

A Socratic Approach to Optimizing Aerospace Manufacturing Costs

Lisa Sivertson

Raytheon Technologies Corporation, Waltham, United States

Email address:

lmsivertson@rtx.com

To cite this article:

Lisa Sivertson. A Socratic Approach to Optimizing Aerospace Manufacturing Costs. *International Journal of Systems Engineering*. Vol. 4, No. 1, 2020, pp. 7-11. doi: 10.11648/j.ijse.20200401.12

Received: June 2, 2020; **Accepted:** June 17, 2020; **Published:** July 4, 2020

Abstract: In aerospace development and manufacturing environments, the cost of tests contributes a significant portion to the overall program costs. An extensive test program results in increased costs and unforeseen delays in fielding needed products and technologies. On average test represents approximately 30% of overall costs. The lack of a well thought out test strategy developed early and maintained through the entire program lifecycle results in high operational field failures, increased test equipment and unit production costs, delays in unit integration, redundant manufacturing tests, and poor transition into production as well as an increase in program risks. This paper describes the concept of an evolving program test strategy and the role of a Test Architect to achieve the goal of reducing test costs across the entire program lifecycle. Defining a test strategy results in clearly structured test plan and architecture, optimized test event planning and comprehensive test artifacts early in the program lifecycle. As a Subject Matter Expert, the Test Architect sets and drives the test strategy ensuring an overall test program is optimized and aligned across three phases of development: User Operations, Development and Production. To achieve a robust test strategy, the Test Architect uses a Socratic approach to question why a test needs to be performed, increasing the likelihood of executing a successful development test program, facilitating a seamless transition into production and optimizing the support of the deployed product.

Keywords: Strategy, Test, Manufacturing, Architecture, Aerospace

1. Introduction

To be competitive in the aerospace defense industry, companies are driven to have low costs and short development and manufacturing schedules to win government contracts. Since a large portion of the product cost is due to test, it is essential to have a comprehensive, well-planned test approach and strategy that considers all aspects of an overall test program – development testing, field or operational users' tests as well as production testing. Lam indicates test costs contribute to product costs as high as 50 percent [1]. In addition, Turino recognizes product costs for electronics have risen ranging from 35 to 55 percent [2]. The Defense Acquisition organization found that most weapon programs proceed with low levels of knowledge in technology maturity, design stability and incomplete test programs resulting in increased cost and delayed schedules [3]. A good test strategy drives a reduction in production test content, only conducts necessary development tests to meet program objectives, and

develops essential field tests that are required to demonstrate operational readiness [4]. A Socratic approach is straightforward and questions why a specific test or group of tests need to be performed. It also helps to determine whether extensive testing can be performed as a simple manufacturing test rather than setting test limits to the extreme outer limits of the product's specification or environments like in design verification tests. This approach guides discovery of information from the team and is more likely to reach adaptive conclusions that challenge the standard approach of product testing [5]. A Test Architect with test knowledge in development, production and field test activities drives this approach in the development of an evolving test strategy and asks probing questions which results in a low cost, streamlined and effective test program.

2. Test Architect Role

In order to successfully use this approach, a Test Architect

is a key contributor and needs to be part of any development or production test program. A Test Architect is technically accountable for development and execution of the program test strategy ensuring all test related activities are properly coordinated and well-structured across the program. This person is a Subject Matter Expert (SME) in test and is responsible for developing, implementing and executing a consumable test strategy for the program. The Test Architect is responsible for all test related strategies and activities during user operations, engineering development, assembly and test. The Test Architect sets and ensures the test strategy is optimized to meet program objectives and defines metrics to gauge progress, creates a seamless transition between engineering development and production test program phases and optimizes production assembly and test solutions to reduce product unit costs. The Test Architect helps to set the culture of questioning the overall test program, defines the required tests to prove the product or system meet the requirements and optimizes the test flows as the product transitions into production. As depicted below (Figure 1), a key factor to ensuring a successful test strategy execution is the communication of the defined test strategy to all those engaged on the program as well as receiving and incorporating feedback from individuals executing the strategy. The Test Architect drives the team to determine what decisions need to be made based on testing, what test data is required to support decisions and establishes an effective test strategy to get the necessary data.

