Developing Verification Requirements to Assure Project Success
Mark A. Powell

Abstract
A well thought out and planned set of verification requirements can make all the difference between a happy customer and a dissatisfied customer. If you cannot provide confidence to the customer that you met their requirements, you won’t have a happy customer. Yet few projects these days invest much effort in developing good verification requirements. Quite often, verification requirements are treated as an afterthought to the performance requirements. But, if we produce the verification requirements concurrently with the performance requirements, the verification requirements can expose problems in the performance requirements they are intended to verify. Of course, this not only leads to better verification requirements, but to better performance requirements, which leads to a happy customer.

Why is it then that most requirements training courses just gloss over the development of verification requirements? Verification requirements are actually very different from the requirements they support. While they share the same syntax and basic requirements writing rules as performance requirements, you usually need many verification requirements to properly verify that a single performance requirement is satisfied.

This tutorial will enlighten both the new SE as well as the grizzled veteran who needs a refresher about how to properly develop good verification requirements. We will go though a number of real world examples of good and bad verification requirements, and how they contributed to success and failure. We will establish the philosophical foundation for writing verification requirements, and present a simple process for developing verification requirements for broad classes of performance and functional requirements.

Biography
Mark A. POWELL, USA, has practiced Systems Engineering for over 35 years in a wide variety of technical environments including DoD, NASA, DOE, and commercial. His roles in these environments have included project manager, engineering manager, chief systems engineer, and research scientist. Mr. Powell began teaching in academia in 1980, and has been teaching Systems Engineering at the graduate level since 1989. He is currently an adjunct member of the Stevens Institute of Technology Systems Engineering Faculty, and of the University of Houston, Clear Lake Systems Engineering Faculty. Mr. Powell maintains an active engineering and management consulting practice (currently in affiliation with SAIC), providing consultation to the Army’s Future Combat Systems program during 2003-2005, and now for the NASA’s Constellation Program. He is an active member of Sigma Xi, AIAA, ISBA, and helped to found the Texas Gulf Coast Chapter when he joined INCOSE in 1991.
Developing Verification Requirements to Assure Project Success

Mark Powell
Attwater Consulting
11 July 2010
Chicago, IS 2010
Tutorial Schedule

13:30 – Preliminaries, Introductions
13:45 – Some Basics
14:55 – Break
15:10 – Verification Requirements and How to Write Them
16:50 – Summary and Evaluation
17:00 – You are free to go
Attendee Introductions
(Time Allowing)

• Your Name, Employer, and Type of Business
• Your Job and Type of Work You Do
• Do You Work with Verification?
• What you are hoping to gain from today’s tutorial
Your Tutorial Instructor: Mark A. Powell

- Professor, Systems Engineering
  - Stevens Institute of Technology
  - University of Houston Clear Lake
  - University of Idaho
- Over 35 years Experience in Systems Engineering and Project Management
- Former Chair, INCOSE Risk Management Working Group
- INCOSE Technical Leadership Team, Former Assistant Director for Systems Processes
- Contact Information at the End of Tutorial Presentation, Contact Welcomed

Developing Verification Requirements to Assure Project Success, INCOSE IS2010 © Mark Powell, Attwater Consulting 2000-2010; attwater@aol.com, 208-521-2941 Slide # 4
First a Word about Terminology

• I Hope to Establish a Terminology for Today’s Tutorial
• Many Uses Abound for Many Terms you will see Today
  • Not all Uses in Agreement or Consistent
  • Even if Hard Definitions Exist, Terms in Most Projects are *Used Freely*, with Good and Effective Understanding
• I will be Using Many Terms, and If you are not Sure about My Meaning in the Particular ad hoc Context
  PLEASE ASK!
• *Substance, Concepts, and Content* in Verification and Verification Requirements are Much *More Important Than Definitions*
Before We Get to Verification Requirements

... Some Basics
I V&V

• For this Tutorial:
  • Integration, Validation, and Verification
  • NOT Independent Verification & Validation

