A Day in the Life of a Verification Requirement

Stephen J Scukanec and James R van Gaasbeek

Northrop Grumman Corporation, One Hornet Way, El Segundo, California 90245

stephen.scukanec@ngc.com; james.vangaasbeek@ngc.com

Abstract. One measure of the quality of a product requirement is that it be verifiable. Verifiability assessment is one of the exit criteria for the Systems Requirements Review and is necessary for requirement validity. Nomination of one or more verification methods (examination, analysis, demonstration or test) is often taken as the sole evidence of verifiability. A completed Verification Cross Reference Matrix is frequently considered as the final verifiability assessment and responsibility for the remainder of the verification effort is transferred to the test and evaluation and other implementing communities for completion.

Lessons learned from many Programs have shown that a more robust application of systems engineering should include the requirements engineers (with detailed knowledge of product requirement intent) working with the verification implementing organizations as the best combination to define the verification requirements. Such definition should include statement of the verification objectives, success criteria and environment. Including this information in the ”Quality Assurance” section of the requirements document allows for buy-in by the customer well in advance of implementing the verification activities. This information is used by verification personnel to generate one or more verification plans and to develop the detailed verification program. Verification requirements are planned into verification events which are executed using the proper system elements and environments. These verification requirements are key to establishing long lead verification facilities, tools and laboratories. Early definition of these requirements helps prevent facility re-designs and verification re-plans that can cause expensive delays. Finally, verification data analysis is performed, and the information compiled into verification reports certifying system product requirements compliance. This robust verification approach will provide proof of requirements satisfaction, leading to systems that meet the customers' needs at a lower life-cycle cost.

This paper is written to explore the value of well-crafted verification requirements developed early in the Program. A “Day in the Life of a Verification Requirement” shows the interaction and benefits of verification requirements to the verification execution teams. The reader will be offered a lifecycle description of the verification requirement from conception to certification.

Introduction

Requirements are fundamental to every product development activity. They form the basis for the design of the product. After implementation of the product requirements into the design, the product must be shown to operate in accordance with the product requirements. The verification process is used as the final determination of requirements compliance. Agreement between the developer and the customer that the design complies with the product requirements is the justification for payment.

Execution of the verification activity is an expensive and time consuming task. A project's verification activity is conducted through the use of verification requirements. Development of high-quality verification requirements minimizes the verification costs and the risk of schedule delays, which can occur in the verification program, resulting in saving to the overall Program bottom line.

Three reasons for early development of verification requirements have been identified [Hooks and Farry 2001]. First, by developing the verification requirements concurrently with the product requirements, it demonstrates the verifiability of the product requirements, and leads to better product requirements. Second, early assessment of the verification requirements helps to identify additional requirements needed to support the verification activities. Third, early definition of the verification requirements supports early and thorough planning of the activities, identifying the tools, equipment and facilities needed. An additional benefit derived from the early planning is that the tools, equipment and facilities can be procured or built in a timely fashion and verified, qualified, and/or accredited at or even prior to their need dates. Establishing the verification requirements baseline early reduces the possibility of downstream misinterpretations and significant changes, thereby allowing for effective programmatic control of both verification program cost and schedule. Finally, verification requirements establish an agreement between the customer and the Project on “how design requirements are to be proven”. This agreement is critical to ensure all parties are on the same page when it comes to showing design compliance.

The authors recommend that verification requirements be developed in concert with the product requirements, and that the verification requirements be developed by the domain experts who develop the product requirements. The place of verification in the development Program is discussed in the next section, followed by a discussion of product and verification requirements. The remaining sections address the life cycle of the verification requirements.

Definition of Verification and Validation

An important aspect of systems engineering is to “language the Program” [Ring 2000]. In like manner, it is important that the terms to be used in this paper be defined. The terms “verification” and “validation” are often used differently in different contexts. For the purpose of this paper, we shall define requirements verification and validation and product verification and validation. See Table 1.

Table 1. Definition of Terms

Verification	Are the requirements right? Do they meet the basic quality criteria (e.g., correct, complete)	Is there objective evidence that the product satisfies the requirements?
	Are the requirements the right requirements, i.e., do they	Does the product, when operated by representative operators, in the representative

Validation	properly represent the customer need?	operational environment, satisfy the customer needs?
	Requirements	Product

Requirements verification and validation must take place as the requirements are discovered to avoid costly rework later in the development cycle. Verification of requirements is best accomplished by writing good requirements from the start. This has been addressed many times over the years (Hooks 1990, Hooks 1993, RWG 2010). Requirements validation is performed through several mechanisms, including customer-contractor detailed reviews, modeling and simulation. Requirements validation can also be accomplished through application of the Continuous Early Validation (CEaVa) method (Larsen and Buede 2002).

Product verification and validation are performed once the product begins to be realized and sufficient portions of the product exist upon which to apply the verification and validation procedures. Product verification is the subject of this paper. Product validation follows product verification, and is usually performed by the customer or end user.

Verification in Product Development

The development of a product has been represented by the "Development Vee" diagram. Such a diagram is shown in Figure 1 (based upon [Forsberg, Mooz and Cotterman 2000]).

In this representation of the product development activities, problem definition, solution discovery and design occur on the left-hand side, in the "Decomposition and Definition Downstroke". Implementation of the system occurs across the bottom of the Vee, and integration and verification occur on the right-hand side, in the "Integration and Recomposition Upstroke", culminating in validation of the system preparatory to full-scale production, deployment and operation. The arrows in the center of the Vee show that activities are performed on the "upstroke" to verify that the products available at that point in the integration comply with the specifications produced during the corresponding steps in the "downstroke". As an example, subsystems whose specifications were developed during preliminary design are verified during the subsystem integration step. Verification comprises a significant portion of the development activity and is crucial to successful qualification of the product development. The underlying assumption of this development model is that good Product requirements have been developed, along with their corresponding verification requirements. At each level, verification requirements must be established consistent with the configuration established by the product requirements. That is, system requirements must be accompanied by system verification requirements and so on.

Figure 2 illustrates the life cycle of a verification requirement. Verification requirements play a part in the design requirements, verification planning, execution and report and certification phases of the design development and certification activities.

Key points to remember across the verification lifecycle are that:

- verification events never satisfy the design requirements. They instead satisfy the related verification requirements. Completion of the verification activities are based on the verification requirements attributes not the product requirement attributes. Verification is conducted against verification requirements agreed by both the customer and the developer;

