance)
- IV&V; This Presentation
- Overview of Independent Verification & Validation (IV&V)
- Tailoring IV&V
- *Examples of IV&V*
- Questions/Comments
- References

IV&V Examples

- *Extreme Example of IV&V*
- Independent Verification of the Requirements Specification Process
- Independent Validation of Requirements
- Evaluation of an IV&V Process
  - » Recommendations to the Program Office
- Application of IV&V with Limited Resources

IV&V Examples
Extreme Example of IV&V

- Extreme IV&V can be as expensive and time consuming as the development effort
- An example of this is Nuclear Safety Cross Check Analysis (NSCCA)
  - » Conducted by an organization independent of the development organization (usually a different contractor)
  - » Purpose is to identify and eliminate defects related to nuclear vulnerabilities
    - > The reentry vehicle (RV), with a nuclear war head, shall hit the intended target
      - Not New York or Washington D.C.

IV&V Examples
Extreme Example of IV&V

- Corporation XX had developed the system and software for the targeting flight program for an nuclear weapon, reentry vesicle (RV)
- Corporation YY was the NSCCA contractor conducting IV&V on the development contractor's software
  - The software development approach was independently verified
  - Intermediate products were Independently verified

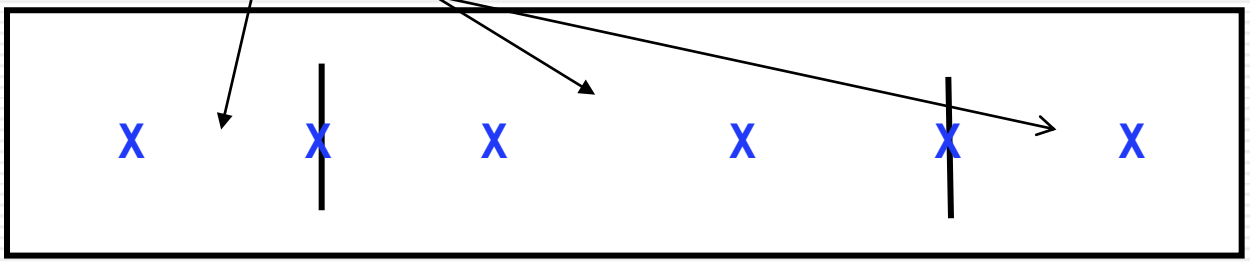
IV&V Examples
Extreme Example of IV&V

- The developed code was independently verified and validated
  - » This author was assigned to the team that conducted this effort
  - » Independent of, and in addition to, the development and test program
  - » A simulated flight test environment was developed reflecting all RV flight scenarios
  - » All requirements were validated
  - » Code was validated, all possible:
    - > functions
    - > performance
    - > branches
    - > inputs
    - > outputs
    - > conditions
    - > functions
    - > etc.

    were execute and tested

IV&V Examples
Extreme Example of IV&V

- In bound, on bound, and out of bound parameters and anomalies were extensively tested
  - Nominal
  - Off nominal



- Operational and test code was in assembly language
- Code evaluation and test execution often at machine language level

IV&V Examples
Extreme Example of IV&V

- Independent Validation
  - » Was conducted over a two year period
  - » Involved 12 testers and a manager
  - » Uncovered several nuclear vulnerabilities
  - » Cost $$$$$$$$$$$
- Independent Verification
  - » No data, author not involved
- Assured RV would hit intended target
  - » enemy ICBM silo

IV&V Examples

- Extreme Example of IV&V
- *Independent Verification of the Requirements Specification Process*
- Independent Validation of Requirements
- Evaluation of an IV&V Process
  - » Recommendations to the Program Office
- Application of IV&V with Limited Resources

Audience Participation

What is wrong with this requirement?

After the system receives the Validation file, the system shall:

- notify the individual about acceptance or rejection.
- the acceptance file must contain the name and ZIP code of the individual.
- rejected validation request must include the Reason Code.

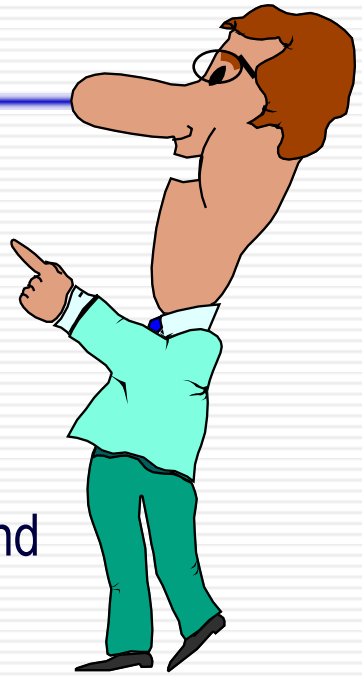
Independent Verification of Specified Requirements

- Verification *(Are we building the product right ?)*
  - The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.
- These examples show a critique of specified requirements:
  - Against critical attributes that requirements need to comply with, and
  - Their re-specification.

IV&V Examples
Independent Verification of Specified Requirements

- Requirements Phase – The process of evaluating a system or its components to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase with a focus on requirements.
- Products to verify may include:
  - *Requirements Specifications*
  - Interface Specifications
  - Development Schedules
  - Development Plans
  - Quality Assurance Plans
  - Configuration Management Plans
  - Risk Management Plans
  - Test Plans

Requirements' Challenges



- Some of the biggest challenges faced by engineers are those of requirement definition, specification, analysis, validation and verification.
- In many documents of requirements the requirements are ambiguous and inconsistent.
- They may not be uniquely identified making them untraceable and difficult to test.
- In many cases they are not specified at the correct level:
  - » Too much detail (design implications)
  - » Not enough detail (conceptual without substance)

If these challenges are mitigated the risk of developing systems that do not satisfy their requirements will be reduced.

