AEROSPACE REPORT NO.
TOR-2014-02537-REV A

The Test Like You Fly Process Guide for Space, Launch, and Ground Systems

September 30, 2016

Julia D. White and Lindsay G. Tilney
Systems Integration and Test Office
Mission Assurance Subdivision

Prepared for:

Space and Missile Systems Center
Air Force Space Command
483 N. Aviation Blvd.
El Segundo, CA 90245-2808

Contract No. FA8802-14-C-0001

Authorized by: Space Systems Group

Developed in conjunction with Government and Industry contributions as part of the U.S. Space Programs Mission Assurance Improvement Workshop.

AEROSPACE
Assuring Mission Success

Foreword

Much of the foundational material in this document is taken from the original technical operating report (TOR), *Test Like You Fly: Assessment and Implementation Process* (Aerospace Report No. TOR-2010(8591)-6), and a chapter written for the second edition of the *Space Vehicle Test and Evaluation Handbook* [3].

This material has been presented to industry in tutorial form for Ground Systems Architectures Workshops (GSAWs) and Aerospace Testing Seminars (ATSs) since 2009. The most recent tutorial was updated and given during the 29th ATS [1].

This guide addresses all types of system acquisitions. It reflects a more mature and evolved formulation of the test-like-you-fly (TLYF) process as it has been introduced to industry and applied to space, launch, and ground systems.

This revision to TOR-2014-02537 results from industry feedback, participation in TLYF process assessments, and review from subject matter experts involved in the Mission Assurance Improvement Workshop (MAIW).

Acknowledgments

The authors acknowledge the following individuals who participated in a recent review of the material as part of the MAIW effort:

Daniel J. Byrne	The Aerospace Corporation
Jeff Grandilli	Boeing Space and Intelligence Systems
Elizabeth. J. Klein-Lebbink	The Aerospace Corporation
Geoffrey A. Larsen	The Aerospace Corporation
David Shelton	Lockheed Martin Space Systems Company
David. Spiegelthal	The Aerospace Corporation
Beth E. Springsteen	The Aerospace Corporation
Charles P. Wright	The Aerospace Corporation

The authors also gratefully acknowledge several Aerospace champions who have supported the process and its evolution:

Bruce L. Arnheim
Frank L. Knight
Gail A. Johnson-Roth
William F. Tosney
Jacqueline M. Wyrwitzke

Contents

Figures

Tables

1. Scope

1.1 Introduction

Test like you fly (TLYF) is a term that has progressed from undefined jargon to a comprehensive assessment and implementation process within the systems engineering context. It is most widely encountered in space system testing. When a string of mission failures occurred, the post-mortem lessons learned fed directly into the process with the intent to avoid failures for future systems. Because it was codified to address failures, it became evident that the process needed to go beyond the test domain and include systems engineering disciplines as well. It has developed further by performing program assessments and workshops with government and industry communities of practice. The fundamental emphasis of this systems engineering approach is broader than "test" and starts with "like you fly" (LYF).

Hard and expensive lessons have driven the development of the TLYF process. These lessons have helped form the steps described in this document. The intent of implementing the TLYF process is to minimize the likelihood of mission failures. The process takes advantage of related systems engineering, test artifacts, and methods, and helps focus those efforts to more effective ends. This process allows a program to tailor the scope of mission-oriented testing from a risk management perspective. An understanding of what can go seriously wrong when attempting to execute mission activities will help set priorities for operationally realistic LYF tests.

This guide covers the following topics:

- Philosophical underpinnings for the TLYF process—the basis for the process and what makes it distinct from traditional systems engineering practices and types of testing

- The acquisition strategy implications for adopting the TLYF process

- The steps for *implementing* the TLYF process—practical ways to build operationally realistic LYF tests that are perceptive in detecting mission-critical flaws

- How and when to effectively apply and implement the TLYF process at any program development phase

1.2 Application

The TLYF process has key implications for acquisition strategy, requirements definition, interactive ground and space product development, systems engineering, fault analysis, and risk management. The process applies to complete systems (i.e., space, ground, launch, and user segments). It can also be used at lower levels of integration (i.e., element, subsystem, unit), as well as higher levels (i.e., system of systems and enterprise). The term "like you fly" may not seem fitting to ground-centric systems, so for instructional purposes the reader is asked to replace the phrase with a meaningful statement such as "like you operate" or "like you execute the mission" when applying this process to those particular systems. The process and methodology may be tailored according to the acquisition mission risk class assignment (i.e., Classes A-D[1]), but does not necessarily contribute to those mission risk class assignments.

The purpose of this document is to provide step-by-step guidance on how to implement the TLYF process. For simplicity, the material herein assumes that the TLYF process is on contract (i.e., its methods

[1] A classification criterion developed to communicate the acceptable level of risk for contractual requirements and risk management strategy implementation with Class A having the least risk tolerance and Class D having a higher risk acceptance.

and products are identified in the statement of work) and the program is at the beginning of the acquisition lifecycle. Considerations for adding the TLYF process to an existing program will also be addressed later (see section 5).

1.3 Background

Historically, typical systems engineering practices utilize testing as a method to verify requirements. Based on mission failures, it seems that alternate processes are needed to catch flaws preflight. **Requirements verification is necessary, but insufficient**. Tests performed under non-mission conditions (e.g., mission sequence, timeline, concurrency, etc.) with non-flight hardware (HW) and/or software (SW), with incomplete or previous (pre-repair) configurations, or without the last preflight SW build/version will miss the flaws that can only occur under mission conditions.