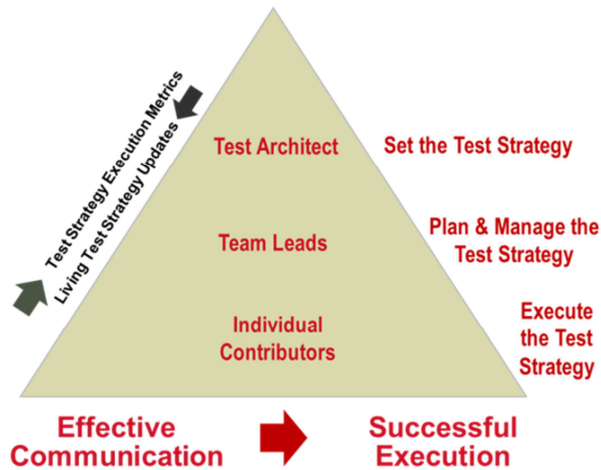


Figure 1. Test Strategy Communication Flow.

3. Approach

A strategy is a plan of action designed to achieve an overall objective, therefore, in the development of aerospace product design it is important to have a clear product testing path to prove the developed product meets all the customer's objectives and requirements, can be produced within the required cost and schedule, and optimizes the testing throughout the entire product lifecycle. A Socratic approach employs cooperative, challenging dialogue to stimulate critical thinking based on asking and answering questions to

draw out ideas.

A good test strategy is characterized by diminished operational failures in the field, little or no production down time, reduced test equipment and product costs, shortened integration times, reduction in rework and successful transitions to production [6]. Therefore, a Socratic approach to optimizing a test program involves questioning the need for conducting a specific test or groups of tests and analyzing the appropriate number and levels of tests during the manufacturing and testing process. It is the Test Architect's responsibility to ask probing questions and generate the critical dialog in order to get the program test team to question the importance and purpose of performing a specific test or set of tests and what type of data will be gained when performing the tests. Understanding program objectives and goals are critical knowledge necessary to obtain and become inputs when defining the test strategy. It is important to have defined costs goals (developmental and unit production costs), Built In Test (BIT) coverage requirements [7], Manufacturing and Test Readiness Level (MRL/TRL) requirements [8] as well as an understanding of the overall program strategy. The ultimate goal is to only conduct tests that will demonstrate a capability or prove a system will meet requirements and to remove any tests that will not accomplish these objectives, therefore, reduces the overall costs.

A test strategy is developed during early phases of a program, continues to be maintained through the end of the program and contains items such as:

- a) Product background information - description of the product and any pertinent contract related information
- b) Concept of Operations (ConOps) – description of how the product is going to be used
- c) Program strategies, requirements and assumptions driving test strategy (e.g., where the product will be built, logistics concepts)
- d) Cost requirements
- e) Required unit production costs, rates and schedule
- f) Test specific requirements (e.g., BIT, required developmental and field testing)
- g) Master Phasing Schedule - the overall schedule for the program (major milestones) and how testing fits into that schedule
- h) System Block Diagram
- i) Test Principles - the philosophies driving the decisions around test (e.g., test the product at the earliest point or at the lowest level, test using embedded test only)
- j) Lifecycle Phase Approach - assumptions about how the various phases of testing evolve and relate to each other.
- k) Assembly & Test Flows for engineering and production testing
- l) Maintenance Concept for Operational Availability & Field Upgrade Testing - description of the logistics concept as it relates to test

Figure 2 depicts an approach to defining an overarching test strategy that uses high level program inputs, such as objective and costs, and builds on foundational test architecture, environments, standard events and artifacts

covering the product lifecycle. The lifecycle considers all aspects of how the product will be tested and used during development, production and user operations which are depicted as three pillars of a test strategy approach. Typical questions the Test Architect must ask to determine the overarching test strategy include: Will the customer be running the same tests as those being conducted in the manufacturing process? What data will be analyzed during development testing? Is this the same data needed during production or when the customer is using the product? How

does additional testing affect the overall cost of manufacturing the product? Could a test be eliminated and not affect the ability to meet the product specifications and meet the customer's objectives?