Nature of requirements - *what are they?*

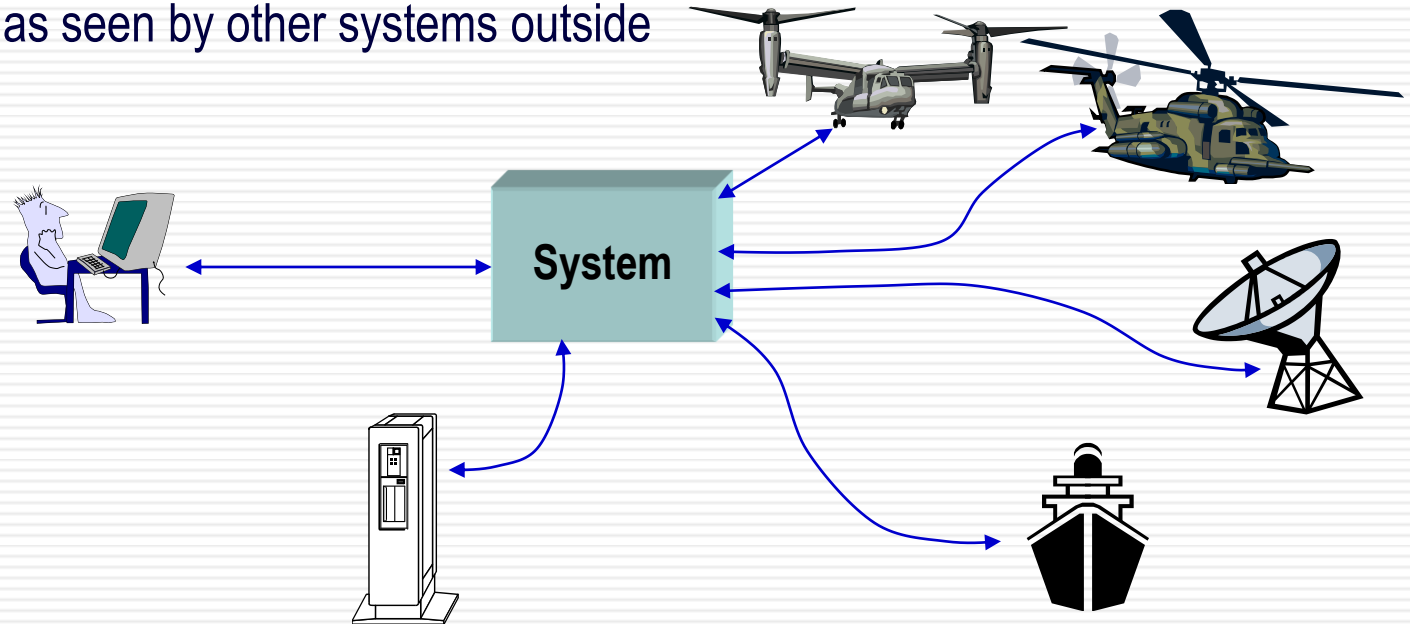
- IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications:

"A requirement specifies an externally visible function or attribute of a system"

  - » We can see inputs and the outputs, but not what happens inside

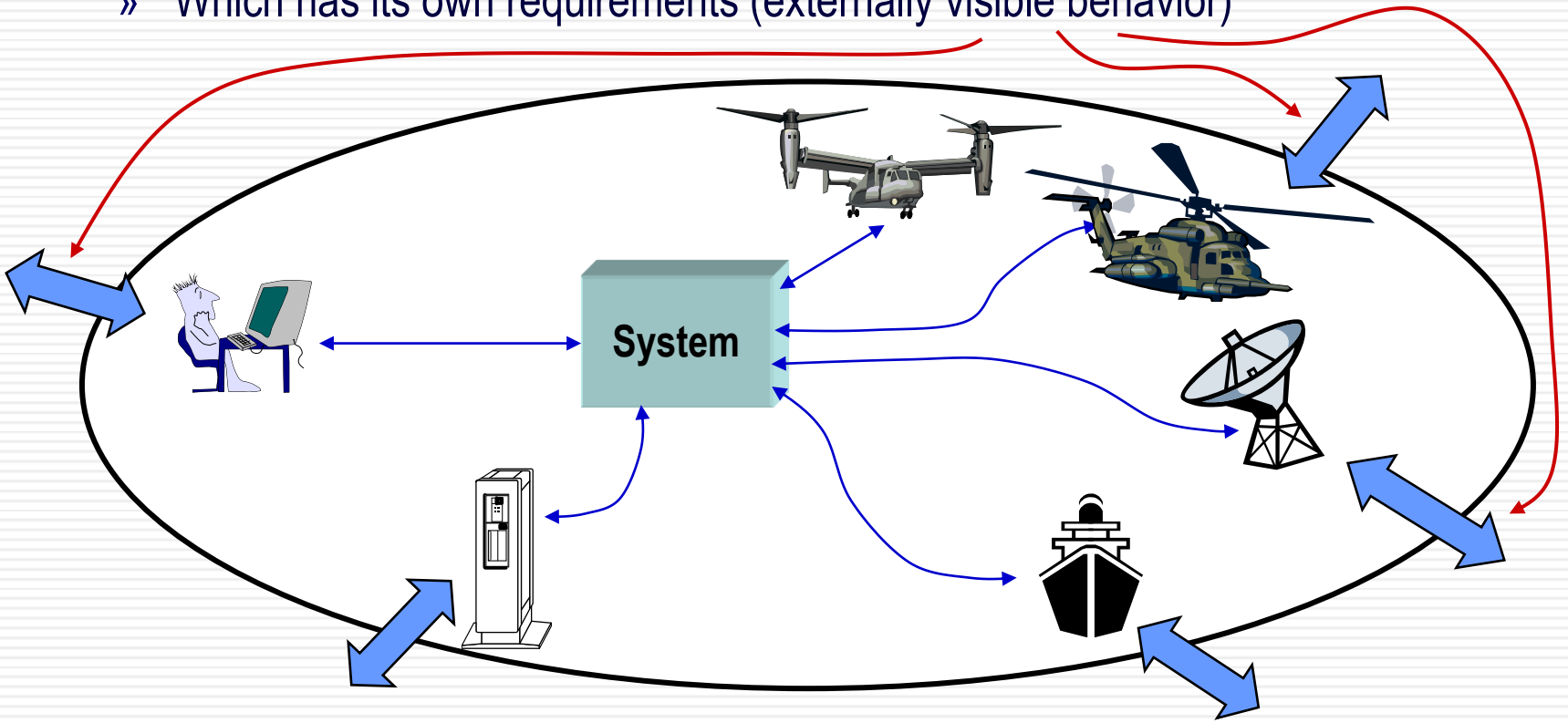
- For any product (SW, HW, total system), the behavioral requirements for that product specify its externally visible behavior