Demonstrating that a system can successfully perform its mission (fly/operate) is fundamentally different than demonstrating that the system meets requirements. Post-mortem analyses of failed missions show that systems were not being tested in an operationally realistic manner. These deviations from operational realism have led to loss or degradation of mission.

The TLYF process described here is based on mission failure root cause investigations. Each step was created with the goal of preventing mission flaws from occurring before a system becomes operational. A primary lesson about incorporating TLYF is that program leadership must recognize the collaborative nature between testing and systems engineering processes. It is ineffective to have the sole responsibility for TLYF in the test organization. The TLYF process does not begin with test; it begins with understanding the system and mission execution. It requires participation from systems engineers, mission designers, operations personnel, flight HW and SW developers, integration and test personnel, ground HW and SW engineers, users, and more [1]. The bottom line is “TLYF is a team sport.”

2. Definitions

Key definitions are provided for clarity and context in the discussion of the TLYF process.

Term	Description
Automated fault management	A part of space vehicle, launch vehicle, or ground systems (SW and HW) designed to detect selected criteria (off-nominal telemetry and other indicators) and respond without human intervention, putting the vehicle, ground, or selected equipment into a predefined state.
Component	A functional item that is viewed as a complete and separate entity for purposes of manufacturing, maintenance, or record keeping. (See definitions of “subassembly” and “unit.”)
Computer software configuration item (CSCI) or software Item (SI)	An aggregation of SW that satisfies an end-use function and is designated for purposes of specification, interfacing, qualification testing, configuration management, or other purposes. An SI is comprised of one or more software units. An SI is sometimes called a computer software configuration item [2].
Contingency	A planned action, including action decision guidance, to be executed by the ground operations team in response to defined criteria (off-nominal telemetry values and other indicators) concerning the state of the space vehicle, launch vehicle, or ground system.
Critical event	An event that can consist of any of the following types of events that are necessary for a successful mission: first-time events, critical reoccurring events (i.e., daily nominal operations), and critical situations (i.e., transitions between events).
Critical mission contributors	All items (HW and SW) and conditions (initial conditions, dependencies/pre-requisites, sequence, timing, interactions) that have failure consequences leading to end of mission or severe mission degradation.
Element	A complete, integrated set of subsystems capable of accomplishing an operational role or function, such as navigation. It is the configuration item delivered by a single contractor (payload, bus, space vehicle, launch vehicle, or ground station).
End-to-end configuration	A list of the HW and SW items and interfaces included in the mission event and settings (i.e., on/off), including configuration identification and build versions.
Fidelity	The accuracy with which the system reproduces the characteristics and behavior of the object of interest. In general, a closer behavior to flight is considered higher fidelity [4].
Fidelity—hardware fidelity	The accuracy of the device baseline against the flight unit. Utilizing non-flight parts or other part substitutions reduces the hardware fidelity [4].
Fidelity—interface fidelity	The accuracy of the electrical, physical, or software boundary between two or more components. For some purposes, the interface fidelity is more important than the overall fidelity of the component [4].
Fidelity—simulation software fidelity	The accuracy of the simulation in behaving like the component/environment it represents. This can be in multiple different regards such as timing, precision, functionality, etc. The baseline measurement for this fidelity is against the real component or environment that is being simulated [4].
Ground segment	A network of Earth stations and user terminals that provides applications and services to end users. May include ground station, mission control center, user

Term	Description
	elements, backup mission control center, and an external command and control center.
Ground system	A collection of ground elements with common interfaces (e.g., user, ground station, control center, and communication system). Ground systems may be used to support space and/or launch vehicles and may have multiple architectural levels (e.g., elements or segments).
Hardware unit	A separately testable component of a hardware design that is part of a subsystem.
LYF test	(See definition of “operationally realistic test.”)
Mission activity	A discrete event within a mission phase. A sequence of mission activities may constitute a mission scenario or phase. Mission activity is at the lowest level in the mission pyramid (see Figure 14).
Mission characteristics	Includes all aspects for mission operations/execution in terms of components, conditions, interfaces, transitions, transactions, processes, and environments. Typical mission characteristic classes are (1) end-to-end configuration; (2) time, sequence, and timeline; (3) environments (internal, ascent, space, command, ground and telemetry); and (4) any relevant operational conditions that are present during the mission (people, processes, and procedures).
Mission coverage	Consists of phases, transitions, environments, and events in an end-to-end system configuration (i.e., combination of HW/SW and data when functioning as an integrated system); acceptance of nominal mission inputs; execution of nominal mission functions; and production of mission outputs according to the typical mission rhythm, timelines, and sequences resulting in end-user goals (products, services, and timeliness).
Mission-critical failure	Defined as any of the following: • Failure leading to inability to meet/achieve mission objective (e.g., payload, spacecraft bus, or ground element is no longer capable of supporting mission objectives) • Inability to meet minimum performance specifications • Degradation of condition where the trend indicates a loss of mission before mean mission duration (MMD) or design life • Repetitive transient condition(s) that, uncorrected, would lead to an unacceptable loss of mission performance, data, or services (e.g., satellite with processor susceptibility to single-event upsets in orbit with mean time to upset that is much less than mean time to recovery from upset) • Inability to correctly generate, process, and/or transmit data to requester (e.g., end user of the mission data)
Mission-critical fault analysis	A pre-launch analysis that examines a system’s operational timeline (discrete critical events and transitions) and focuses on what could go wrong with the mission (i.e., mission-ending failures). “It is a top-down analysis of potential failures that is performed before a system is fielded or flown” [12]. It addresses faults, which may involve multiple and concurrent items (HW and SW), that may occur as the system executes its mission. It may use common fault analysis tools (e.g., fault trees, Ishikawa diagrams) to develop fault paths and contributors, and is an adjunct to a failure modes and effects analysis (FMEA), or to a failure modes, criticality, and effects analysis (FMECA).
Mission objective	A brief description of each event and how the related HW and SW product(s) or product component fulfills its intended use when placed in its intended environment. The objective includes what will determine the pass/fail criteria for that event in the context of the primary mission goals.