As a program matures, a program strategy continually changes based on program events and results, therefore, the program test strategy must be a living artifact that changes based on contract updates, schedule execution, material deliveries, supplier data and test event results.

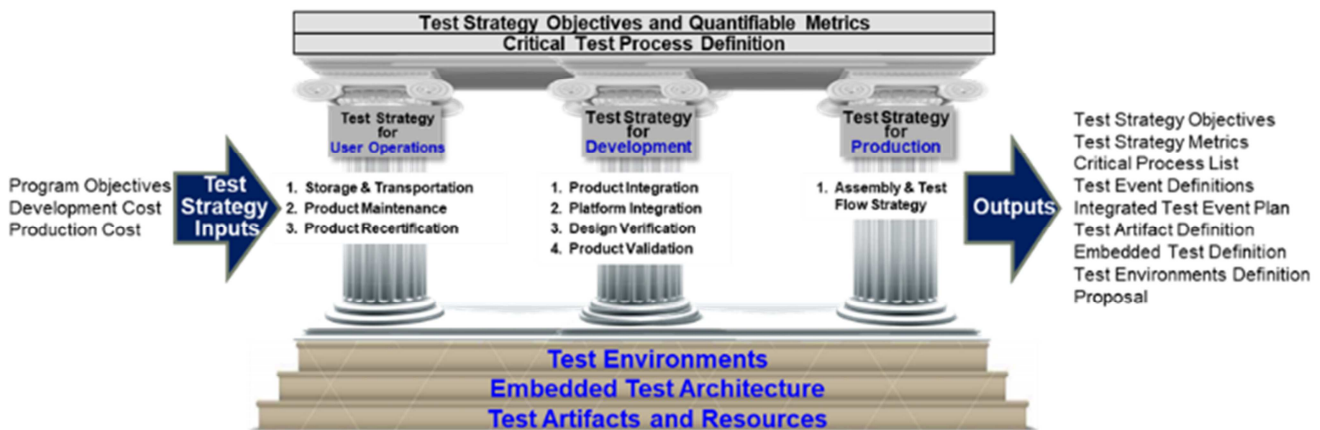


Figure 2. Test Strategy Approach.

The User Operations test strategy pillar takes into account how the product is used, tested and evaluated once deployed. This is the stage when the product is in the customer's hands being used in operational situations as well as being maintained, transported and recertified in the field. User operations events include storage and transportation to forward bases; tactical utilization and all environments associated with product during weapon system usage; maintenance and shelf-life extension program; recertification; and possibly demilitarization. User operations costs are reduced by implementing an effective BIT philosophy to eliminate or extend period between recertification, structuring the product design with a modular architecture allowing for simple method to add enhanced or new technology [9, 10], and allowing for field reprogramming capability to enable algorithm performance enhancements while the product is deployed.

Typical questions that need to be asked when defining user operation test requirements are: How will the product be stored and maintained? Are there any extreme environmental requirements? Will the product be powered in the field? Will the unit be recertified in the field? If so, how will it be recertified? Will Built In Test (BIT) be used to understand the condition of fielded units?

It is important to understand the product capability that the customer is expecting when in use by defining Concept of Operations (ConOps) for all aspects of usage [11]. For example, there may be an expectation that the product is only to be tested once, when it is first received, and then expected to operate as required when given a command which could be months or years later. Or in other situations, the product may

be powered continuously with BIT being run periodically in an extremely hot environment which dictates that the product would need to be cooled while operating. These type of requirements need to be understood when developing the product and how it will be used in testing when verifying that the product meets the customer requirements.



Figure 3. User Operations Test Strategy Pillar Artifacts.