• I V&V are All Intermingled
  • Usually Managed Together
  • Mostly Contemporaneously Performed
  • Sometimes Confounded
Integration

• The Word *Integration* has Many Meanings and Uses

• For Systems Engineering Practice, Comprises Two Very Important Activities
  • Interface Definition, Management, and Control
  • System Builds
Interface Management and Control

• In Development of a System of Any Appreciable Size, Potential for Exponential Growth in Numbers of Interfaces

• The Process to Control and Manage Interfaces should be Documented in the SEMP
  • Interfaces Identified in Development Phases
  • Definition and Control Usually Performed Through Interface Control Working Groups (or Integration IPT’s)
  • Interface Control Documents become Part of the Formal Baseline and Provide Means to Manage Interfaces
A Relevant, but Complex Process
System Builds

- Actually, just a Build from a Lower Hierarchy Level to the Next Higher – e.g., Components built into Subassemblies, Subassemblies into Subsystems, etc.
- Consider the Requirements Allocation Process in Functional Analysis and Decomposition
- Component Performance Requirements Must Satisfy Subassembly Performance Requirements after the Build
Integration Summary

- Interface Definition, Management, and Control
  - Definition: Mostly Left Side of “Vee”
  - Management and Control Everywhere
- System Builds
  - Identified and Planned on Left Side
  - Executed on Right Side
Validation

• The Root: VALID
  • 1. sound; just; well-founded*
  • 2. producing the desired result; effective
  • 3. having force, weight, or cogency; authoritative
  • 4. legally sound, effective, or binding; having legal force: a valid contract.
  • 5. Logic. (of an argument) so constructed that if the premises are jointly asserted, the conclusion cannot be denied without contradiction.

• Often Confused with Verification
  • Sometimes Erroneously
  • Sometimes NOT!

*Definitions from Dictionary.com
Validation

• Answers Two Fundamental Questions (with a YES of Course)
  • Are we Addressing the Right Problem?
  • Is our Solution Solving that Problem?
• When Validation Activities Occur
  • Continuously, by Every SE on the Project
  • The Primary Purpose of Every Control Gate (Review)

Verification ensures you built it right

Validation ensures you built the right thing
Practical Validation

• Many Use the Terms Verification and Validation Interchangeably
  • May not Understand the Subtleties
  • May just be Using Sloppy Diction
  • May be Focused on a Point Solution

  Be Careful!

• The Problem Always Evolves with Time
  • By Continuously Validating, You catch Problem Evolution Earlier – Easier to Fix Solution and Cheaper – Always!
  • By Continuously Validating, You Catch when the Solution Goes Astray Earlier – Easier and Cheaper to Fix, Always!
  • All Reviews Always Address Problem Space as well as Solution Space
  • All Program Documentation Must Evolve with the Evolution of Problem Space to Maintain a Realistic and Contemporaneous Baseline
Now Something to Add to the Confusion

- You have to Verify that the Problem Space is Valid
- You have to Verify that the Solution Space is Valid
- The Term Verify in this Case is Not Always a Formal Verification Process
  - Typically, a Lot of SE Audits are Used to Verify Validity
  - Reviews also Use a Lot of SE Audits
  - However, If Uncertainty Exists about the Validity of a Solution – Formal Verification Process may be Needed
- Be Careful in your Communications!
  - Validation Always Answers the Two Questions
  - Verification of the Correct Problem and that the Solution Solves the Problem Provides the Validation
- Most Importantly, Be SURE it Gets Done!

Developing Verification Requirements to Assure Project Success, INCOSE IS2010 ©
Mark Powell, Attwater Consulting 2000-2010; attwater@aol.com, 208-521-2941
Verification
The Fundamental Concept of Verification

• Verification is a Risk Mitigation Process
  • If we could be Absolutely Certain that the Contractor would Build Exactly What we Intend in our Specifications, there would be No Risk
    • BUT, Not Always Sure Our Requirements will Be Understood and Interpreted Properly
    • Not Always Sure Our Requirements are Complete and Consistent
  • Verification Reduces this Risk, but Cannot Completely Eliminate it
• How Well Verification Requirements are Written Determines How Well Execution of Verification Mitigates the Risk
Short Break!