Term	Description
Mission phase	A portion of a mission that has a specific objective or set of objectives and has defined initiation and completion events. Mission phases will vary according to the nature of the mission area, but there is likely to be some commonality of phases between similar end items. Typical mission phases for a launch vehicle are pre-launch initialization, first stage, second stage, payload deployment, collision avoidance, and reentry. Typical mission phases for a space vehicle are pre-launch initialization, ascent, separation and initialization, orbit transfer, spacecraft and payload checkout, nominal operations, and disposal. Mission phases for a command and control system might include test, training, rehearsal, activation, shadow operations, initial operations, contingency (backup operations), maintenance, and sustainment. There are likely to be additional mission phases for more complex missions involving constellations, multiple agencies, and multiple users.
Mission readiness test (MRT)	An operationally realistic test that demonstrates the *entire* system's ability to conduct its mission by accounting for all elements and aspects of the operational chain. It involves activities conducted according to a mission-equivalent timeline, using as many elements of the operational chain as is feasible. It includes information and data flows, as well as transactions between interfaces in the operational chain. It is conducted over a period of time (i.e., days in the life [DITL] or even weeks in the life [WITL]) to cover critical events and transitions, as necessary.
Mission scenario	An operational sequence within a given mission phase. The sequence consists of an ordered list of functions and activities required to achieve a distinct mission objective.
Mission thread	A time-sequenced slice of a mission scenario in which the system explicitly supports actions to be executed by the user. Each mission thread is basically a story that comprises the events, actions, stimuli, information, and interactions applicable to utilizing the deployed system within the context of its place in the overall enterprise architecture. Mission threads include descriptions of the system's role in sample user missions and information regarding interactions with the user, interfaces to other systems, and any pertinent system states or modes. Mission threads also provide a framework for the system threads, mapping specific system threads to mission thread steps. From a ground-user perspective, mission threads may be likened to "work flows"—strings of activities required by operators to execute a specific ground system capability (e.g., collect a target, calibrate the system).
Mission timeline	A time-sequenced list of events of a satellite beginning at launch and extending to end of life. A timeline is at the highest level in the mission pyramid (see Figure 14).
Operational timeline sequence	Events per mission phase and the intended order for the events to occur. The operational timeline sequence is a time-sequenced list of events of a system (space/ground) beginning at launch/initialization and extending to end of life.
Operationally realistic test	A test that demonstrates the system's ability to conduct its mission. It may be allocated to any level in the system-integration, mission-level, and supplier-level pyramids. It represents the mission in terms of phase, transitions, environments, and events in an end-to-end system configuration context (i.e., combination of HW/SW and data when functioning as an integrated system); acceptance of mission inputs; execution of mission functions; and production of mission outputs according to the typical operational rhythm, timelines, and sequences resulting in end-user goals (products, services, and timeliness).
Part	A single piece, or two or more joined pieces, which are not normally subject to disassembly without destruction or impairment of the design use. The lowest level of separately identifiable items (e.g., piece parts, resistosr, capacitors, inductors, application-specific integrated circuits, etc.) [6].

Term	Description
Segment	A logical and integrated group of similar functions provided by a combination of people, HW, SW, and data. Each segment is composed of both internal and external interfaces—the former is where segment elements are joined together and the latter where segments are joined as part of a more complex integration [18].
Software unit	An element in the design of a software item, such as a major subdivision of a software item, component of that subdivision, class, object, module, function, routine, or database. Software units in design may or may not have a one-to-one relationship with the code and data entities (routines, procedures, databases, data files, etc.) that implement them or with the computer files containing those entities. A software unit was sometimes previously called a computer software unit [2].
Subassembly	An entity containing two or more parts that is capable of disassembly or part replacement. An integrated set of components and/or parts that comprise a defined portion of an assembly.
Subsystem—hardware	An assembly of two or more components, including the supporting structure to which they are mounted and any interconnecting cables or tubing. A subsystem is composed of functionally related components that perform one or more prescribed functions [6].
Subsystem—software	Synonymous with, or used instead of, software item in some programs. The specification level consisting of software requirements [6].
System	A composite of equipment, skills, and techniques capable of performing and/or supporting an operational role. A complete system includes all equipment, related facilities, material, software, services, and personnel required for its operation and support to the degree that it can be considered self-sufficient in its intended operational environment.
System thread	A description of the activities that must be executed by the elements of the system and its users to provide particular system functionality or an end-to-end user service/product.
Systems of systems/[enterprise]	A set or arrangement of related interdependent systems that provide a given capability. The loss of any part of the system will significantly degrade the performance or capabilities of the whole [7].
System of systems	A collection of task-oriented or dedicated systems that pool their resources and capabilities together to create a new, more complex system which offers more functionality and performance than simply the sum of the constituent systems.
Unit	A functional item that is viewed as a complete and separate entity for purposes of manufacturing, maintenance, or record keeping. (See definition of “subassembly.”)
Use case—software	A description of the major functions the system will perform. It can include descriptions of flows or scenarios, and may also include a collection of scenarios that together accomplish a specific goal. Each scenario corresponds to a single path or flow through a use case.