  - » as seen by other systems outside



Nature of requirements - *what are they?*

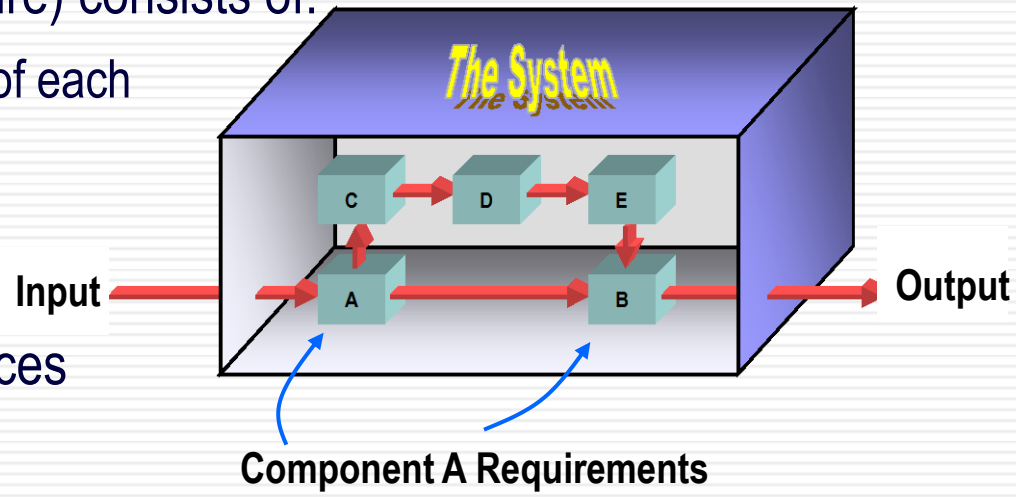
- But each such system could be part of a larger system
  - » Which has its own requirements (externally visible behavior)



For the rest of this briefing, "requirement" denotes externally visible behavior

Context of requirements

- All requirements are defined in context of a specific component (e.g., black box)
  - » Which may consist of additional constituent components (e.g., subsystem, modules,...)
  - » Hence there are multiple levels of requirements based on level of component
    - > System level, subsystem level, software configuration item (SCI) level, component level, software unit level,...
- Component design (its architecture) consists of:
  - » The requirements for behavior of each constituent component
  - » The interrelationships between the components
- Interaction of components produces the behavior of parent component



Independent Verification of Specified Requirements

- The following examples show a method of verifying requirements during their specification.
- Specification in this context means documenting/writing the requirements.
- Many methods of verifying requirements exist:
  - The verification method here is to verify that the requirements are specified in a fashion that satisfies the needs of the stakeholders (users, developers, customers).
  - These needs have been established in prior activities such as:
    - conceptual design / system level requirements analysis
    - request for proposal
    - proposals
    - planning, etc.

Independent Verification of Specified Requirements
Criteria for Specifying a Good Requirement

The following are some critical attributes that requirements must adhere to.

Used to critique requirements

- Completeness: Requirements should be complete.

  They should reflect system objectives and specify the relationship between the software and the rest of the subsystems.

- Traceability: Each requirement must be traceable to some higher-level source, such as a system-level requirement.

  Each requirement should also be traced to lower level design and test abstractions such as high-level and detailed-level design and test cases.

- Testability: All requirements must be testable in order to demonstrate that the software end product satisfies its requirements.

  In order for requirements to be testable they must be specific, unambiguous, and quantitative whenever possible. Avoid negative, vague and general statements.

Independent Verification of Specified Requirements

Criteria for Specifying a Good Requirement

- Unambiguity: In order for requirements to be understood, verified and validated they must be specific and unambiguous.

  Avoid vague, general statements

- Consistency: Requirements must be consistent with each other; no requirement should conflict with any other requirement.

  Requirements should be checked by examining all requirements in relation to each other for consistency and compatibility.

- Feasibility: Each requirement must be feasible to implement.

  Requirements that have questionable feasibility should be analyzed during requirements analysis to prove their feasibility,

- Unique identification: Uniquely identifying each requirement is essential if requirements are to be traceable and difficult to test.

  Uniqueness also helps in stating requirements in a clear and consistent fashion.

Independent Verification of Specified Requirements

Criteria for Specifying a Good Requirement

- Design Free: Software requirements should be specified at a requirements level not at a design level.

  The approach should be to describe the software requirement functionally from a system (external) point of view, not from a software design point-of-view, i.e. describe the system functions that the software must satisfy. Some requirements may have design embedded due to constraints placed on them by the system, interfaces or legacy.

- Use of “shall” and related words: In specifications, the use of the word "shall" indicates a binding provision.

  Binding provisions must be implemented by users of specifications. To state non-binding provisions, use "should" or "may". Use "will" to express a declaration of purpose (e.g., "The Government will furnish..."), or to express future tense. MIL-STD

Note: Methods other than the use of “shall” can be used to specify requirements such as using a matrix with a column for requirements and another column for comments or italics or underlines for comments or requirements.

Independent Verification of Specified Requirements

- A Civil agency, while modernizing its information systems, reverse-engineered requirements.
- With domain knowledge of the application, several teams were involved.
  - They represented:
    - the users
    - the contractors
    - the acquisition organization
- This author was assigned as a consultant to guide the teams in the proper specification of requirements.
- The examples presented show some of the requirements:
  - as initially specified by the teams
  - next a critique of the requirements by this author
  - finally the re-specified requirements based on the critique

Background

- It needs to be noted that requirements do not “live alone”
  - » They depend on other requirements and/or
  - » on clarifying comments to present a complete view of the functionality associated with a related set of requirements.
- A related set of functional requirements may be introduced with a preamble describing the capability of the functional set.
  - » The preamble does not itself establish requirements; this is done later in the requirements’ specifications.
- Some requirements may be amplified with clarifying comments which are, again, not part of the requirements, but add understandability.

Background (cont.)

- Some requirements are documented sequentially with the requirements stated first setting the "stage" for the following requirements which add more and more capability.
  - » The later stated requirements depend on the earlier requirements to complete their functionally.
  - » An example may be the use of the word "processing". If the processing of a functional set of related requirements has been described in earlier requirements the later requirements may amplify and/or reference the processing without having to restate the processing.
- This is the case in the following examples; they have been extracted from a larger set of functionally related requirements and may not present a complete picture of the entire set.
- If a single requirement was to be a complete picture of a complex capability, one requirement would have to describe the entire capability making it extremely complex and difficult to understand, implement, and test.

Independent Verification of Specified Requirements
Example 1

Initial specification:

Software will not be loaded from unknown sources onto the system without first having the software tested and approved.

Critique:

- If it's tested and approved, can it be loaded from an unknown source?
- If the source is known, can it be loaded without being tested and approved?
- This requirement is ambiguous and stated as a negative requirement, which makes it difficult to implement and test.
- A unique identifier is not provided, which makes it difficult to trace.
- The word "shall" is missing.
- This is a programmatic requirements that should not be specified along technical requirements but should be elsewhere such as in the Statement of Work (SOW)

Re-specification:

3.2.5.2 Software shall be loaded onto the operational system only after it has been: a) tested; and b) approved..

Example 2

Initial specification:

3.2.5.7 The system shall process two new fields (provides production count balancing info to states) at the end-of-state record.

Critique:

- This requirement cannot be implemented or tested.
- It is incomplete. What are the two new fields?
- “Info” should be spelled out.

Re-specification:

3.2.5.7 The system shall provide the following data items (provides production count balancing information to states) at the end-of-state record:

a. SDATE, and

b. YR-TO-DATE-COUNT

Example 2 (cont.)