Please Be Back by 15:15
Now to Verification Requirements

- Background on Verification and Verification Requirements – *To Answer the Question of Why?*
- A Process for Writing Verification Requirements – *To Answer the Question of How?*
  - Verification Methods, Some Clues How to Select One
  - Structure of Verification Requirements
  - How to Write Verification Requirements
What Can Happen without Verification Requirements

- Space Station PRD Requirement
  - *The Space Station orbit shall be nominally circular.*
  - No Corresponding Verification Requirement in PRD
- Author had Air Force background: “nominally circular” means Orbit Eccentricity $< 10^{-2}$ for the Air Force
  - For Space Station, Eccentricity of $10^{-2}$ is *physically impossible*
  - Max Possible Eccentricity for LEO: $\sim 5e-4$
- *Could* have Written a Simple Verification Requirement
  
  Analysis shall show that the Space Station GN&C Systems will maintain Space Station orbits with eccentricities less than $10^{-2}$.
What Happened without a Verification Requirement

- Work Package 2 Contractor *interpreted “nominally circular” a little Differently*
  - Interpreted it to mean orbit eccentricity of *identically zero* (perfectly circular orbits) during *“nominal operating conditions”* (eventually at end of reboost)
  - Required Very Expensive GN&C System On-board to Plan and Control Trajectory to Achieve Zero Eccentricity
  - Required $25M GPS Development to Measure or Verify Zero Eccentricity (Huge Cost Growth!)
- Major Contributor to *Cancellation of Work Package 2, after Five Years of Unsuccessful Customer Change Requests*
- *A Simple Verification Requirement Could have Changed Everything*
What Can Happen with Poor Verification Requirements

• Contractors *Always* Design and Build to the Verification Requirements, *Not Section 3!*

  *They want to get Paid!*

• Example: Joint STARS
  • Section 3.2.x: *System Shall Direct and Deliver Ballistic Bombs with 100m CEP Accuracy.* (100m Value Made Up)
    • CEP is circle about Target containing 50% probability
    • Aside for Section 3: CEP’s Great Way to Specify 2D Accuracy Requirements (SEP is 3D analog)
  • Section 4.3.2.x: *JSTARS bomb delivery accuracy shall be satisfied if 5 or more of 10 bomb drop tests land within 100m of target.*
  • Sounds *Very Reasonable, Doesn’t It?*
What Happens with Poor Verification Requirements

• Suppose the Contractor Met the Requirement: JSTARS System as Built would Deliver Bombs within 100m of Target with 50% Probability

• What was the Probability that the Contractor would Pass this Test and Get Paid?
  • Rather Straightforward Binomial Probability Calculation
  • Answer: Just 62.3%!

• How the Contractor Responded to This Result
  • Over-designed System to Deliver 80% of all Bombs within 100m, Increasing Cost Several Fold
  • Obtained 99.4% Probability of Passing Test and Getting Paid
What Else Can Happen

- Verification Requirements *Establish the Level of Risk* the Customer will Accept for Satisfaction of a Section 3 Requirement, *Whether Stated or Not!*
  - Usually, Acceptable Risk is Not Specified, But Could Be (Test)
  - A Thorough Analysis of Verification Requirements for Acceptable Risk is a useful SE and Management Practice
- What *Acceptable Level of Risk* did this Joint STARS Verification Requirement Establish for the Customer?
  - Can be Calculated
  - The Answer: *The Customer Accepted a 50% Risk*
  - The Probability that the Contractor Really Provided a System with a 100m CEP, *given* that 5 or more of 10 Bombs landed within 100m (*they passed the test*), was only 50%
How to Fix It