3. The TLYF Process Overview

TLYF has historically meant different things to different people and organizations. This guide promotes a specific meaning for the phrase. It is understood that other definitions may exist, but for the purpose of describing what is involved in this guide, an "evolved" definition is provided:

> TLYF is a *prelaunch/pre-operational* systems engineering process that translates mission operations concepts into perceptive operationally realistic tests to detect latent mission-critical flaws. It assesses the risk of missing those flaws when it is either not feasible to do those tests or adequately represent key mission characteristics while executing such a test.

It is key that the formulated set of "like you fly" or "like you operate" tests is conducted prior to flight or fielding in order to provide an opportunity to find mission execution flaws in the actual system before the system is launched or goes operational. It is important to include or account for as many critical operational characteristics as practical in an end-to-end fashion. This approach leverages mission concepts to define perceptive tests that can address potential fault paths and contributors.

3.1 The Need for a TLYF Process

Unlike typical tests, there is a duality for those formulated as a result of the TLYF process, as shown in Figure 1. On one hand, it helps validate mission timelines, operations, and conditions as much as is practical. On the other hand, it promotes finding or uncovering particular classes of flaws that only manifest themselves in particular mission usage cases.

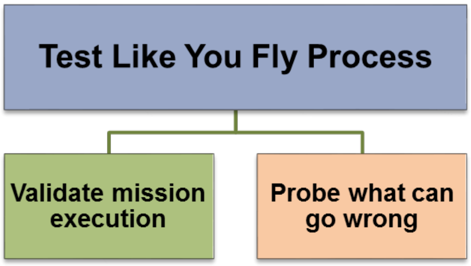


Figure 1. TLYF duality.

The TLYF process consists of sound overall systems engineering practices that inform test development to ensure that acquired systems can accomplish their intended missions. It does this by focusing on demonstrating the system's ability to execute critical mission activities prior to mission activation/launch. It promotes operationally realistic testing that incorporates critical mission characteristics (i.e., operational configurations, environments, procedures, and people). These tests are considered "mission execution validation tests"[2] or operationally realistic LYF tests because they are derived from the concept of operations (CONOPS) and/or related mission operations requirements documents. These tests are *mission operability-centric* instead of *requirements-centric*. Understanding the difference between the two can be accomplished by the types of questions that are asked:

[2] Confirmation through the provision of objective evidence (pre-flight system testing) that the requirements for a specific intended use or application have been fulfilled.

- **Mission operability-centric question**: Can the system (space and ground/launch vehicle (LV)/ground only, products, and personnel) accomplish the mission as envisioned?

- **Requirements-centric question**: Does the system meet defined requirements?

These are important and entirely different questions. History shows a number of missions that met all requirements on the ground but failed early in the mission.

The TLYF process described herein is derived from specific lessons learned that have not necessarily been captured in existing test specifications and standards. Tests meant to verify design and performance are distinct from those designed as a result of following the TLYF process.

The following tests are not necessarily considered "like you fly":

- Tests whose objectives include a demonstration of performance margin

- Tests whose approach is determined by qualification/protoqualification/acceptance levels and related guidance

- Tests whose verification relevance is assessed by merely evaluating form/fit/function

Tests created to demonstrate or verify individual requirement compliance may not be sufficient for complex payloads, spacecraft, systems, and missions. Necessary does not always equal sufficient. One critical lesson derived from past mission failures is that they occurred in spite of reasonably rigorous adherence to requirements verification testing. The root causes for these failures are not related solely to environmental conditions; they are related to the way in which the mission is really executed, which was never demonstrated/validated in test.

While environments are included in mission characteristics, an environmental test—as traditionally executed—is not considered a LYF test unless it has been formed with the mission in mind. Demonstrating that HW survives an environment, although a necessary prerequisite, is not the same thing as showing that the integrated HW, SW, processes, and procedures actually work together. It also does not show that HW performs mission activities adequately while experiencing the concurrent mission characteristics (power levels, signal protocols, etc.) in the combined radiation, thermal, and electromagnetic interference (EMI) environments. A combined effects test that includes other mission characteristics (e.g., duty cycle, power level) may be more LYF than serial environment testing.

Top-level requirements are usually oriented to mission-specific performance characteristics, e.g., resolution, antenna gain, images per pass, bit error rate (BER), and data latency. These requirements are generally derived from end-user needs without reference to how they are influenced by regular mission operations. Tests are created to demonstrate individual requirement compliance; however, it may not be true that the requirements can be met in the context of mission operations where time, max loading, order, and transitions, as well as environmental interactions, may affect the ultimate product or service.

In addition, failure in properly identifying, decomposing, and communicating requirements can be sources of error, especially where those requirements do not adequately account for the operational environments and other conditions associated with flight. The fundamental requirement, *it shall work*, is usually not included in requirements specifications. This observation stems primarily from the fact that national security space (NSS) projects generally do not write operability-type requirements. *If a program makes a specific effort to capture critical operations with operability requirements, then tests identified as a verification method may result in being more operationally realistic.*

Many test programs are rightfully focused on using testing to verify requirements. However, another very important purpose of testing is to uncover existing flaws in the system before it is launched or goes operational. When verification is the only goal of a test, flaws found during the test are frequently seen as “bumps in the road,” slowing down the test from its “true” and noble purpose. When the “bumps in the road” are removed, the results can give a false sense that there are no flaws in the system. Systems always have flaws. Ignorance—anywhere in the system—is not bliss. Testing brings value when it uncovers latent mission-affecting flaws that have escaped from previous processes.