Re-Critique:

- This rewrite has design implications SDATE record and YR-TO-DATE-COUNT.
- From a requirements viewpoint it should specify what the data in the records are, not the name of the record as it exists in the design and implementation.

Re-Re-Specification*:*

3.2.5.7 The system shall provide the following data items (which provides production count balancing information to states) at the end-of-state record:

a. submission date and time, and

b. year-to-date totals.

Independent Verification of Specified Requirements
Example 3

Initial specification:

3.2.5.9 All computer-resident information that is sensitive shall have system access controls. Access controls shall be consistent with the information being protected and the computer system hosting the data.

Critique:

- Two “shalls” under one identifier.
- The requirement is vague and incomplete. Need to identify the sensitive information.
- What does “consistent” mean?
- As specified it cannot be implemented or tested

Re-specification:

3.2.5.9 All sensitive computer-resident information shall have system access controls, consistent with the level of protection. (Reference *Sensitive Information,* Table 5.4.1 and *Level of Protection for Sensitive Information*, Table 5.4.2)

Example 4

Initial specification:

3.3.2.1 The system shall have no single point failures.

Critique:

- This is an ambiguous requirement. Needs identification of what components and/or functions the “no single point failures” applies to.
- As specified it cannot be implemented or tested.

Independent Verification of Specified Requirements
Example 4 (cont.)

Re-specification:

3.3.2.1 The following system components shall have no single point failures:

a. host servers
b. networks
c. network routers
d. access servers
e. hubs
f. switches
g. firewalls
h. storage devices

Example 5

Initial specification:

3.2.7.1 The system shall purge state control records and files that are older than the operator or technical user-specified retention period.

Critique:

- Requirement is incomplete and vague without specifying the retention period or providing a reference as to where the information can be obtained.
- Requirement cannot be implemented or tested as stated

Re-specification:

3.2.7.1 The system shall purge state control records and files that are older than the retention period input into the system by either the:

a. operator, or

b. technical user.

Example 6

Initial specification:

3.2.6.3 The system shall receive and process state return data from the State Processing Subsystem. The system shall provide maintenance of the state data files and generate various reports.

Critique:

- Two “shalls” under one requirement number and multiple requirements in the specification.
- The word “process” in the first shall is vague. Need to define the processing required.
- The second “shall” does not provide for valid requirements; they cannot be implemented or tested as stated.
  - » Needs identification of type/amount of maintenance required.
  - » “various reports” is ambiguous.

Independent Verification of Specified Requirements
Example 6 (cont.)

Re-specification:

3.2.6.3 The system shall receive:

a. production data that contains data from multiple states, and

b. state total amount for one or more states,
extracted by the Returns Processing Subsystem.

3.2.6.4 The system shall parse multi-state data to respective state files.

3.2.6.5 The system shall display a summary screen reporting the results of processing for each state containing:

a. state totals,

b. state generic totals, and

c. state unformatted totals.

Independent Verification of Specified Requirements
Example 7

Initial specification:

3.2.7.1 The system shall not prevent the individuals from entering the year for which they intend the payment, but shall provide a check-point for them to ensure that they are not making a mistake in entering the correct year.

Critique:

- This is a negative requirement, negative requirements should not be specified. They cannot be implemented.
- A requirement should have all conditions that are required. If conditions are not required they will not be implemented
- Two “shalls” under one requirement number.
- Suggest that this requirement be structured in a positive fashion.

Example 7 (cont.)

Re-specification:

3.2.7.1 The system shall:

a. allow individuals to enter the payment year, and

b. provide a check-point to ensure that individuals enter the correct payment year.

Independent Verification of Specified Requirements
Example 8

Initial specification:

After the system receives the Validation file, the system shall:

- notify the individual about acceptance or rejection.
- the acceptance file must contain the name and ZIP code of the individual.
- rejected validation request must include the Reason Code.

Critique:

- The second and third bullets don't make sense, try to read them as such:
  - the system shall the acceptance file must...
  - the system shall rejected Validation…
- Use of both "shall" and "must".
- No unique identifier, use of bullets. Bullets cannot be traced
- This requirement is ambiguous and cannot be implemented or tested.

Independent Verification of Specified Requirements
Example 8 (cont.)

Re-specification:

3.2.7.3 When the system receives a validation file, the system shall:

a. reject the file if it does not contain the individuals:

1. name;
2. ZIP code, and

b. notify the individual about acceptance or rejection with a reason code. (Reference *Reason Code*, Table 5.4.8)

Independent Verification of Specified Requirements
Example 9

Initial specification:

3.2.8.2 The enrollment process shall take from one to ten calendar days to complete for all payment types.

3.2.8.3 The enrollment process shall take no more than three days to complete for:

a. credit payment, and/or

b. note payment.

Critique:

These requirements are inconsistent and in conflict with each other.

Example 9 (cont.)

Re-specification:

3.2.8.2 The enrollment process shall take:

a. one to three calendar days to complete for:
   1. credit payment, and
   2. note payment, and

b. one to ten calendar days to complete for all other payment types.

Independent Verification of Specified Requirements
Example 10

Initial specification:

3.2.9.1 When doing calculations the software shall produce correct results.

Critique:

- Really? This is not a requirement
- This type of requirements should not be specified!
- It should be deleted.

Re-specification:

Requirement deleted

Attendee Participation

What is wrong with these requirements?

3.2.1 The Distinguished Name for an Entry can be the value in an attribute for another System Entry or multiple Entries.

- The interval for propagating changes to Suppliers shall be configurable.

3.7.2.1 The System shall create audit log entries for all update transactions. These log entries shall contain at least the following information: transaction type, transaction initiator identifier, data before transaction, data after transaction, affected entry, and time of transaction.

Independent Verification of Specified Requirements
Summary

- The teams identified over 1000 requirements.
- The issues with their initial specification represented the entire spectrum of the following critical attributes:

  - » Unambiguity
  - » completeness
  - » traceability
  - » testability
  - » consistency
  - » feasibility
  - » unique identification
  - » design free
  - » use of shalls
- The teams were receptive to the critique, resolved issues and implemented the recommendations willingly.
- The requirements resulting from this effort were:
  - » reviewed with senior management and stakeholders
  - » accepted as specified
  - » baselined
  - » allocated to development teams for implementation.

Conclusion

- If sufficient time and proper effort is taken to verify requirements against critical attributes during their definition and specification, projects will improve their probability of success considerably.
- If this is not done, projects pay the consequences during implementation, integration and test – not to mention during operation.

But you *knew* that, didn't you?
(I hope!)