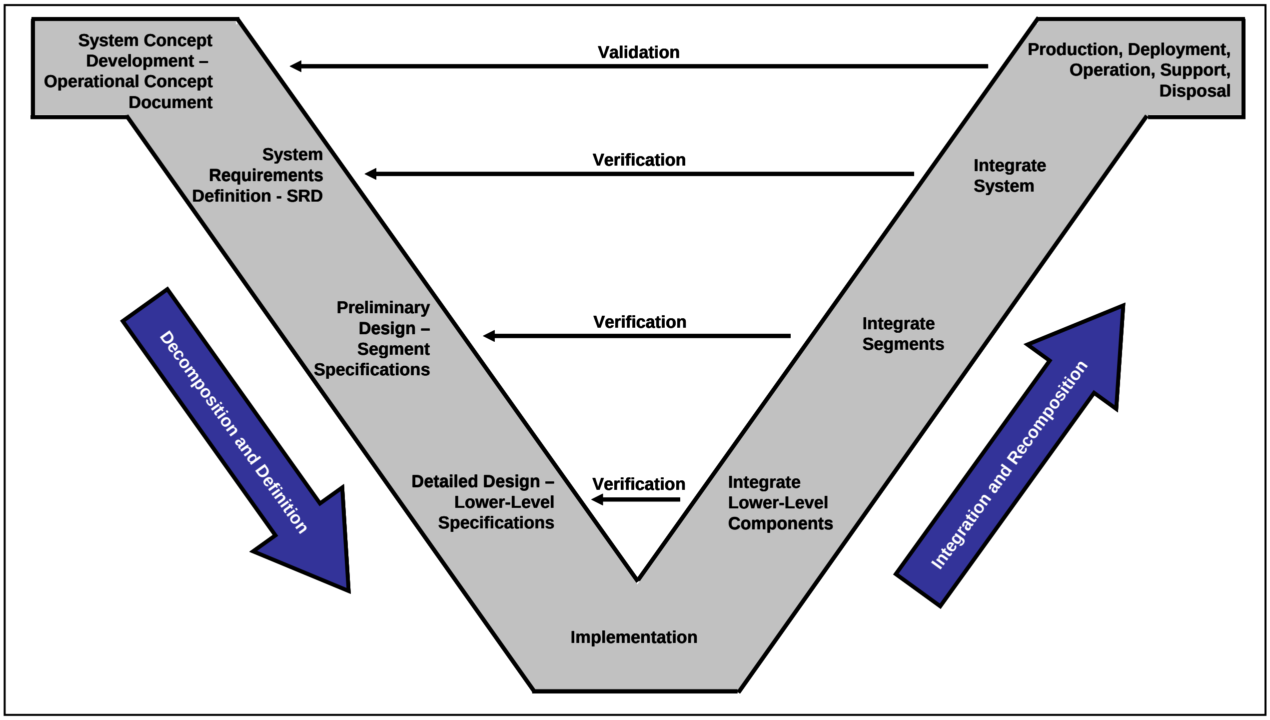


Figure 1. The Development Vee Diagram.

The Verification Requirement Life-Cycle

- product requirements development is not complete until the verification requirements set for the product requirements are completed and agreed. Simply placing an "X" in the verification cross reference matrix under one of the acceptable program verification methods is not sufficient evidence that a product requirement is verifiable;
- the culmination of verification activities of the product requirements result in the verified product;
- verification execution begins with an agreed set of verification requirements and their derived Verification Information Sheets (VISs). A Verification Information Sheet is used by verification teams in planning the events to be executed during the verification program. Verification events are planned to be executed in an operationally-representative environment based on a collection of verification requirements sharing similar configurations, circumstances and success criteria; and
- experience with large programs (e.g., the B-2, Joint Strike Fighter) has shown that, on average, approximately three verification requirements are needed to capture the verification of a particular product requirement.

This last point is not obvious. As an example verification of the communications performance requirements on an aircraft will usually require a test of the system in its operational environment. This results in verification by test. In addition, laboratory testing of the communications equipment under nominal conditions, connected within the system environment, would result in a second verification task. And, finally, analysis may be required to ensure that the communications system can operate under adverse, but untestable, conditions, resulting in a third verification method for that single product requirement. This one system level product requirement must have three verification activities performed to determine its compliance. As a consequence, the verification section of the specification is larger than the section describing the product requirements, and the Verification Cross Reference Matrix (VCRM) for the single product requirement will show the link to three verification requirements.

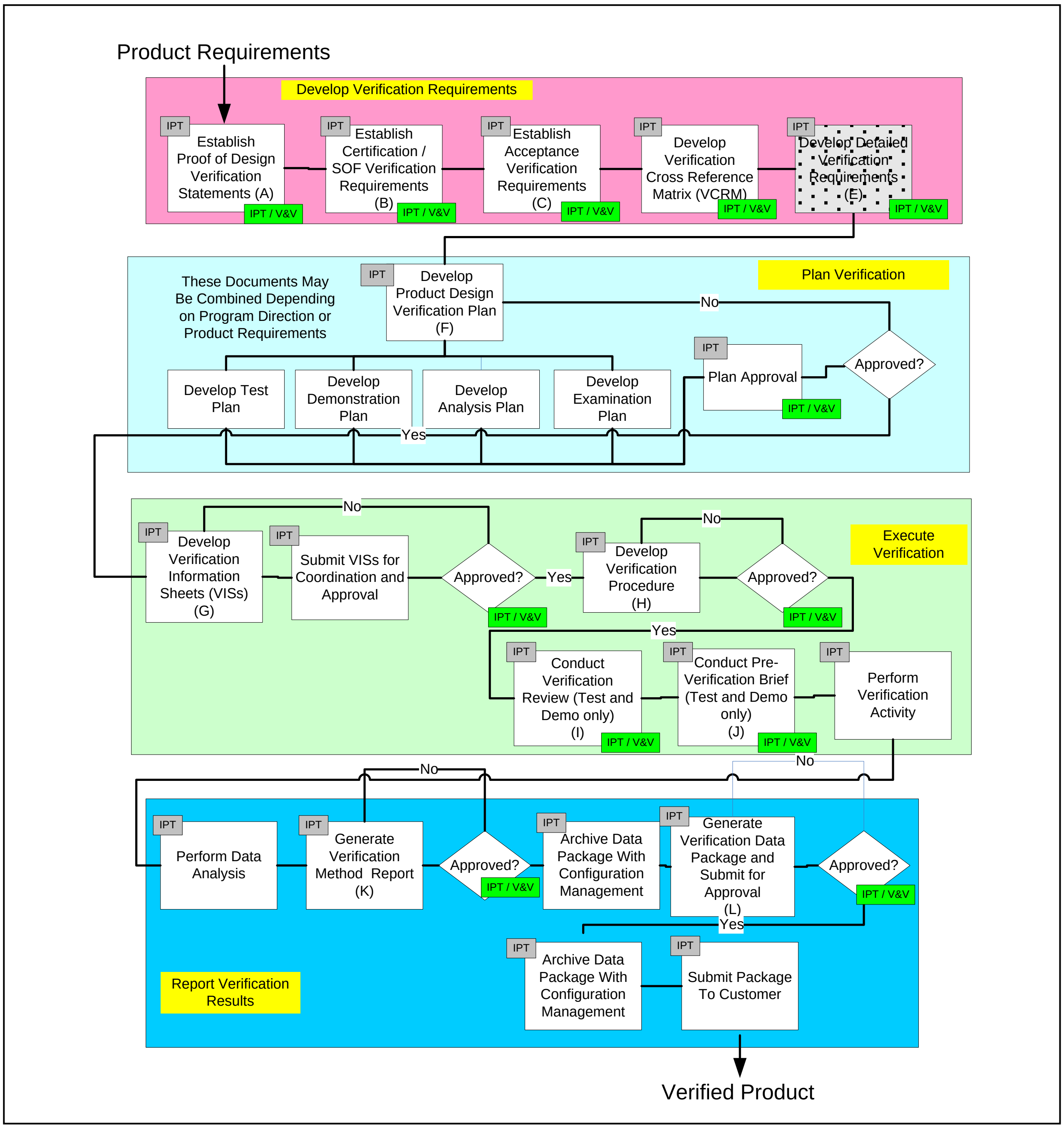


Figure 2. The Verification Requirement Life-Cycle.

The four steps in the verification life-cycle shown in Figure 2 are described below.