The tests resulting from the TLYF process address the mission operability-centric question by accounting for all relevant mission characteristics when building those tests. Mission characteristics may include, but are not limited to, the environment (internal, ascent, space), configuration(s), timing, sequence, transitions, and other applicable mission attributes (see section 4.2.2 for examples). TLYF provides a method to assess mission characteristics for testability and to assess the risk for those concepts that are not readily testable. This is the basis for two fundamental steps, explained later in this document (see sections 4.4 and 4.8).

3.2 Distinguishing Operationally Realistic LYF Tests

LYF testing features need to be distinguished from other type of testing for clarification of its design and purpose.

3.2.1 Environmental and Qualification Testing

There are a series of tests that may be considered to be operationally realistic LYF tests because they include space environment characteristics. However, it is important to understand each test’s focus and be aware of what is not like the mission. With a closer look, these tests usually do not address critical mission characteristics.

The qualification strategy establishes a series of tests intended to verify HW design to qualification or protoqualification levels, followed by acceptance testing of subsequent flight HW to find workmanship defects. Many aspects of spacecraft acoustics and thermal vacuum testing differ significantly from the actual launch or mission conditions. Examples include:

- Spectrum duration in acoustics testing is 60 seconds. Total ascent duration can be as long as 60 minutes, depending on launch vehicle and target orbit.

- The acoustic spectrum in acoustics testing is constant. Real-flight sound pressure level varies with time and is interspersed with shock events at separation events.

- The acoustics chamber creates the same sound pressure level in all directions. Due to launch site architecture, real launch has asymmetrical sound pressure levels until the vehicle is clear of the launch site.

- Vibration induced by the launch vehicle is not included in acoustics testing.

- All units that are not susceptible to damage by being powered during acoustics testing are powered. They are sequenced through operational modes and monitored to detect intermittent failures. In real flight, units operate concurrently and run the ascent profile.

- The total number of thermal cycles in specification-driven thermal vacuum testing is four to eight because it is used for proof of design or workmanship only. Depending on orbit and design life, real spacecraft experience hundreds or thousands of thermal cycles.

- Thermal dwell time and rate of temperature change in thermal vacuum testing are driven by the time needed for functional and performance testing and the characteristics of the chamber. The soak time at hot or cold temperatures for real spacecraft is orbit dependent. Low-Earth-orbiting spacecraft experience thermal transitions in seconds.

- Thermal radiation view angles are compromised in thermal vacuum testing by blockages to chamber walls.

- The ascent pressure profile is not simulated in thermal vacuum testing. It may take eight hours to pump down the chamber, whereas static pressure decrease takes minutes in real flight.

- Deployables remain stowed during thermal vacuum testing because of chamber size limitations or 1-g effects.

- Some types of heat pipes may not operate during thermal vacuum testing because they require a zero-g environment.

- Functional and performance testing are typically done subsystem by subsystem during thermal vacuum testing. Subsystems operate concurrently on orbit.

3.2.2 Other System Tests

In addition to environmental tests, there are others (i.e., compatibility, interface, functional, and performance) that are valuable and perceptive for uncovering specific flaws; however, they are generally not considered LYF unless they are specifically designed to be operationally realistic. The worst example of this would be a program where function and performance test scripts were ordered and executed in alphabetical order. Emphasis on individual performance requirements does not offer insight into the challenges of performing the actual mission. **The goal is not necessarily to remove these tests and replace them with operationally realistic LYF tests. The goal is to be aware of the potential gaps and mission flaws that are escaping due to traditional system testing.**

In addition, most engineering tests are driven by the test design principle of varying a single independent variable to isolate dependencies on that variable. Much of traditional testing is based on test requirements standards (i.e., SMC-S-016) that follow this principle. No matter how well that serves the engineering test perspective, the principle fails to create a complete understanding of what happens when actually executing the mission, where many characteristics vary and interact as a function of time and sequence. Time and sequence are vital in successfully executing a mission.

3.2.3 The Value of End-to-End Testing

A good deal of design work centers on interfaces between items. The underlying philosophy is that if the interface is carefully and thoroughly specified, and each interface is carefully designed, built, and verified to meet the specification, the system should work after the items are integrated. The hard reality is that space systems are presently so complex that it is virtually impossible to completely specify an interface across enough combined conditions to ensure adequate operational margins for all relevant parameters [22].

End-to-end operationally realistic tests provide the opportunity to expose anomalies not discernable in other test before launch or system fielding. The resulting test design includes system end-to-end aspects and crossing interface boundaries while executing transfer of information activities. It incorporates other operational attributes (i.e., mission threads, scenarios, and mission phases) from a representative mission timeline.

The knowledge concerning what the mission is and how it is to be flown/executed is typically documented in the CONOPS at a very high level. Unfortunately, critical mission details (at lower levels) are not captured in any system-level requirements verification matrix nor formally documented. Further, detailed mission CONOPS for ground systems may not be defined until late in the development cycle (i.e., critical design review [CDR] time frame). Both of these factors can result in many lower-level operational concepts never being scrutinized or formally validated.

The U.S. Air Force (USAF) conducted LYF end-to-end tests for a program late in the acquisition cycle which resulted in the discovery of anomalies not otherwise detected. These tests proved to be highly perceptive to both SW and HW anomalies, effectively uncovering numerous mission-degrading/mission-critical SW defects which escaped all earlier testing. Specifically, a payload ("end-to-end") mission demonstration uncovered many critical/mission-degrading flight software defects. The mission aborted at the 68th hour of a 72-hour demonstration. Problems were traced to flight SW defects which had escaped all earlier testing, including software qualification testing. Without the test, it is likely these problems would have been discovered on orbit as mission-critical. Fortunately, all issues were resolved prior to launch [24].