IV&V Examples

- Extreme Example of IV&V
- Independent Verification of the Requirements Specification Process
- *Independent Validation of Requirements*
- Evaluation of an IV&V Process
  - » Recommendations to the Program Office
- Application of IV&V with Limited Resources

Independent Validation of Requirements

- Validation *(Are we building the right product?)*
  - » The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.
- These examples show a method of constructing validation scenarios and procedures to support the evaluation of a software product to ensure that the it satisfied its requirements as specified.

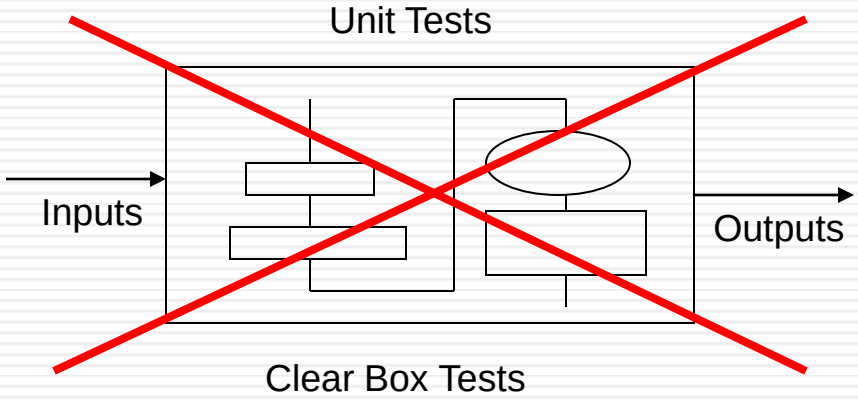
Independent Validation of Requirements

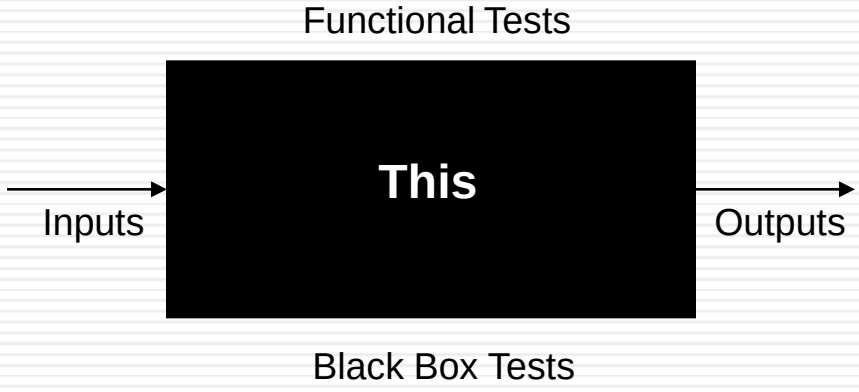
- A Civil agency's system
  - had been in operation for 5 years
  - had never been through either functional or acceptance tests
  - had almost no documented requirements
- MITRE was asked to support an IV&V effort on the system.
  - reverse-engineered requirements from:
    - whatever legacy documentation existed
    - interviews with domain experts and system users
  - developed:
    - Validation Plans
    - Validation Schedules
    - Validation Scenarios
  - supported the development of validation procedures
- The Government agency conducted the validation testing

Independent Validation of Requirements

- The following examples Illustrate:
  - » High-level validation scenarios
  - » Detailed validation procedures

Test execution examples are not provided since test activities are beyond the scope of this presentation.





Independent Validation of Requirements
Validation Scenario

Validation scenarios describe at a high-level what needs to be accomplished during testing to ensure that the implemented system satisfies its requirements as specified. Validation scenarios describe the functionality that is to be tested.

- Validation Scenarios include:
  - A high-level statement of the purpose of the scenario (requirement/functionality tested)
  - Description of scenario
    - test conditions
    - test conduct
    - test validation
  - Description of what/where to validate
  - Identifies the validation method : test, demonstration, inspection, analysis

Independent Validation of Requirements
Validation Procedure

- Detailed validation procedures implement the validation scenarios and describe how the testing is accomplished to validate that the system, as developed, satisfies its requirements as specified.
- The validation procedures establish the specific data and steps needed to be performed in order to validate the system/software against its requirements.

Independent Validation of Requirements
Validation Procedure (cont.)

- The following provides a sample of the information contained in each validation procedure:
  - Identification of requirements tested by the procedure.
  - Identification of test data or other information required to determine test results.
  - Test operators' actions for each step, as required:
    - Initiate the test case and apply test inputs
    - Perform interim evaluations of test results
    - Request data dumps
    - Record data and test results
    - Modify data, if needed
    - Repeat the test case, if needed
    - Use evaluation criteria to validate that requirements are satisfied
    - Determine Pass/fail
    - Provide test comments

Independent Validation of Requirements
Scenarios/Procedures

- Scenarios and procedures can be developed at different levels for either:
  - » an individual unique single requirement
    - > One scenario and one procedure may be necessary for a requirement
  - » a logically related set of requirements that provides a functional capability
    - > In this case, the set of requirements may be grouped and addressed by one or a few scenarios and procedures

Independent Validation of Requirements
Example 1

Requirement

3.3.1.3 The Financial Agent (FA) shall send the Government the following critical data collected from the enrolled individuals by 6:00 PM ET on the same day as receipt or the next day if received after 5:30 PM:

a. Name,

b. Address,

c. Zip code,

d. Social security number.

Independent Validation of Requirements
Example 1 (cont.)

Validation Scenario

S0032 for 3.3.1.3 - Validate that the FA sends the Government critical data collected from the enrolled individuals by 6:00 PM ET on the same day as receipt or the next day if received after 5:30 PM ET.

1. Construct a file with the required critical data for an individual.
2. Initiate input to the system of the constructed file.
3. Validate that the requirement was met.

Validation Method - Demonstration

To validate that the requirement was met, check to see if the Government received the critical data by 6:00 PM ET on the same day as receipt by viewing the appropriate file in the Government's system.

Independent Validation of Requirements
Example 1 (cont.)

Validation Procedure

Date	Req#	P/F	Scen#	Procedure P0024	QA
	3.3.1.3		S0032	1. Data: Al Florence, 26 Dutch Creek Drive, Columbine, Colorado, 80123-1623, 374-XX-4237 2. Input data into Enrolled Individual Critical Data file on FA System. 3. Initiate the execution of the Enrollment Function on the FA System. 4. Validate that the Government received the data in (1) by a) 6 PM ET on the same day as receipt; or b) the next day if received after 5:30 PM ET by checking the Enrolled Individual Critical Data file in the Government System	
Comments					

Date - Date test conducted
P/F - Pass/Fail indication
Procedure - Procedure's number and text
Req # - Requirement(s) being verified
Scen # - Scenario being implemented
QA - Quality Assurance witness' initials
Comments - Comments on test results

Al Florence

Independent Validation of Requirements
Example 2

- Requirement 1

  3.3.2.1 Prior to noon each day, the FA shall accept a payment file from the enrolled individual.

- Requirement 2

  3.3.2.2 Within one hour after receipt of the payment file from the individual submitting the payment file, the FA shall provide the individual an acknowledgement of its receipt.