• This Verification Requirement for Joint STARS
  • NOT Good for the Contractor
  • NOT Good for the Customer

• Write a Better Verification Requirement:
  JSTARS bomb delivery accuracy shall be satisfied if bomb drop tests show a 90% probability that 50% of bombs dropped will land within 100m of target, with at least 3 and no more than 10 bomb drop tests.
  • The 90% Probability (for example) indicates the government will accept a 10% risk that the completed system will satisfy this accuracy requirement – Very Clear to Everybody
  • Note the opportunity for Cost Savings – Does not require all 10 bomb drop tests
  • Note that the Probability for the Contractor passing the Test is Purely Dependent on whether or not they Satisfied the Section 3 Requirement
Advantages of Writing Good Verification Requirements

- Improves the Section 3 Requirements
  - Measure of Verifiability of Section 3 is the Ease of Writing Section 4 Requirements
  - Measure of Clarity of Section 3 is Number of Iterations on Writing Section 3’s and Section 4’s for Consistency
- Quantifies, or enables Quantification, of Acceptable Risk to the Project for Section 3’s
- Establishes a priori Agreement on Evidence Customer will Accept that Section 3’s are Satisfied
- Enables Good Life Cycle Cost Estimates and Early Planning

Very Important Concept
The Morals of these Examples

- You **Want** to Write Verification Requirements to be Sure your Section 3 Requirements are *Understood Properly, the way you mean them!*
  - Verification Requirements Provide Further Clarification and Communication – *Writing the Verification for a Section 3 concurrently Really Helps!*
  - Good Verification Requirements Can Go a Long Ways to Compensate for Poorly Written Section 3’s
- You **Want** to Write **Good** Verification Requirements!
  - Good Verification Requirements Establish *the Acceptable Risk* for the Customer
  - Good Verification Requirements Establish the Risk to the Contractor
  - Good Verification Requirements *Eliminate* or Reduce the Potential for *Wasteful* Contractor Costs
Who Should Write the Verification Requirements?

• **Who Knows Best** what the Section 3 Requirement to be Verified *Really Means*?
• Practice of Throwing Section 3’s *over-the-fence* for Someone Else to Write Section 4’s is *Very Expensive*, *if Ever Effective*

**Obvious Answer:**
**The Section 3 Requirement Author**
Cost Considerations

• Planning for Verification Can Identify Needs for *New Major Projects or Facilities*
  • E.g., Projects to Build Test Articles and Test Facilities
  • Early Identification Enables Most *Cost Effective* New Projects
• *How you Verify* Can Pose Very Different Costs
  • *Repeated Experiments* on Actual HW/SW to Collect Observations/Data for Statistical Processing (Often Test Method): *Most Expensive*
  • *Actually Operating the System HW/SW to show that it Works* (Often Demonstration Method): *Can be Very Expensive*
  • *High Fidelity Simulations* (Often Analysis Method): Can be Expensive, but *Usually More Cost Effective*
  • *Observing Compliance by Examination* (Often Inspection Method): Almost always *Cheapest*

*Verification Activities Can Consume the Vast Majority of Project Resources!*
Verification in the Project Life Cycle

• Verification Activities *Change as the Project Matures*
  • Different Levels of Design Detail Lead to Different Verifications
  • Verification Method Usage Changes with Maturity Level
• *Top Down* Design Leads to *Bottoms Up* Verification
• *Bottoms Up* Verification Execution Occurs *after* Stages of Integration
  • Integration Occurs *after* Sub-pieces have been Verified Individually and Successfully
  • Verification that the Integrated Piece Meets its Requirements *May Consider* Sub-piece Verification Results
• Verification Requirements written for Highest Levels of Design Detail Cannot Well Anticipate Lower Level Verification Results – so *Use some Thought when Writing*
Verification Categories

- **Development**: Conducted on new items to demonstrate proof of concept. Testing may be done on breadboard, brassboard, engineering prototype, or partial model. Often used to reduce risk, prove feasibility, and provide Validation.