Table 1 shows the results from an examination of the outcomes of 50 end-to-end tests. These tests were performed on 23 space vehicles (SVs) for up to weeks in duration. Each test involved the spacecraft in its assembly, integration, and test (AI&T) location being operated from the ground station. These tests detected between 3 and 9 mission-degrading anomalies (MDAs) per test, with an average of 3.6 anomalies per test [10]. An MDA in this study was defined as a flaw that causes the system to abridge its primary mission. These data show that this type of test has a comparable effectiveness rating for finding flaws to that of a SV thermal vacuum test, which has the highest effectiveness (four to six MDAs) of any environmental test [11]. It must be emphasized that these end-to-end, WITL tests are run after requirements verification activities—including interface checks and compatibility tests—have been completed.

Table 1. End-to-End Test Anomaly Detection Summary

Buyer	Prime Contractor	Space Vehicle	Number of Space Vehicles	Number of End-to-End Tests	MDAs Detected/Test
NASA	JHU/APL	MESSENGER	1	5	6.2
NASA	JHU/APL	New Horizons	1	4	9
ESA	Various	Various	20	40	2.6
USAF	Boeing	ARGOS	1	1	3

Experience shows that MDA escapes are reduced by requiring operationally realistic LYF tests at the highest level of integration (end-to-end configuration) to validate mission readiness prior to launch.

Operationally realistic lower-level tests are executed as needed for practicality, perceptivity,[3] resource optimization, or risk reduction.

The TLYF process can help generate operationally realistic tests that:

- Show the ability to properly transition from one activity to the next
- Reveal that the system can sustain continuous and duty-cycle–driven activities
- Demonstrate proper timing interactions between asynchronous activities
- Allow for error growth and discovery in SW execution
- Simultaneously support necessary "threads"

3.2.4 TLYF Process Builds Tests to Look for Potential Mission-Critical Flaws

Tests can be designed to look for flaws that obstruct mission objectives. The TLYF process promotes fault-informed testing by incorporating a mechanism (see section 4.3) to identify potential mission-ending fault paths. Tests are designed to address (and hopefully exonerate) the potential faults. Finding mission-critical flaws may require multiple types of testing. Operationally realistic tests that represent critical operations and conditions increase perceptiveness to discovering flaws in the system.

3.3 TLYF Process Key Elements and Flow

The TLYF process provides a comprehensive approach throughout the acquisition cycle to validate a system's capability to perform its mission prior to launch or fielding. It consists of three key elements:

1. **TLYF planning**, which pertains to the aspects of the TLYF process to be incorporated when creating and evaluating request for proposals (RFPs). It is a proactive way of promoting mission success by contractually identifying specific TLYF tasks and products

2. **TLYF process implementation—**the focus of this guide—which is the seven-step method that enables the formulation of operationally realistic LYF tests and the discovery of potential risk to mission success via systems engineering and test development activities

3. **TLYF assessment**, which is the method to help determine potential gaps in a program's use of recommended TLYF-related systems engineering and test development activities that may introduce mission execution risk to the system under review

Pre-Phase A programs begin with TLYF planning. The result of planning is a set of recommendations for implementing the TLYF process into the contract. The recommendations are translated into specific contractor tasks and related products, and the details are included in the RFP package where appropriate. For programs that are beyond Phase A, the entry point for applying TLYF is the assessment. The findings from the assessment may result in updates to the program [17]. A program may implement the TLYF process either as a result of TLYF planning or a TLYF assessment.

[3] Note: It's not always the goal to perform tests at the highest level. A lower-level test conducted in an operationally realistic manner may be more practical and perceptive than one performed at the system level (e.g., sensor testing).

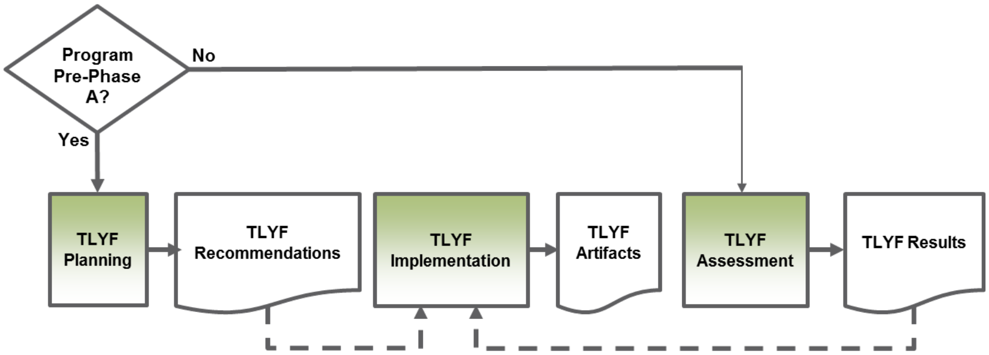


Figure 2. TLYF process application.

3.3.1 TLYF Planning

TLYF planning promotes mission success by placing hooks in contracts based on the steps in the TLYF process. These steps are translated into specific contractor tasks and associated products. TLYF planning reduces critical fault risk by incorporating the TLYF process upfront and should be considered when supporting the creation and preparation of program documentation and RFP packages. Ideally, TLYF planning begins early in the acquisition lifecycle (i.e., DODI 5000.02, Pre-Phase A [20]), since much of the planning activities pertain to the acquirer/buyer.