- Requirement 3

  3.3.2.3 Upon receipt of the payment file, the FA shall:

  a. Reject the payment file if the individual is not enrolled.
  b. Reject the payment file if the payment type is invalid.
  c. Send the payment file to the Government if the payment file is not rejected.

Independent Validation of Requirements
Example 2 (cont.)

Validation Scenario

S0033 for 3.3.2.1, 3.3.2.2, 3.3.2.3. Validate that the FA receives payment file, sends acknowledgement, correctly processes, and sends the payment file received from the individual submitting the payment to the Government.

1. Construct:
   a. enrollment records,
   b. payment files - multiple sets representing enrolled and non-enrolled individuals, and valid and invalid payment type,
2. Initiate input to the FA of the constructed files,
3. Validate that the requirements were met. Validation Method - Demonstration

Validation Method - Demonstration

- To validate that the Financial Agent received and accepted the payment files from the individuals submitting these files, check that the FA sends the acknowledgement.
- To validate that the FA correctly processed the payment file and sent the payment file to the Government, check the appropriate Government files.

Al Florence

Independent Validation of Requirements

Example 2 (cont.)

Validation Procedure

Date	Req#	P/F	Scen#	Procedure P0025	QA
	3.3.2.1		S0033	1. Enrollment Data: a. Steve Jenkins, 244 Maple St, Fairfax, VA 20171, 334-XX-4445; b. Jeff Hunt, 517 Main Ave, Fairfax, VA 20171, 422-XX-5555; 2. Payment Data: a. Steve Jenkins, 334-XX-4445, Valid Payment Type; b. Jeff Hunt, 422-XX-5555, Invalid Payment Type; c. Barbara Jones, 335-XX-1234, Valid Payment Type; d. Fred Smith, 275-XX-4321, Invalid Payment Type; 3. Initiate input of enrollment data to the FA System. 4. Check for enrollment file acknowledgements. 5. Initiate input of payment file to the FA System. 6. Check for payment file acknowledgements. 7. Analyze Government files for receipt and correct processing by the FA. Only the payment file for Steve Jenkins should be in the Government files since only Steve Jenkins was enrolled and presented a valid payment type.	
	3.3.2.2				
	3.3.2.3				
Comments					

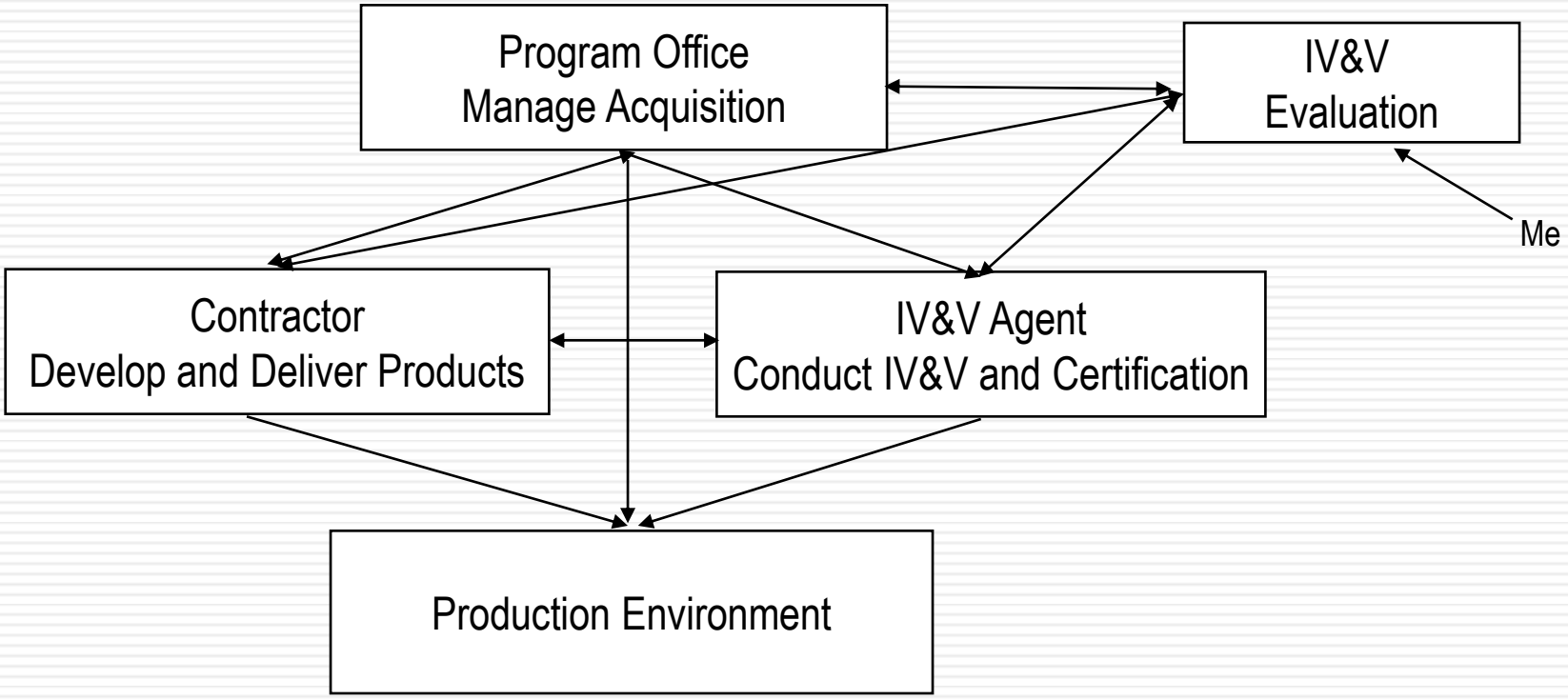
IV&V Examples

- Extreme Example of IV&V
- Independent Verification of the Requirements Specification Process
- Independent Validation of Requirements
- *Evaluation of an IV&V Process*
  - » *Recommendations to the Program Office*
- Application of IV&V with Limited Resources

Evaluation of an IV&V Process

- A Program Office was confused as to what IV&V entailed
  - » Mentioned earlier in this presentation (IV&V White Paper)
- Could not understand
  - » Why the IV&V cost so much
  - » Why IV&V took so long
  - » Why Certification took so long
- Asked this author to
  - » Define IV&V (IV&V White Paper)
  - » Investigate current IV&V and Certification Process and make improvement recommendations