- **Qualification**: On aerospace equipment these ground tests are conducted to prove the design on the first flight article produced, using elevated environmental conditions for hardware. The hardware qualification test items cannot generally be used in an operational test due to overstress.

- **Acceptance**: Conducted to prove workmanship and materials on the second and succeeding articles. Tests conducted are a subset of the qualification tests, performed at lower stress levels.

- **Operational**: Conducted to verify that the item meets its specification requirements when subjected to the actual operational environment. Some requirements, such as radiation hardening, may be fully verified at the parts level by testing. Many requirements at the system level may be verified only by simulation (supported by test data from lower levels).
Verification Methods

• The Method Selected is not as Important as Specifying What the Success Criteria should be and How it is Obtained
• The Methods – Test, Analysis, Demonstration, Inspection
• You Can Use Almost Any Method for Any Section 3, but There are Consequences and Limitations
  • Risks Vary by Requirement and Verification Process
  • Costs Can Vary Widely
• You will Need to Use some Thought and Best Engineering Judgment
Verification Method
Definitions

Inspection: an examination of the item against applicable documentation to confirm compliance with requirements. Inspection is used to verify properties best determined by examination and observation (e.g., - paint color, weight, etc.).

Test: an action by which the operability, supportability, or performance capability of an item is verified when subjected to controlled conditions that are real or simulated. These verifications often use special test equipment or instrumentation to obtain very accurate quantitative data for analysis.

Demonstration: a qualitative exhibition of functional performance, usually accomplished with no or minimal instrumentation. Demonstration (a set of test activities with system stimuli selected by the system developer) may be used to show that system or subsystem response to stimuli is suitable.

Analysis: use of analytical data or simulations under defined conditions to show theoretical compliance. Used where testing to realistic conditions cannot be achieved or is not cost-effective. Analysis (including simulation) may be used when such means establish that the appropriate requirement, specification, or derived requirement is met by the proposed solution.
Some Guidance for Selecting Inspection

- If a Human Being can Observe or Use a Simple Measurement to Determine if the Requirement is Satisfied or Not – then use Inspection
  - Binary Decision, Go/NoGo
  - Simple Judgment Required of Tester using Their Senses
  - May be Simple Measurements (Torque, Length, etc.)
  - Generally, Non-destructive and Executed on Actual System HW/SW
- Risk with this Method is Generally Inherent with the Inspector
- Least Expensive Verification Method
- Used Extensively To Verify CDR Level of Design Detail Requirements
Some Guidance for Selecting Test

- If the Section 3 is an “ilities” Requirement, then You should Use Test or Analysis
- If you are thinking that to Verify a Section 3 that you need to run an Experiment on Actual System HW/SW, Collect Data and Statistically Analyze it, then the Method should be Test
- Specification of Test Does Not Require the Design of the Experiment, but a Thorough Description of the Experiment
- The Test Method Success Criteria is best stated Probabilistically (Recall Joint STARS Example)
- Test - the Best and Most Effective Method to Really Quantify and Reduce Risk for Requirement, but Expensive
- Usually Used Extensively To Verify CDR Level of Design Detail Requirements
Some Guidance for Selecting Analysis

- If the Section 3 can be Verified by *Evaluation of Equations* (say from physics, or from Integration equation from a Lower Level), then the Method is *Analysis*.  
- If *Test* would be *Preferred, but is Impractical*, and You can use a Simulated Experiment, the Method is *Analysis*.  
- *Mathematical Models* are Typically Used to Represent the System, Scenario, and/or Environment.  
- May be *More Difficult to Specify Than Test*, May have to Completely Specify Experiment and All Assumptions.  
- Generally *Much Cheaper than Test*.  
- *Risk* is Inherent in *Assumptions* and *Model Fidelities*.  
- May Require Extensive Model *Validation*.  
- Used to Verify Requirements At *All Levels* of Design Detail.
Some Guidance for Selecting Demonstration