3.3.2 TLYF Assessment

The TLYF assessment process leverages the Aerospace-defined TLYF process and has the following objectives:

- Evaluate a program's systems engineering and test development work products and processes using the TLYF process as a baseline
- Identify risks in current approach
- Recommend improvements for those products and/or processes as they pertain to operationally realistic LYF testing

These objectives are accomplished specifically by:

- Identifying compliance (pluses) and gaps in a program's use of the TLYF process (systems engineering and test processes/practices) to accomplish preflight/pre-operational mission validation
- *Identifying any disconnects between mission-critical fault analysis (MCFA) results and test program that may introduce mission risk (i.e., mission-critical events that are not represented in test and/or mission-critical flaw paths/flaw contributors that are directly tied to LYF test exceptions)*
- Providing impacts and mitigation recommendations to address highlighted gaps
- Summarizing the risks to mission success that remain in the program due to not addressing identified gaps

There are trades to consider when scheduling an assessment. The timing will influence the opportunity for modifying the test program. If it is conducted too early in the lifecycle, the system will lack details for designing an operationally realistic test. If it is conducted late in the schedule, changes to the test program may be difficult and costly. More detail is provided in section 5.

3.3.3 TLYF Process Implementation

Implementing TLYF into a program is the centerpiece of the TLYF process and is the focus of this guide. The following sections will describe the key steps, considerations, and products for successful TLYF process implementation.

4. The TLYF Process—Implementation

The TLYF process results in operationally realistic LYF tests that address potential mission-critical flaw paths and contributors. The steps herein will describe how to get there. It is necessary to reach beyond the test domain to effectively implement this process and develop meaningful operationally realistic tests.

The TLYF process comprises test development (operationally realistic test development) and systems engineering activities (mission fault-informed risk management), as shown in Figure 3. These activities inform each other throughout the process. The systems engineering activities include characterizing the system and mission, MCFA, mapping the mission to LYF tests (i.e., identifying, assessing, and allocating candidate LYF tests), and performing critical fault risk management. Test development activities include architecting, designing, executing and evaluating LYF tests. Test development relies on the outputs from systems engineering activities. These activities ensure that the resulting LYF tests include essential mission characteristics and are perceptive to faults that can cause loss or severe degradation of the mission.

The tests that can be planned and executed to emulate mission operations will necessarily deviate from actual mission operations in a number of ways. Understanding the opportunities to surface mission-critical flaws when architecting and designing LYF tests is crucial. These missed opportunities are risks and must be managed to ensure that program resources are adequately deployed to address them. How it will be managed can vary from program to program. Ultimately, it will require an organization, function, individual, or team that has purview of the complete picture—the mission, the system, systems engineering, integration, and testing.

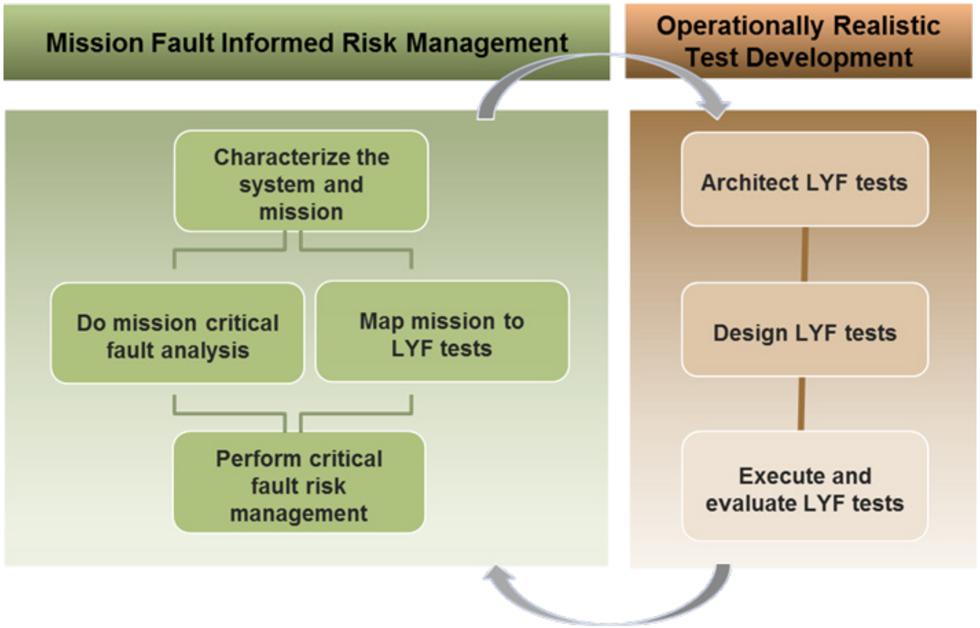

Mission Fault Informed Risk Management
Operationally Realistic Test Development
Characterize the system and mission
Do mission critical fault analysis
Map mission to LYF tests
Perform critical fault risk management
Architect LYF tests
Design LYF tests
Execute and evaluate LYF tests

Figure 3. TLYF process implementation.

A summary of the main components of the TLYF process implementation is provided in Table 2. Each step will be explained in more detail.