Develop Verification Requirements. The verification requirements process begins with the assessment of the product requirements by three key program functions. Systems engineering ensures that the requirement translates the operational needs into an executable design criteria. The design team assesses the requirement for feasibility. And finally, the verification team assesses the verifiability of the requirement.

1. Product Requirements

Several papers and books have been written describing the methods for development of a good requirement (Hooks and Farry 2001, Alexander and Stevens 2002, and Hull, Jackson and Dick 2005). Unambiguous, concise, traceable and, of course, verifiable are some of the qualities of a good requirement.

Verifiability is not discussed in any great detail among the requirements references. Other than mentioning that a verification method must be assigned to a requirement, the verifiability assessment of a requirement seems to remain an afterthought. Documentation of the requirements' satisfaction is often left to test personnel, as they know how to test the products. However, one can only assert that a product requirement is verifiable when product requirements are generated and accompanied by a set of verification requirements.

Investment in developing the verification requirement at the start of the program is crucial to ensure good product requirements are created. Experience with large programs has shown that up to 80% of the product requirements generated are rewritten as a result of the development of the verification requirements for that product requirement. Merely placing an "X" in the VCRM does not improve this statistic.

A well-crafted series of agreed verification requirements assures that a requirement is verifiable. As only the domain-specialist author of a product requirement understands the requirement’s intent, only he or she can properly specify the verification requirement(s) that show satisfaction of the requirement.

Development of a good verification requirement starts with a good product requirement. Examples for use in this paper are:

- PR-1: LRU Markings
  The product line-replaceable units shall be marked in accordance with MIL-STD-130M.
- PR-2: Operational Availability
  The product shall have an operational availability (A_0) of 97.5% at IOC.

- PR-3: LRU Accessibility
  Each product line-replaceable unit shall be able to be removed and replaced without removing any other item or displacing any cables.
- PR-4: Recovery Force Communication
  The product shall provide a communications system capable of communicating with the recovery forces pre- and post- landing

2. Verification Requirements

After the product requirement has been evaluated, the next step is to get an agreement by the verification organization, systems engineering and the requirement's author to the primary objective, method, environment and success criteria needed to satisfy the design requirement. The documentation of these attributes creates a foundation for agreed verification requirements. Completion of this step ensures verification feasibility, allows for evaluation of alternatives to support programmatic goals, assesses completeness of the verification requirements and, finally, produces a verifiability assessment to be used for the various design reviews' entry or exit criteria. Agreement on the verification requirements is much easier when the conditions are understood by all parties.

Five key attributes are associated with verification requirements: verification objectives, methods, environments, success criterion and (if required) special conditions. (See Figure 3.)

1. The Objective: establishes purpose of the verification.
2. The Method: establishes the verification methods. This can include examination, analysis, demonstration, and test. (Verification is not just test.) It should be noted that with the advent of model-based systems engineering, model-based acquisition, and the increasing use of modeling and simulation, many practitioners have proposed the use of modeling and simulation as a fifth verification method. While this change is well underway, this paper will use the four classical methods, in accordance with [MIL-STD-961E 2003]. Note also that these four methods may be divided into sub methods. For example, the test method may be subdivided into a laboratory test method and a flight test method.
3. Environment: what are the experimental conditions under which the item will be verified? This is to include environmental conditions such as temperature, pressure, altitude etc., as well as the operational environment needed to execute the verification. This could be environmental characteristics such as mountainous terrain, underwater, specific rainfall amounts, electromagnetic interference conditions, or other such environmental constraints necessary to show the operation of the system to the intent of the design criteria.
4. Special Conditions: these are the unique attributes necessary to show that the product verification is conducted under conditions that represent worst case conditions if possible. The angle at which a line replaceable unit must be positioned relative to rainfall would be considered a special condition, e.g., place the line replaceable unit at 45° angle to the rain path and conduct functional testing periodically (at a minimum of five minutes) throughout the course of the test.
5. Success Criteria: this is the data set to be collected to show that the design criterion has

been satisfied. Tolerances and ranges must be specified to ensure that data collection and verification criteria can be achieved.

These five attributes comprise the necessary criteria needed for a good verification requirement.

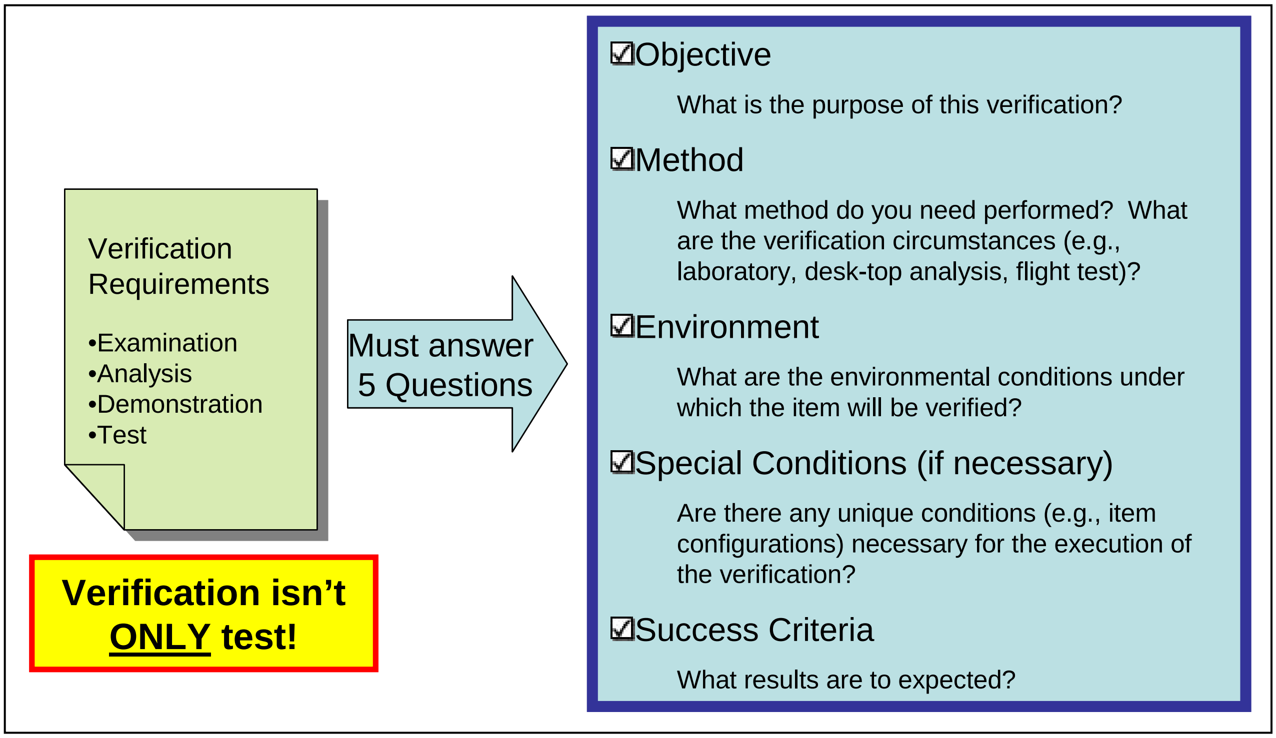


Figure 3. Characteristics of a Verification Requirement.