- If you are thinking that to Verify a Section 3 that you need to run an Experiment on Actual System HW/SW, and see only One Datum or Result to be Satisfied (No Statistics, Simple Go/NoGo), then the Method should be Demonstration.
- Specification of Demonstration Does Not Require the Design of the Experiment, but a Good Description of the Experiment.
- Usually Performed at Extremes of Range of Performance (worst case environment, scenarios, etc.).
- When Successful, Generally Non-destructive.
- Risk: Only one Repeat of the Experiment, Only One Datum on which the Go/NoGo Decision is Made – Is one enough?
- Generally used to Verify Highest Level of Design Detail Requirements.
Verification Method
Life Cycle Compositions

T – Test
I – Inspection
A – Analysis
D – Demonstration

Caveat: Ignore All Specific Terminology, For Concepts Only

T & I Usually Negligible at this Maturity Level
How to Select a Verification Method

Ask Yourself Some Questions:

• How Important is this Section 3 Requirement?
  • Properly Written, all Section 3 Requirements are Equally Important, all are Mandatory or Mission Fails
  • System Cannot Perform the Mission if Any Properly Written Section 3 is not Satisfied
  • Back to the Real World!

• How Risky is this Section 3 Requirement?

• How Precise is this Section 3 Requirement?

• How Sensitive is this Section 3 Requirement?

• What are the Consequences of not Satisfying this Section 3 Requirement?
Ask a Few More Questions

• Can Someone with but a High School Education Use Their Senses to Determine if this Section 3 requirement is Satisfied with little Risk? If so, *Inspection* is a Good Candidate

• Would a Simple *One-shot Test*, Perhaps at an Extreme Scenario or Environmental Conditions, Reduce the Risk Enough? If so, then *Demonstration* is a Good Candidate

• Do you Need to Examine *Performance over a Wide Range of Scenarios or Environments*? Or is it an “ilities” Requirement? If so, then you should Use *Test* or *Analysis*
  • Cost is One Discriminator between *Test* and *Analysis*
  • *Feasibility* of the *Test* is Another
  • Is the *Expense* of *Test* Justified to Reduce the Risk?
There is More to Specify Besides the Method

• Each Verification Method has *Different Aspects* that Must be Specified
  • As with Any Section 3, Section 4 Requirements Must not be Subject to Interpretation – *Write Good Requirements*
  • Method Designation is Insufficient
  • Each Method *Needs Subtier Verification Requirements* for Clarity to Achieve Risk Reduction

• One Attribute Common to All Methods
  • Verification Method *Success Criterion*
  • Specified Last in Subtier Verification Requirements

• Other Attributes Relate to How the Verification is to be Performed, e.g., the experiment
Inspection Attributes

- **What is to be Inspected**
  - 3.2.x.y The MPLM RSR Locker door shall be retained and remain closed during shuttle ascent and descent.
  - Design Architecture described in Section 3.1: Screws Used to Retain Door with sufficient hold down torque and keenert (Last Thread Malformation) as secondary locking mechanism.

- **How it is to be Inspected**
  - 4.2.x.y Requirement 3.2.x.y shall be verified by inspection.
    - MPLM RSR Locker Door Retention screws shall be inspected by measurement for minimum hold down torque.
    - Retention screws shall be inspected visually for keenert engagement.
    - Inspection personnel shall be certified for torque measurement systems.
    - Verification of 3.2.x.y shall be considered successful if all screws are observed with hold down torques of at least 15 in-lbs and all screws protrude at least one thread beyond back of screw holes.