The Development test strategy pillar is used to make program data driven decisions as a result of conducting a specific test or analysis ensuring success of product testing and deployment. Development testing is used to drive technology maturity for intended product use, verifies interfaces including electrical, mechanical, and messaging,

verifies design against requirements, establishes margin over environments, and validates that the product meets customer's needs. It also defines detailed product integration, verification and validation activities which include qualification tests, flight tests and reliability growth testing. This is part of the process where the product design is maturing and evaluation testing occurs. In this phase, cost drivers lead to poor performance and low reliability including late failures in test program due to immature technology; interface problems accounting for unreliable margin performance and measurement errors; and poor testability for one-shot items [12]. Issues in any one of these areas results in the product not meeting the customer's needs. Typical questions asked when defining the development test strategy pillar include: What type of testing will be required to gain access to a government range, facility or platform? What testing will be completed to verify and validate the product design meets all the requirements? What level of assembly are tests conducted? How many system level tests will be needed? Can simulations or analyses be used to verify requirements instead of performing a costly test? Are requirements verifiable by test within the current capabilities? Are there any special tests or analyses that need to be performed post qualification to verify the requirements have been met [13]?



Figure 4. Development Test Strategy Pillar Artifacts.

For example, if the product will be transported over rough roads or may experience large shocks levels it would be important to understand and test to these extreme levels during the development tests before the product is delivered to the customer. The developmental test event portion of the test strategy has collection points for knowledge gained and is linked to the test events, as well as a master test phasing plan which includes all integration, verification, validation and test events. Working through the types of testing needed and data to be collected assists in making program decisions based on explicit data required to meet program objectives which results in the smallest number of developmental tests being conducted and reduces the overall test program costs.

The Production test strategy pillar defines the recurring process for assembling and testing the product. In many cases,

the product is in this stage for many years and the most program costs are incurred if assembly and test are not optimized [14]. This pillar starts with the product being used for production verification and validation of the assembly and testing processes and continues to mature as it enters full rate production. If the production test strategy is not thought through early in the program, costs will increase as a result of performing too many tests, not decreasing test complexity at higher levels of assembly, lack of test requirement funneling based on environmental exposure, design verification testing being performed in production and poor First Pass Yield due to a) poor design margins, b) poor test equipment, c) poor design, d) poor test limits.



Figure 5. Production Test Strategy Pillar Artifacts.

Typical questions asked for this stage of the product lifecycle are: Have Design Verification Tests (DVT) and product qualification tests been removed? Does testing verify that a product has been assembled correctly? Are only key parameters being tested? Are the tests being performed at the lowest level to catch failures early? Having a plan to transition a product into production with a focus on only testing to make sure the product is assembled correctly helps to reduce the amount of tests conducted during product assembly decreasing recurring production costs. Thus, it is critical to make decisions early about the assembly and test flow ensuring that manufacturing tests are conducted at the correct time to limit rework and re-tests. For example, when developing the assembly and test for a stack of circuit cards, it is recommended that each circuit card be functional tested with an environment screen. Once these lower level tests have proven the circuits cards are working, then the stack of cards are assembled and only tested to confirm the stack of circuit cards are assembled correctly. There is no need to run all the detailed lower level tests again or test under environments again.

A test strategy and its three pillars are built on a strong foundation of embedded test architectures, environments, artifacts and resources. An embedded test architecture is a group of test capabilities, resident and fully integrated into the product design via both hardware and software. Utilizing BIT to perform functional tests is a solution to be considered [15].

An embedded test architecture provides the product with a test capability that allows for easy testing during every phase of the product's life-cycle from the design verification testing during product development; acceptance testing in manufacturing; and logistics operations during field utilization. A test environment specification is used to define the test equipment requirements needed to test all mission critical functionality at a given assembly level. Functional requirements define what are the key product requirements for all subsystems integrated into a product and are used to ensure that a product is properly assembled and will function correctly. Performance requirements specify how a product function interacts with another in order to perform a tactical mission. Physical requirements are characteristics and attributes of the system. These requirements are derived and traced from the requirements defined in each pillar of the test strategy. All these requirements and artifacts are necessary to define a good test strategy.