Verification requirements for the example product requirements provided above are:

- VR-1E: Compliance of product markings shall be verified by examination of design drawings at the LRU supplier's location prior to the LRU CDR. The examination will show that each marking on the LRU conforms to MIL-STD-130M.
- VR-2A: The product operational availability shall be calculated using the results of the Government-accredited contractor-developed reliability and maintainability analyses performed during the design in conjunction with the Design Reference Missions documented in Report XXXX. The analysis will show that the product, in its operational environment, supported with its support equipment and personnel, across all missions, will have an operational availability of at least 97.5%.
- VR-3D: Removal and replacement of all LRUs shall be demonstrated on the aircraft to show that each LRU can be removed and replaced without removing any other items or moving any cables.
- VR-4D: Perform a demonstration of the communications systems capability to provide voice and beacon communications with recovery forces pre and post landing while within a representative environment and using a production equipment configuration. The demonstration will show the ability for the communications systems to verbally communicate with the on board communication production configuration equipment. The

demonstration will also show beacon tracking within communication ranges established by TBD.

- VR-4T: Perform Integrated System Test of the communications system capability to provide a voice communications and beacon with recovery forces pre and post landing within an integrated hardware / software environment. Testing will show that the communications system can transmit and receive audio at frequencies and ranges (power) represented by standard ground recovery force communications devices as defined in TBD.

Let's examine the sample verification requirements. "Perform Integrated System Test of the communications system capability to provide a voice communications and beacon with recovery forces pre and post landing within an integrated hardware / software environment. Testing will show that the communications system can transmit and receive audio at frequencies and ranges (power) represented by standard ground recovery force communications devices as defined in TBD." Under this example the phrase "[p]erform Integrated System Test of the communications system capability to provide a voice communications and beacon with recovery forces pre and post landing within an integrated hardware / software environment" represents the verification objective. (See Figure 4.) The product requirement author is asking for the verification that shows proper ground communication can be performed by the communications system with a ground command team. The phrase "[p]erform Integrated System Test … within an integrated hardware / software environment" represents the verification method chosen for this requirement. In this case, testing is the required verification method and this test needs to be conducted while the communication system is integrated in its operational system. Additionally the term "integrated hardware / software environment" represents a laboratory environment. So this one phrase offers information with regard to the method, configuration and test facility needed to perform this verification. The environment that is specified for this verification consists of a phrase "the communications system can transmit and receive audio at frequencies and ranges represented by standard ground recovery force communications devices". This phrase provides the verification team with the conditions under which the verification must occur. This phase provides us with the information that the test must be conducted with ground recovery force communication devices. The definition of these ground recovery force communication devices however is to be determined. The use of "TBD" is perfectly acceptable in the early development of a verification requirement. TBD's are a promissory note to the verification team that additional information will be developed as the requirement is refined. "TBDs" are acceptable when the remainder of the verification definition remains intact. TBDs should not be used to postpone work; rather they are used to postpone the definition of details which will not affect the verification planning activities.

Finally the phrase "can transmit and receive audio at frequencies and ranges represented by standard ground force communication devices defined in TBD" represents a success criteria. Once again the TBD is used to defer the definition of the success criteria until such time it is known. Agreements between the implementing organization and the verification author has to when the TBD's are to be completed must be performed to ensure the completed verification statement is provided on time. One final note, it is important to remember that a verification requirements is written to the configuration established by the title of the document. That is to say "The title of the document is the item under test". If the verification requirements cannot be written against the configuration established by the title of the document, it is a good indication that the requirement been written may be at the wrong level. Upon completion of the generation of a verification requirement is determined that the configuration required to satisfy the verification is not consistent with the document titled either the verification statement is an error or the requirement is located at the wrong specification level.

3. Verification Cross Reference Matrix

A verification cross reference matrix summarizing the verification methods selected for each product is generated upon completion of the development of the verification requirements. The verification cross reference matrix is a common deliverable during the early part of many programs. Many programs are satisfied simply with a verification cross reference matrix as an indication of the verification program. As discussed earlier, a simple “X” in a column on the VCRM is insufficient to establish the true intent of the verification program. The development of the VCRM is the last step in performing the verification requirements process. It should never be the only product defining the verification requirements. The VCRM is then used by systems engineering as a tracking tool for obtaining the completion criteria of each of the product requirements.

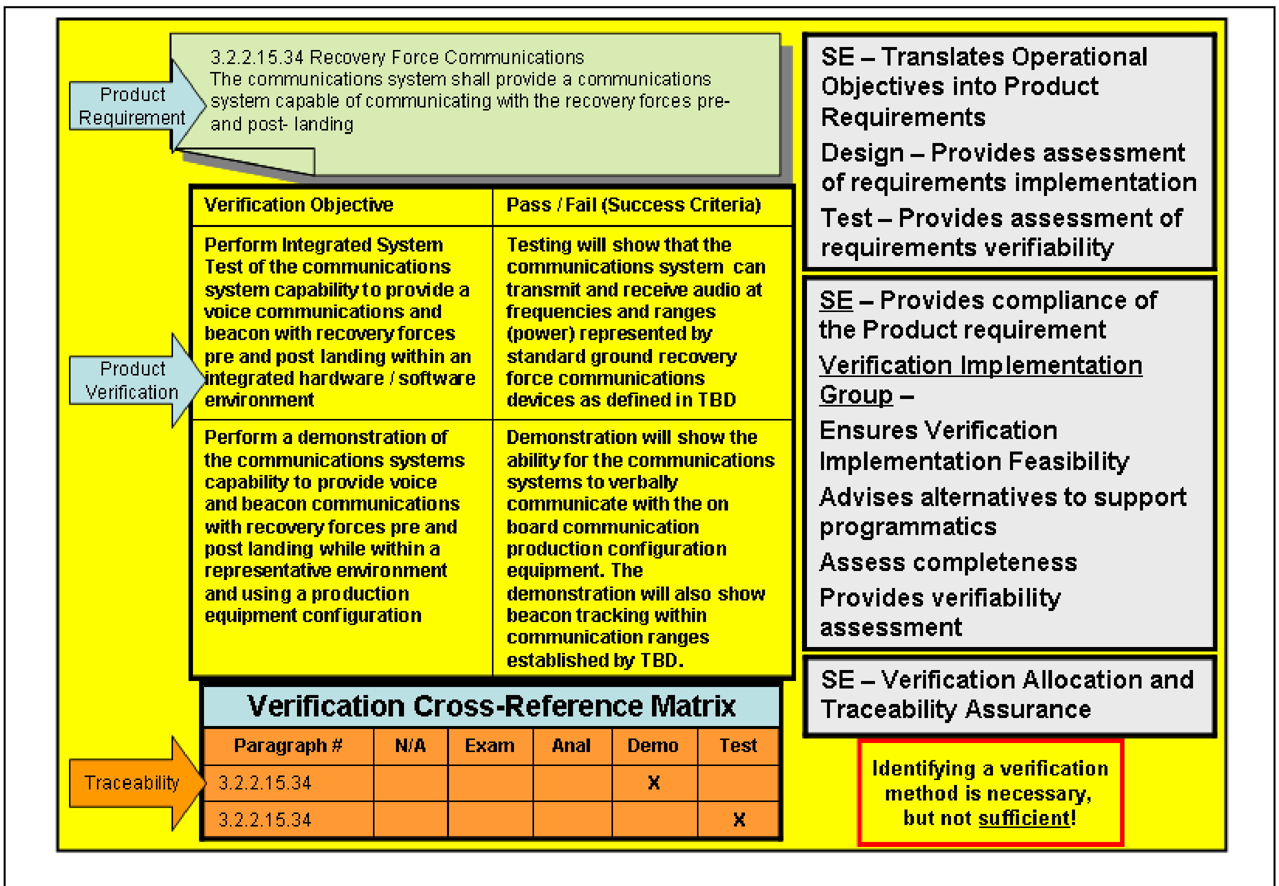


Figure 4. Verification Requirements Analysis.