- **Who will Inspect it**

- **Inspection Success Criterion**
Test Attributes

• Specify the Measure for the Test
• Specify the Initial Conditions and All other Important Assumptions
• Describe the Experiment, e.g.,
  • Minimum Numbers of Samples
  • Minimum Numbers of Test Items
  • What can be Simulated, and Simulation Assumptions
  • Specify what System HW/SW will be Used
• Specify Success Criterion in terms of the Measure
An Auto Industry Test Example

3.2.x.y The vehicle shall have 95% reliability at 100,000 miles.

4.2.x.y Vehicle reliability shall be verified by Test.

--- The test shall use accelerated life testing procedures in accordance with TSP 432-1.
--- The test shall use at least 3 prototype vehicles.
--- The test shall use as data simulated mileage at failure and total simulated mileage for prototype vehicles that do not fail by the end of the test.
--- The test shall statistically process the data to calculate the probability that the vehicle provides 95% reliability at 100,000 miles.
--- The test shall succeed if the probability that the vehicle provides 95% reliability at 100,000 miles exceeds 90%.
Analysis Attributes

• Also, Almost the Same as Test
  • Specify the Measure for the Analysis
  • Specify the Initial Conditions, All other Assumptions, and Sources of Equations
  • If a Simulation, Specify the Extent of the Simulation
    • How many Repetitions (think Monte Carlo)
    • Extent and Range of Simulated Environmental Conditions to be Considered
  • Specify if System HW/SW will be Used
  • Specify Success Criterion in terms of the Measure
An Auto Industry Analysis Example

3.2.x.y The vehicle shall have 95% reliability at 100,000 miles.

4.2.x.y Vehicle reliability shall be verified by Analysis.
---- The analysis shall simulate accelerated life testing procedures, environmental conditions, and maintenance in accordance with TSP 543-2.
---- The analysis shall use 100 simulated vehicles and simulate driving for 200,000 miles.
---- The analysis shall use as data simulated mileage at failure and 200,000 miles for simulated vehicles that do not fail.
---- The analysis shall statistically process the simulated data to calculate the probability that the vehicle provides 95% reliability at 100,000 miles.
---- The verification shall succeed if the probability that the vehicle provides 95% reliability at 100,000 miles exceeds 90%.
Demonstration Attributes

- **Almost the Same as Test**
  - Specify the Measure or *Function* to be Demonstrated
  - Specify the *Initial Conditions, Environmental Conditions*, and All other Important *Assumptions*
  - Specify *Specific Directions*
  - Specify *System HW/SW* that will be Used
  - Specify if Anything (usually environment) is to be Simulated
  - Specify *Success Criterion* in terms of the Measure or *Function*
An Auto Industry Demonstration Example

3.2.x.y The vehicle shall have 95% reliability at 100,000 miles.

4.2.x.y Vehicle reliability shall be verified by Demonstration.

---- The demonstration shall use accelerated life testing procedures in accordance with TSP 432-1.

---- The demonstration shall drive 3 prototype vehicles for 100,000 miles.

---- The verification shall succeed if all 3 prototype vehicles survive to 100,000 miles.

The Method

The IC’s and Assumptions

System HW/SW

Specific Directions

Success Criterion
A Verification Requirements Practice to Avoid

• Picked Completely at Random From ISS Requirements Document Section 4
  “This Requirement shall be verified by Analysis. An Analysis of the USOS Specification shall be performed to determine the segment-level requirements that are derived from this system requirement. The verification shall be considered successful when the analysis shows that the requirements have been successfully verified.”

• Huh?
Summary

• Verification Requirements are *Extremely Important*
  • Establish Contractual Customer and Contractor Risks
  • Good Verification Requirements Can Save Project Resources
  • Good Verification Planning *Really* Can Save Project Resources
• Thanks for your Participation!
Contact Information

• I have published Numerous Papers on these topics that I would be happy to Send you
• I am always looking for new, exciting, and challenging problems to solve so I can write more papers
• Many Useful and Tailored Courses
• I can bound Quantitatively Risk to Customer, and Risk to Contractor for any Verification
• Contact Me!
  • e-mail: attwater@aol.com
  • Telephone: 208-521-2941