Organizational Relationships



White Paper

- Produced IV&V White Paper for Program Office (presented earlier)
  - » Defined a generic IV&V process
    - > Tailored It for current acquisition
  - » Copy sent to IV&V Agent and Contractor

Activities

- Reviewed
  - Contractor test and evaluation process and artifacts
  - Contractor Quality Assurance process and artifacts
  - IV&V Agent IV&V and Certification processes and artifacts
- Interviews
  - Program Office personnel
  - Development contractor test and evaluation engineers
  - IV&V Agent manager and engineers
- Attended
  - Contractor Technical Interchange Meetings (TIMs) and formal design reviews
  - IV&V Agent's Certification meetings
- Surfaced issues/Provided Recommendations
  - IV&V
  - Certification

Evaluation of an IV&V Process

Observations/Recommendations

- IV&V agent only conducting validation
  - » Although they called it IV&V
- IV&V agent should define V&V as per IEEE
  - » Verification –The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.
  - » Validation – The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.
- Recommended
  - » IV&V agent should include verification in their IV&V plans and activities
  - » This would not add effort since life cycle products are independently reviewed at the various major milestone and TIMs in support of IV&V and the project office.

Observations/Recommendations

- Parts of V&V as defined in IEEE are accomplished in other parts of the organization – Systems Engineering, Software Engineering, Test Engineering, Quality Assurance, and Configuration Management.
- Recommended
  - » IV&V Agent coordinate with and leverage Contractor Test and Evaluation results (synergism)
    - > Leverage Contractor's Quality Assurance results
    - > Independent review and witness Contractor's test program and test execution
    - > Use Contractor's test results and reports as appropriate
  - » This approach should be described in the IV&V process descriptions and in the IV&V plans to provide full credit in support of the IV&V efforts

This could be accomplished if the IV&V organization acted in an oversight role in an independent and objective fashion.

Observations/Recommendations

- The IV&V agent tests system level (A Specification) requirements and safety related software requirements
- The contractor was only testing functionality for the tactical system
  - » No one was testing the entire set of software requirements for the system
  - » The contractor did not trace requirements the design or test cases
- Recommended
  - » If the contractor tested all system level and software requirements
    - > The IV&V agent could then leverage these test and test results and reduce their IV&V efforts and costs (synergism)
    - > Then they only need to validate those requirements that need additional validation such as critical safety requirements or requirements that were not adequately tested by the contractor

Observations/Recommendations

- Some requirements in the System Requirements Specifications were obsolete
  - » Some requirements were no longer valid
  - » Some requirements were never implemented
  - » Some requirements were badly specified
- Recommended
  - » If the system requirements were rewritten the implementation, testing and IV&V efforts would be greatly enhanced

Observations/Recommendations

- Like many projects the specification of requirements needs attention
  - Some requirements are specified at too high a level which makes them meaningless
  - Some at too low a level which constraints the design
  - Many were ambiguous and confusing
- Recommended
  - Requirements need to be specified at the proper requirements level in an unambiguous fashion
    - It becomes very difficult and costly to test requirements that are not specified at the proper requirements level
      - If at too high a level then the real intent of the requirements is untested
      - If at too low a level then design is tested which adds effort the to the test activity
  - Also, when formally testing to design, the Configuration Control Board (CCB) needs to get involved with unnecessary test issues
    - CCBs are expensive and time consuming and should only be concerned with “real” requirements issues

Observations/Recommendations

- Certification meetings were taking days if not weeks to complete
  - » Meeting scheduled when certification data were incomplete
  - » Certification Panel members were unprepared at meetings
  - » Certification meetings were often extended or canceled
- Recommended
  - » Certification meeting be held only when all certification data are complete and available
  - » Certification Panel members prepare for meeting else meeting canceled
- Reduced certification meetings to a day or less

Evaluation of an IV&V Process

Final Results

- Program Office had a clear understanding of:
  - The IV&V Process
  - The Certification Process
- IV&V effort and cost reduced
- Certification process streamlined

IV&V Examples

- Extreme Example of IV&V
- Independent Verification of the Requirements Specification Process
- Independent Validation of Requirements
- Evaluation of an IV&V Process
  - » Recommendations to the Program Office
- *Application of IV&V with Limited Resources*

Application of IV&V with Limited Resources

- Agency had prior failures
  - » Projects late and over cost
  - » Projects did not meet customer needs
  - » Projects cancelled
- In order to avoid this on current project
  - » Program office requested IV&V support
  - » This author was asked to provide this IV&V
  - » Program office had limited resources
  - » IV&V effort was one day per week
  - » Author was asked to recommend planed IV&V activities

Planned IV&V Activities

- The amount of IV&V is limited by time and resource constraints.
- As much as can be done will be done based on constraints.
- IV&V activities will be selected based on:
  - Critical items
  - Management direction
  - Random selection
- The approach will be to attempt 100% coverage on activities selected knowing that this may not be feasible.
- IV&V of any one activity will continue until time expires and the next items needs to be addressed.

Planned IV&V Activities

- IV&V areas of concern are:
  - Review System Requirements Specification (SRS) and other related requirements
  - Review specified requirements against the following critical attributes:
    - Ambiguity
    - Complete
    - Traceable
    - Testable
    - Consistent
    - Feasible
    - Uniquely Identified
    - Design Free
    - Using “Shall” and Related Words

Planned IV&V Activities

- Review bidirectional traceability of requirements
  - To/from system level to/from subsystem/segment level
  - To/from subsystem/segment level to/from allocated level (software, hardware, facilities, manual procedures, builds)
  - To/from system/subsystem/segment level to/from system test procedures
  - To/from allocated level to/from design level
  - To/from allocated level to/from functional test procedures
- Review Configuration Management of Requirements
- Review that proper baselines have been achieved:
  - Functional Baseline (FBL)
  - Allocated Baseline (ABL)
  - Product Baseline (PBL)

Planned IV&V Activities

- Review change management of requirements
  - CCB activities related to requirements
  - Change requests
  - Impact assessments
  - Change implementation
- Formal Test Planning
  - Review test plans
  - Review test procedures
  - All requirements accounted for

Planned IV&V Activities

- Test conduct
  - Observe selected tests
- Review defect reports
  - For identification of design issues
  - For identification of requirements issues
  - For implementation of corrective actions
- Review test reports
- Other areas as directed
  - Risk Management
  - Quality Assurance
  - Design Review(s) products

IV&V Activities Conducted

- IV&V executed for 10 months
  - » Reviewed contractor's V&V plan
    - > No issues
  - » Reviewed the System Requirements Specification (SRS) against the critical attributes
    - > Surfaced many issues and violations against the critical attributes
    - > Program Office and contractor accepted comments and made appropriate corrections