Plan Verification. The combination of the product requirements, the verification requirements and the verification cross reference matrix comprise the bulk of the product specification. The verification cross reference matrix is an effective tool to share traceability of the verification activity back to the product requirements. From there the path of the verification requirements splits. The data included in the VCRM is used for traceability from inception to completion of the verification events. The second path of the verification requirement finds its way into the Master Verification Plan (MVP). (See Figure 5.) In developing the MVP, verification requirements are combined with other verification requirements of like method, environment and configuration to develop the verification events which will be executed in the verification program. These verification events, whose source is multiple verification requirements, are combined in the VIS (see the explanation of VIS above) with additional information such as a detailed configuration and its relationship to the production configuration, prerequisites, constraints, relevant environmental conditions, pass/fail criteria and the collected data sets necessary to prove such criteria and any sequencing of the verification activity required to complete the definition of a verification event. It is imperative to ensure that the product customer agrees to the implementation of the verification events early in the Program. Such an agreement is evidence that the proper translation of the verification requirements into verification events has occurred.

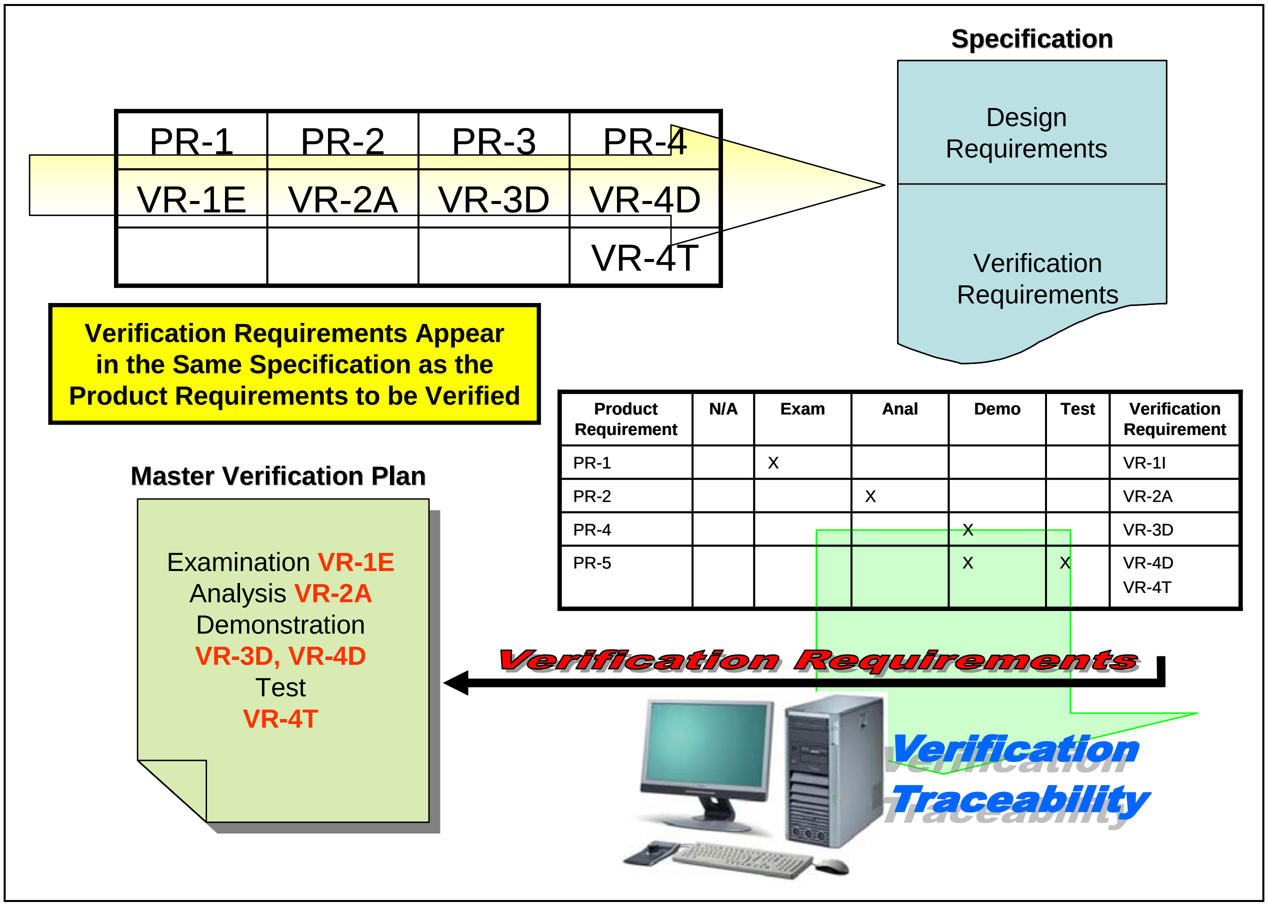


Figure 5. Verification Requirements and the VCRM are used for Traceability and Flowdown During MVP Development.

A relatively new idea associated with the verification requirements is the use of verification modeling. Verification modeling is a technique similar to that used in requirements traceability where the environment, method and success criteria attributes of the verification requirements are traced from parent to child. This technique ensures that the operational environment is preserved throughout the requirement thread. By ensuring the appropriate verification attributes of method, environment and success criteria are appropriately maintained from system to segment to subsystem to component, the program can be assured that no individual component within the system is verified in its operational environment for the first time at the system level. Verification modeling also allows for attributes to be assigned to the various verification events to allow for easy tracking and traceability of risk mitigation tasks, fault tree analysis and any other programmatic attributes assigned to the verification program. This modeling technique requires discipline and manpower be expended throughout the course of the program but the benefits of having this technique in place far outweigh the costs associated with the activity.

As an example, it was determined during the development of a major weapons system that a navigation system using the global positioning system was required. At the system level, the requirements for the navigation system's operational environment included a duration of twelve hours continuous navigation. As the product requirements were decomposed, the environment conditions set at the system level were never maintained throughout the requirements decomposition. When the product specification for the GPS system was developed, it was determined that the supplier's method for verification of the product was sufficient and therefore left to the supplier for determination.

The manufacturer of the global positioning system was more than capable of satisfying the product level specifications imposed on their contract through a series of verifications that never required operation of the GPS for the duration specified at the system level. Tests were conducted to show constellation roll over, leap seconds transition, hemispherical transitions and other attributes of a GPS navigation system. It was decided late in the GPS test program that that the 12 hour operational constraints needed to be examined by the supplier prior to completing the acceptance of the GPS units. The supplier did not accept this assessment and was reluctant to conduct additional testing without compensation. It was determined that the lack of verification flowdown of the environmental conditions was not the supplier's error, but an error by the prime contractor. After an equitable settlement, the GPS manufacturer modified the test program to assess their product under the conditions of the operational system level environment, and the supplier found errors within their system. Extensive modification to the unit was required. Had the problem not been detected until system-level integration and verification, the expense of making the repair, and the retest, would have been far more significant than the original equitable adjustment made to the supplier team

Had the program performed verification modeling through the flowdown of the requirements set and the associated verification cost and schedule impacts associated with the repair could have been avoided. Verification modeling, if applied conscientiously throughout the program, does save cost and schedule. This technique also supports a concept of ensuring that realistic testing is performed throughout the development cycle of a product. "More realistic testing is required prior to product delivery" was a goal stated by the Honorable Charles McQueary, Director, Operational Test and Evaluation, Department of Defense [McQueary 2007]. Dr McQueary provided several examples of fielded systems which required "ad hoc" modifications by the user in the field. He attributes such in-field design changes to the lack of realistic testing during design, development, test and evaluation. Since verification requirements contain the environmental information necessary to ensure realistic testing it only makes sense to implement the technique which will ensure the system level environments are appropriately captured "womb to tomb".