Defining a test strategy results in clear definition of the test plan and architecture, optimized test event planning and detailed test artifacts early in the program lifecycle. The test strategy and architecture describes clear, distinct objectives that align with the program goals, shows that the requirements are clearly met, have metrics that are tracked on a regular frequency and demonstrates understanding of the critical processes that must be in place in order to achieve the test objectives. Test strategy artifacts or outputs include an integrated test equipment plan, test equipment specifications and requirements, embedded test plan, and test environment definitions. These outputs are used in proposal support, for customer communications as well as a way to align the entire program around the test strategy, objectives and path forward.

4. Conclusion

This paper describes how to reduce costs with an evolving test strategy using the role of Test Architect. Development of a test strategy that continually matures is the result of a comprehensive, well-planned Socratic approach using challenging dialogue to stimulate critical thinking based on questions to draw out ideas. A Test Architect drives technical accountability and coordination of the test strategy and all related test events, oversees the execution of a test program resulting in successful product development and demonstrations, and helps to transition a product into production resulting in an effective test program and low product costs.

References

- [1] H. Lam. (2004, July). New design-to-test software strategies accelerate time-to-market. In *Electronics Manufacturing Technology Symposium*, 2004. IEEE/CPMT/SEMI 29th International (pp. 140-143). IEEE.
- [2] J. Turino. (2012). *Design to test: a definitive guide for electronic design, manufacture, and service*. Springer Science & Business Media.
- [3] Defense Acquisitions: Assessments of Selected Major Weapon Programs. GAO-07-406. Washington, DC.: March 2007
- [4] R. Mahoney. "Integrating Manufacturing Test Strategy with Manufacturing Production Strategy." *1997 IEEE Autotestcon Proceedings AUTOTESTCON '97. IEEE Systems Readiness Technology Conference. Systems Readiness Supporting Global Needs and Awareness in the 21st Century*, 1997, doi: 10.1109/autest.1997.633651.
- [5] N. Singh. "The Socratic Process / Guided Discovery." *Handbook of Evidence-Based Practices in Intellectual and Developmental Disabilities*, Springer, 2018, p. 291.
- [6] Manas, J. and Guise, L., "Systems engineering for test: Implementation of test strategy & architecture at raytheon missile systems," *2013 IEEE International Systems Conference (SysCon)*, Orlando, FL, 2013, pp. 305-311, doi: 10.1109/SysCon.2013.6549898.
- [7] J. Smith and D. Lowenstein. "Built in Test - Coverage and Diagnostics." *2009 IEEE Autotestcon*, 2009, doi: 10.1109/autest.2009.5314056.
- [8] M. J. Ward, et al. "A Readiness Level Approach to Manufacturing Technology Development in the Aerospace Sector: an Industrial Approach." *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 226, no. 3, 2011, pp. 547-552., doi: 10.1177/0954405411418753.
- [9] R Marshall, et al. "Enhanced Product Realisation through Modular Design: an Example of Product/Process Integration." *Figshare*, Loughborough University, 7 Aug. 2019, repository.lboro.ac.uk/articles/Enhanced_product_realisation_through_modular_design_an_example_of_product_process_integration/9340238.
- [10] Nair, C. "Modular Test Architectures for the Aerospace Industry." *Proceedings, IEEE AUTOTESTCON*, 2002, doi: 10.1109/autest.2002.1047895.
- [11] N. James. "Concept of operations and the DoD architecture framework." Massachusetts Institute of Technology. 2018.
- [12] G. Kasouf and D. Weiss, "An Integrated Missile Reliability Growth Program," *Annual Reliability and Maintainability Symposium, 1984. Proceedings.*, San Francisco, CA, USA, 1984, pp. 465-470, doi: 10.1109/RAMS.1984.764336.
- [13] P. Jaramillo, M. Rascon, C. Adams, and E. Jauregui. (2020). Fundamental Principles, Processes, and Roles of Environmental Qualification Test Strategy for Complex Engineered Systems. *J Info Tech Soft Eng* 10: 265. doi: 10.24105/2165-7866.10.265.
- [14] K. Brindley. "Test Strategies." *Automatic Test Equipment*. Newnes, 1991, p. 38.
- [15] R. Drees and N. Young. (2004). Role of BIT in support system maintenance and availability. *IEEE Aerospace and Electronic Systems Magazine*, 19 (8), 3-7.