IV&V Activities Conducted

- Reviewed Risk Management process
  - The program office mandated that at least 200 risks should be identified
  - This was a fairly small program although very critical and somewhat complex
  - Contractor was identifying risks that were not real risks
    - In order to meet the 200 risks
    - 77 risks had been identified
    - Program Office stated that that was not enough
  - Convinced Program Office that only real risks need to be managed
    - To create a risk watch list for problems that are not yet risks and managed at a lower program level
    - This recommendation reduced the 77 risks to 3 real risks managed at the appropriate management level

IV&V Activities Conducted

- Reviewed the System Design Description (SDD) and the Interface Control Document (ICD)
  - The main issues with documents was that they were using an SDD and ICD format of a development effort
  - This effort was a COTS solution
  - The SDD and ICD were written as if the solution was a development effort
  - This was reported as a finding

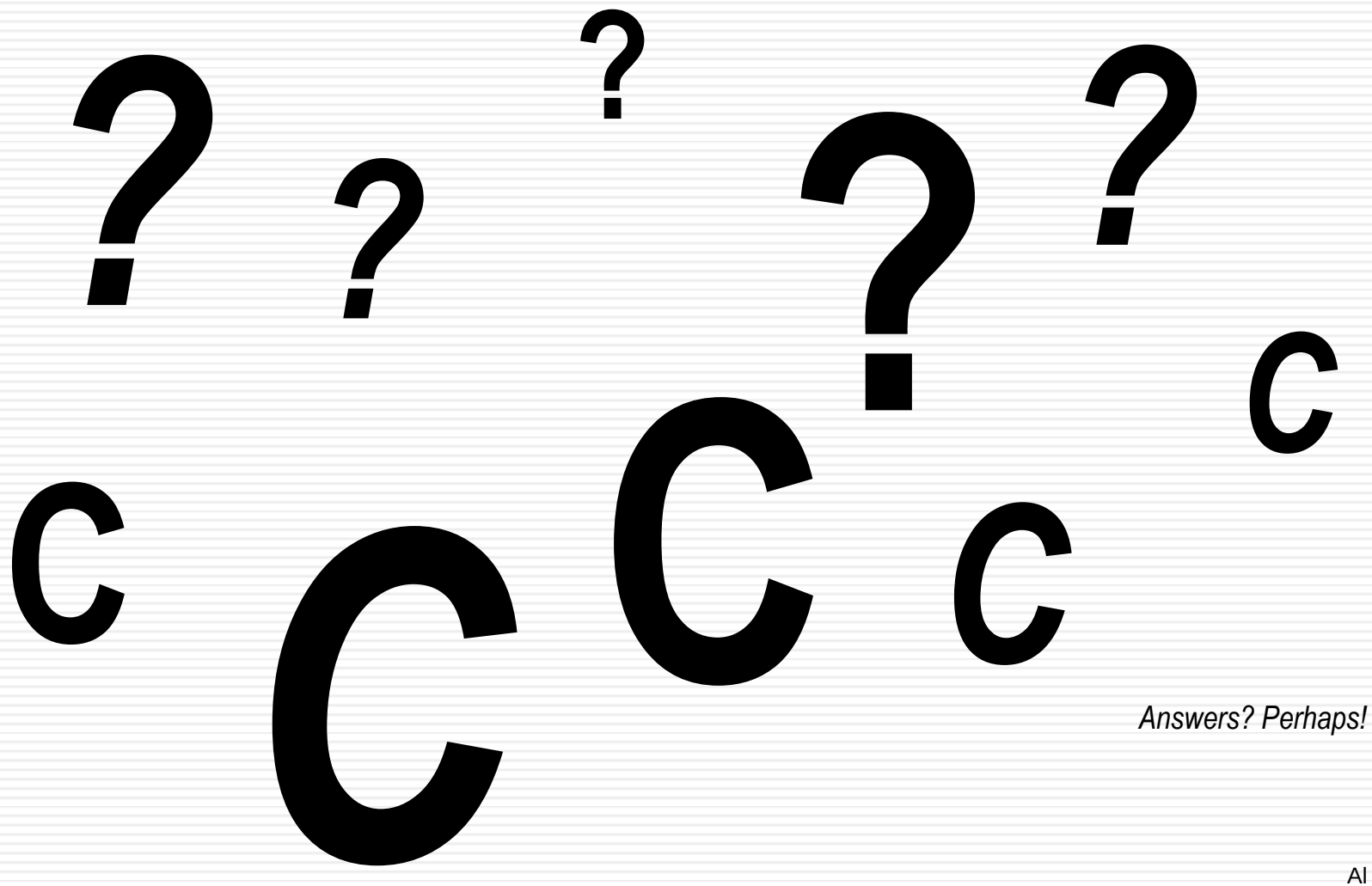
IV&V Activities Conducted

- Reviewed the formal test program
  - Reviewed test plan
  - Reviewed test cases
  - Reviewed test procedures
  - Reviewed requirements traceability to test case
    - Traceability initially was deficient
      - ✓ Reported this as a major problem
    - In later stages of traceability problems were corrected
  - Test engineer agreed that if the requirements' issues surfaced in the SRS review were corrected
    - That her job would be much easier not having to get with the implementers and requirements engineers for clarification of ambiguous requirements

Agenda

- Attendee Participation
- Challenges
- Solutions
- IV&V Defined
  - » Verification
  - » Validation
  - » Independence
  - » IEEE IV&V - CMMI® V&V
- IV&V Relationships to other Disciplines (Test, Quality Assurance)
- IV&V; This Presentation
- Overview of Independent Verification & Validation (IV&V)
- Tailoring IV&V
- Examples of IV&V
- *Questions/Comments*
- *References*

Questions/Comments



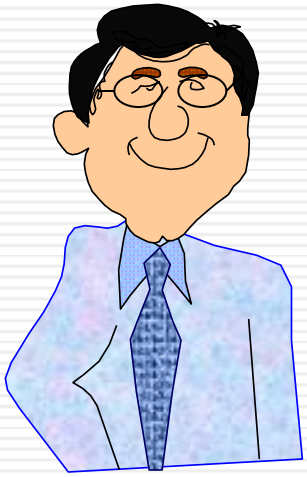
Answers? Perhaps!

Al Florence

References / Suggested Reading

- *IEEE Standard for Software Verification and Validation*; IEEE Standard 1012 2004; IEEE Computer Society
- *Capability Maturity Model Integration (CMMI®) for Development v1.3 2010*; Software Engineering Institute (SEI)
- *Reducing Risks with the Proper Specification of Software Requirements*; Al Florence; CrossTalk; April 2002
- *Requirements Validation*; Al Florence; Quality Assurance Institute Journal; October 2003

Contact Information



Me

Al Florence
florence@mitre.org
703-395-6700 CEL
303-955-2286 Office