Detailed verification plans are generated once the verification events have been defined and the verification modeling completed. (See Figures 6 and 7.)

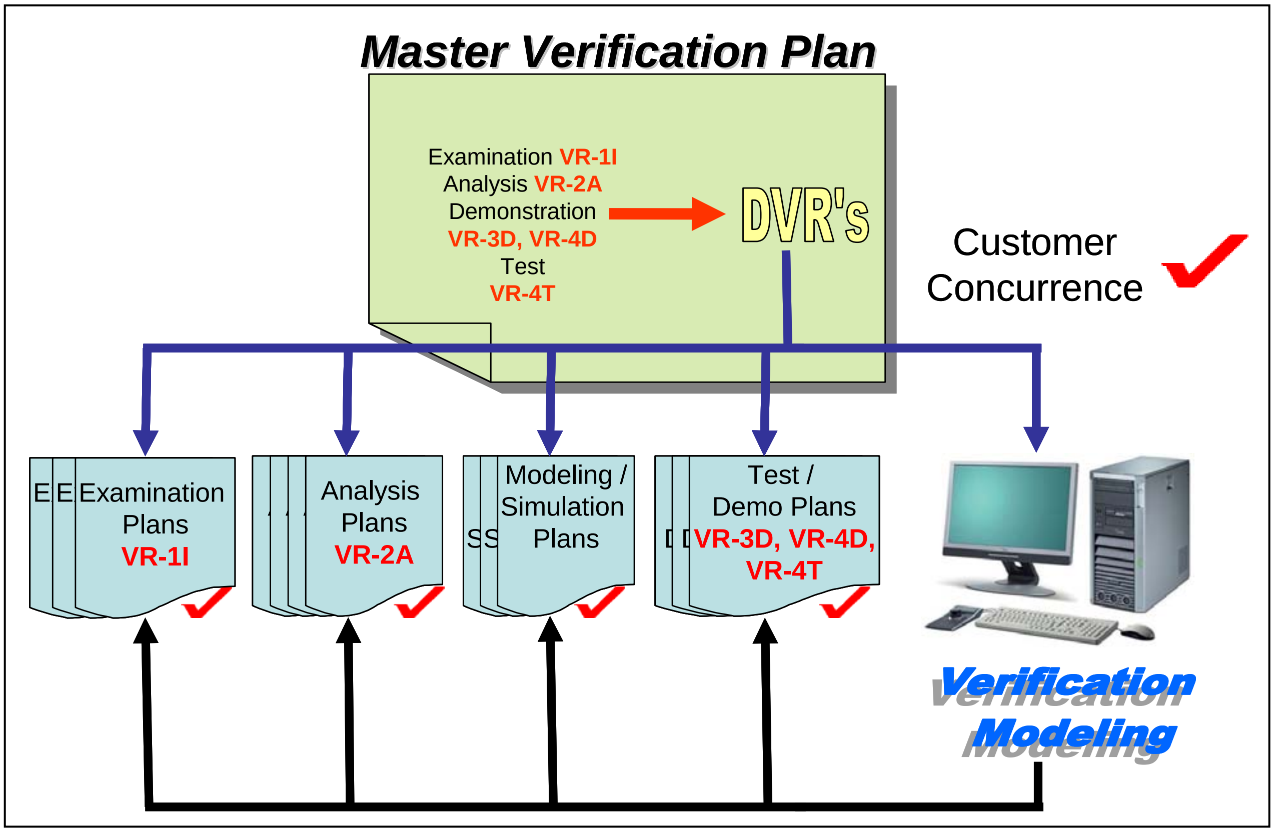


Figure 6. Generate Verification Plans.

Execute Verification. Verification execution spans the tasks from the development of the Verification Information Sheets to the completion of the task defined by the VIS. Activities typically performed during the verification execution phase depend on the verification method being used. Activities for test and demonstration are somewhat different than those associated with analysis and examination. Each organization responsible for the execution of a verification program will have their own methods for completing this verification phase but ultimately the completion criterion is the collection of the data necessary to prove the success criteria established by the verification requirement. From development of procedures, analysis plans and examination procedures, to the documentation of the results that they generate, the purpose of the execution phase is to collect and or produce the data needed to ensure the verification requirement and its intent have been examined in accordance with the agreed to objective, environment, special conditions, and success criteria of the verification requirement. (See Figure 8.)

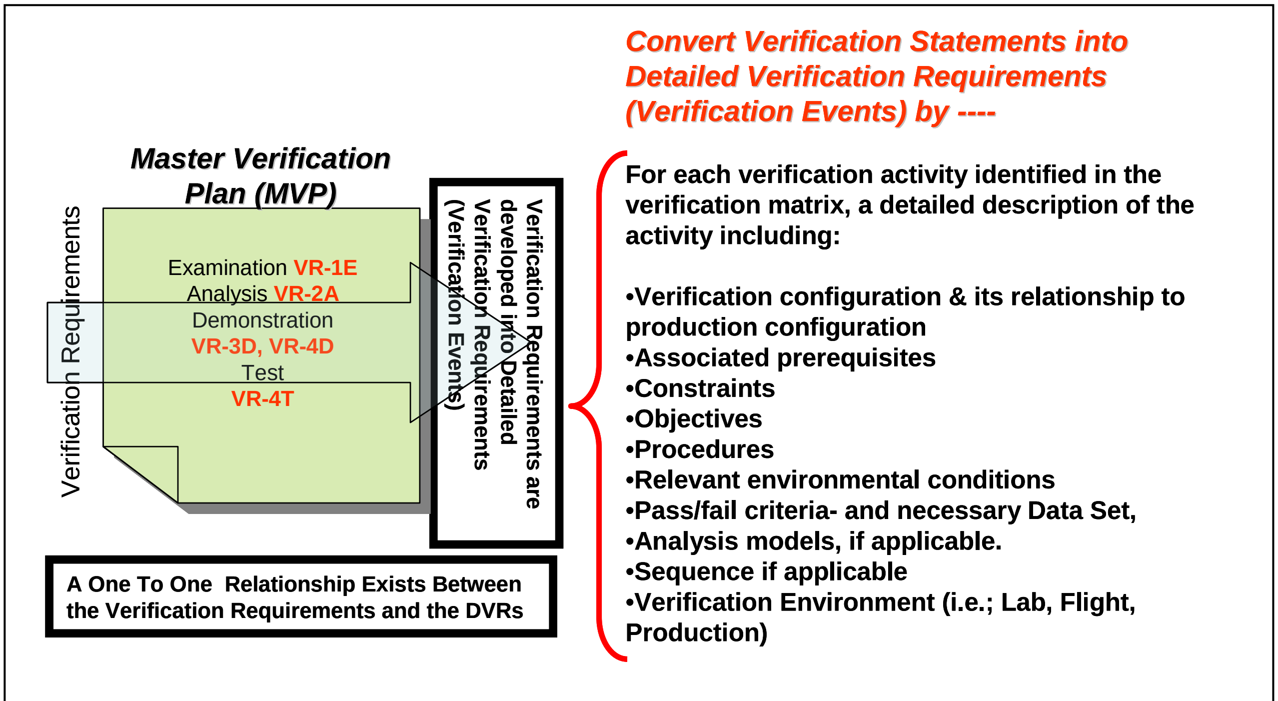
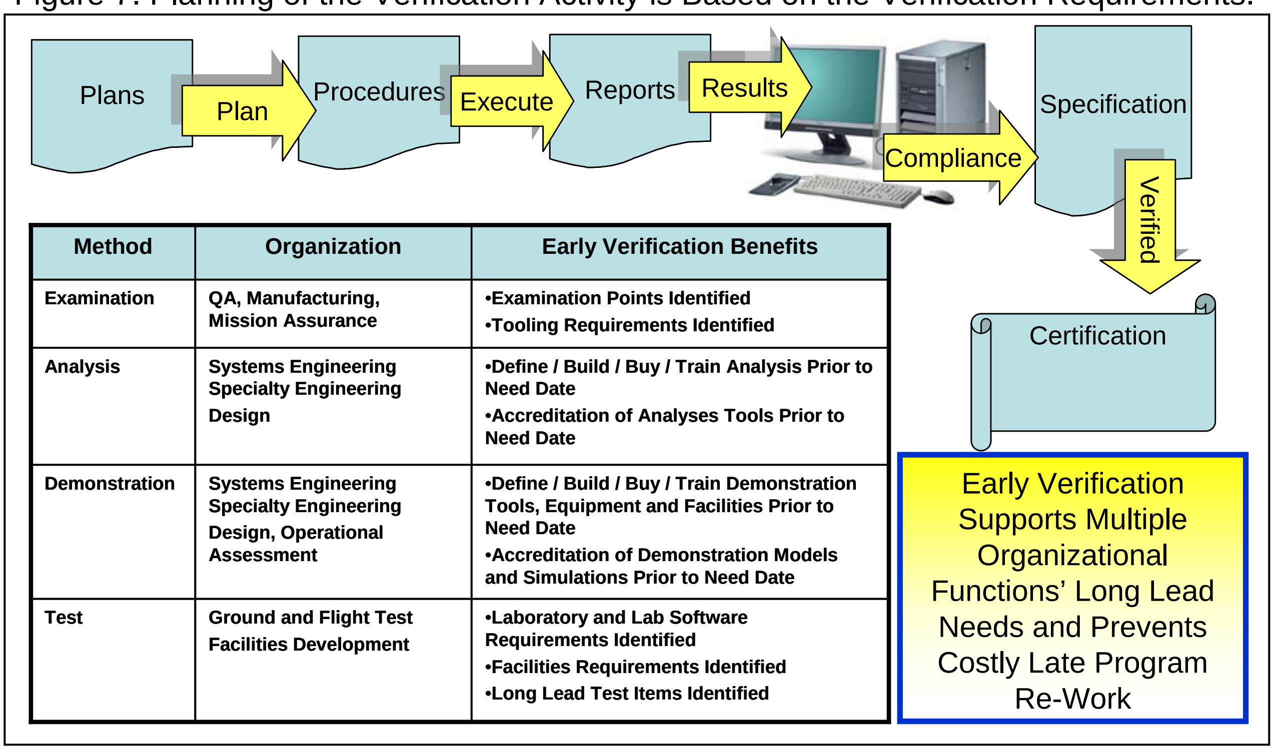


Figure 7. Planning of the Verification Activity is Based on the Verification Requirements.



Method	Organization	Early Verification Benefits
Examination	QA, Manufacturing, Mission Assurance	•Examination Points Identified •Tooling Requirements Identified
Analysis	Systems Engineering Specialty Engineering Design	•Define / Build / Buy / Train Analysis Prior to Need Date •Accreditation of Analyses Tools Prior to Need Date
Demonstration	Systems Engineering Specialty Engineering Design, Operational Assessment	•Define / Build / Buy / Train Demonstration Tools, Equipment and Facilities Prior to Need Date •Accreditation of Demonstration Models and Simulations Prior to Need Date
Test	Ground and Flight Test Facilities Development	•Laboratory and Lab Software Requirements Identified •Facilities Requirements Identified •Long Lead Test Items Identified

Figure 8. Verification Execution Activities.

Report Verification. The final phase in the life of a verification requirement is the reporting of the results through a process which documents the agreement between the customer and the supplier team has indeed been satisfied. Assembly of the collected data into a traceable and documented package is the necessary proof that the job is finished. During the auditing and certification process all of the necessary data representing the culmination of the verification program exercised for each requirement is reviewed and presented to the customer for final concurrence that the product requirements have been satisfied. Each program will perform this final step in order to declare that the program requirement set has been satisfied in accordance with its own agreed to certification and auditing process.

Summary and Conclusion

The verification program is the thread from start to finish of any program. Starting with the definition of the agreed-to completion criteria to the proof that the criteria have been met, the verification requirements connect the product requirements through the design and are finally used as the check that the proof of product requirement satisfaction has been accomplished. This paper has demonstrated the importance of the verification requirement and its role throughout the program lifecycle. The authors have shown that investment in verification up front in the program execution will provide significant benefits across the program lifecycle. This paper has outlined the attributes of a good verification requirement, and the role of the verification requirements in each phase of the program. The authors have attempted to provide concrete examples of a verification requirement, its use and the failures which can result from limiting the early verification investment to an "X" in a matrix.

Key points to remember as outlined by the assertions made by this paper are that:

- early verification benefits the entire program from Systems Engineering to Program Management to the Customer;
- the Verification Cross Reference Matrix does not establish the verification program and should never be used as the proof that a requirement is verifiable;
- a good verification requirement contains 5 key attributes - objective, method, environment, special conditions and success criteria;
- verification requirements are written against the configuration of the item specified in the document, and requirements owners can determine if the verification requirement has been written at the right level of decomposition;
- verification modeling is a good way to ensure operational environmental constraints are incorporated at all levels of the verification program, helping to ensure realistic verification occurs throughout the program;
- the Master Verification Plan is the documented agreement of verification events which the program and the customer agree that upon completion will establish the data set which is to be used to concluded a requirement has been satisfied; and
- early program verification definition helps establish the boundaries for the long lead procurement tasks each program faces.

The overall benefit of a verification program which follows the process described in this paper is outlined in Figure 9. It should be emphasized that early verification is an effective cost avoidance approach for any program and that programs which utilize this approach have a much better chance of executing the program on time and on budget while meeting customer needs the first time.

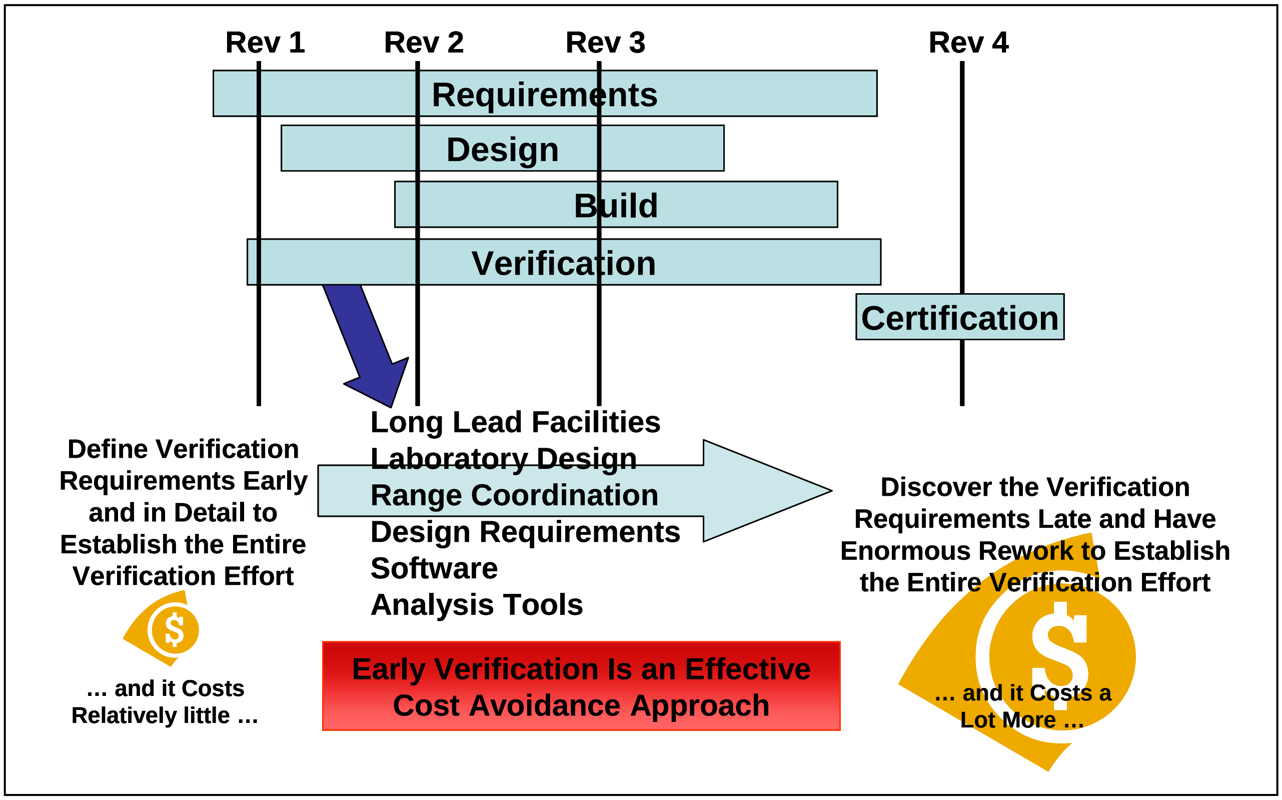


Figure 9. Early Verification Requirements Development Saves Costs and Minimizes Delays.

Nomenclature

Analysis	an element of verification that utilizes established technical or mathematical models or simulations, algorithms, charts, graphs, circuit diagrams, or other scientific principles and procedures to provide evidence that stated requirements were met. [MIL-STD 961E 2003]
CDR	Critical Design Review
Demonstration	an element of verification which generally denotes the actual operation of an item to provide evidence that the required functions were accomplished under specific scenarios. The items may be instrumented and performance monitored.) [MIL-STD 961E 2003]
Examination	an element of verification and inspection that is generally

nondestructive and typically includes the use of sight, hearing, smell, touch, and taste; simple physical manipulation; mechanical and electrical gauging and measurement. [Examination has previously been known as Inspection.)] [MIL-STD 961E 2003]

IOC	Initial Operational Capability
LRU	Line-Replaceable Unit
Objective Evidence	data supporting the existence or verity of something [ANSI/ISO/ASQ Q9000-2000].
Requirement	a statement that identifies a system, product or process characteristic or constraint, which is unambiguous, can be verified, and is deemed necessary for stakeholder acceptability [INCOSE Handbook, Version 3, 2006].
TBD	To Be Determined
Test	an element of verification in which scientific principles and procedures are applied to determine the properties or functional capabilities of items. [MIL-STD 961E 2003]
Validation	(Requirements) confirmation that the requirements individually and as a group meet the requirements quality factors e.g., feasible, consistent, complete, correct) (Are the requirements right?) (Product) confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled [ANSI/ISO/ASQ Q9000-2000].
VCRM	Verification Cross Reference Matrix
Verification	(Requirements) confirmation that the requirements properly represent the customer needs. (Are they the right requirements?) (Product) confirmation, through the provision of objective evidence, that specified requirements have been fulfilled [ANSI/ISO/ASQ Q9000-2000].
VIS	Verification Information Sheet

References

American National Standards Institute	ANSI/ISO/ASQ Q9000-2000, Quality management systems - Fundamentals and vocabulary, December 2000.
Alexander, I F, and Stevens, R	Writing Better Requirements, Addison-Wesley, London, 2002.
Department of Defense	DEPARTMENT OF DEFENSE STANDARD PRACTICE DEFENSE AND PROGRAM-UNIQUE SPECIFICATIONS FORMAT AND CONTENT, MIL-STD-961E, 1August 2003.
Forsberg, K, Mooz, H and Cotterman, H,	Visualizing Project Management, 2nd Edition, J Wiley and Sons, New York, 2000.
Hooks, I	Why Johnny Can't Write Requirements, presented to the AIAA, 1990, last accessed 8 March 2010 at http://www.complianceautomation.com,
Hooks, I	Writing Good Requirements, presented at the Third International Symposium of the International Council on Systems Engineering, 1993, last accessed 8 March 2010 at http://www.complianceautomation.com,
Hooks, I and Farry, K,	Customer-Centered Products, AMACOM, New York, 2001.
Hull, E, Jackson, K, and Dick, J,	Requirements Engineering, 2nd Edition, Springer, London, 2005.
International Council on Systems Engineering	Systems Engineering Handbook, Version 3, INCOSE-TP-2003-002-03, June, 2006.
Larsen, R F and Buede, D M	Theoretical Framework for the Continuous Early Validation (CEaVa) Method, Systems Engineering, 5, 3, 2002, pp 223 - 241
McQueary, Charles,	Remarks during the Opening Plenary, 2008 National Defense Industries Association Systems Engineering Conference, San Diego, California, 23 October, 2007.
Ring, J	Discovering the Value of Systems Engineering, presented at the Tenth International Symposium of the International Council on Systems Engineering, Minneapolis, MN, 2000
Requirements Working Group	RWG Guide to Writing Requirements, to be released as an INCOSE Technical Product, 2010

Biographies

Jim van Gaasbeek has 35 years experience analyzing and developing rotary-wing and fixed-wing aircraft, launch vehicles and spacecraft, both in the United States and European defense environments. Beginning as a rotor aeroservoelastician, his career has progressed with experience in constructive and virtual simulation, accident investigation, vehicle-management system design and systems engineering, concentrating in risk management and requirements development, management and verification.

Steve Scukanec has spent over 25 years as an Aerospace Engineer on various complex programs including the B-2, B-2 Long Term Software Support and the F-35. With a focus on test and evaluation, Steve has been able to participate in programs from inception to completion. This experience over several programs has provided Steve with a rare understanding of the values of a well executed Verification program as well as the problems caused by the lack of one. His experience as a “requirements generator”, “requirements customer”, “requirements manager” and verifier gives him insight into the lifecycle of a requirement and a large lessons learned knowledge base.