Table 2. TLYF Process Implementation Steps Overview

Step	General Description
Characterize the System and Mission	TLYF requires knowledge of the system and its mission. • System aspects: The starting point for a TLYF is understanding the "what" and "how" of the mission (i.e., what elements are involved and how they interface and interact, internally and externally). It is important to understand the basic system elements, structure, architecture, interfaces, and design. How should it work? • Mission aspects: The mission aspects reveal the "how" and "when" of system operation, what defines mission success, and the details of its resulting services or products. • The concept of mission-critical activities becomes a foundation for the later subject of fault analysis of mission-critical situations.
Do MCFA	MCFA results provide a mechanism for identifying mission risk unlike other failure analyses. It uses the system's operational timeline to identify mission-critical events and transitions. It starts from the top of the mission-level and system-integration pyramids[4] and works down to subsystem, component, actions, discrete events, and conditions, as necessary. The MCFA determines the flaws that could prevent success for each of these critical events. If such a flaw exists, it must either be exonerated, discovered by some means and mitigated accordingly, or identified as a risk to mission success. There is risk associated with the characteristics that are deemed critical to the mission that are not tested (i.e., it is an LYF test exception). If the obvious test cannot be performed because it is impractical or impossible, the risk of failing to detect mission-critical flaws increases. When an operationally realistic LYF test decreases use of mission aspects (articles, conditions) with non-flight/non-operational aspects, the likelihood of missing critical flaws increases.
Map Mission to LYF Tests	Candidate LYF tests are based on the mission operational concepts, events, phases, and timelines. Candidate LYF test objectives are tied closely to mission objectives. There is special emphasis on first-time and mission-critical events. Tests may also be derived from the MCFA to surface flaws associated with lower levels of integration or particular mission scenarios. Candidate LYF tests are identified, assessed, and ultimately allocated as result from this step.
Architect LYF Tests	Allocated LYF tests must be architected. This will involve trade-offs among test objectives, mission/operational characteristics incorporated, and the fidelity to the way the activities will be executed during the mission. The architect must be responsible for resource management and initial identification of LYF test exceptions. The architect is responsible for planning, designing, and overseeing the construction of LYF tests.
Design LYF Tests	LYF test design involves selecting operationally realistic test cases that clearly meet the following objectives: (1) making sure test objectives are clearly tied to mission objectives, (2) selecting mission characteristics that need to be included and over what range, (3) deciding on initial, stressing, and end conditions, and (4) determining what test equipment, simulators, and simulations will be necessary to execute the test. The test designer is responsible for identifying and evaluating specific LYF test exceptions.
Execute and Evaluate LYF Tests	Execution and evaluation of LYF tests must provide evidence for operability and exoneration of paths to mission failure identified during the MCFA step. Tests that uncover flaws will provide the basis for additional tests (regression or new). Evaluation of LYF tests includes both the degree to which the test results meet the success criteria and the necessity to rerun the test following changes to hardware, software, processes, or procedures.
Perform Critical Fault Risk Management	Critical fault risk management encompasses identification, analysis, mitigation planning and implementation, monitoring, and elevation of critical fault (CF)-related risks. The mission-critical risks identified are based on (1) the potential flaw paths to mission-critical failure situations, as an output of the MCFA; and (2) the LYF test exceptions identified during the design LYF tests step. Critical fault risk management accounts for the risk of not being able to test in a flight like manner by evaluating LYF test exceptions. High-risk items identified by the MCFA (i.e., potential contributors to mission failure) not addressed in testing must be added to a program's risk management process.

The information flow for the TLYF process does not necessarily follow a sequential path. Some steps will be repeated as the program progresses and some will be done concurrently, as shown in Figure 4. For

[4] See definitions section

example, it is possible to finish *Characterizing the System and Mission* and have those outputs feed into *Do Mission-Critical Fault Analysis* while also providing the initial information for *Mapping Mission to LYF Tests* and *Architecting LYF Tests*. There is a critical feedback loop from test results that informs the *Performing Critical Fault Risk Management* step. Most steps will be repeated as a program progresses through the various development stages.

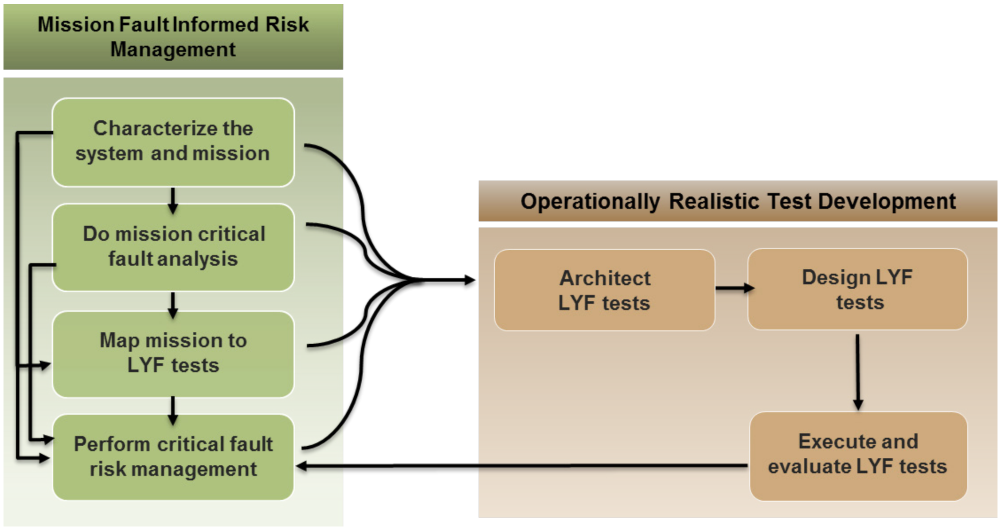

Mission Fault Informed Risk Management
Characterize the system and mission
Do mission critical fault analysis
Map mission to LYF tests
Perform critical fault risk management
Operationally Realistic Test Development
Architect LYF tests
Design LYF tests
Execute and evaluate LYF tests

Figure 4. TLYF data flow.

4.1 TLYF Process Inputs and Outputs Summary

Each step has inputs, outputs, and key methodologies. The type of data that flows from each of the steps in the process is summarized in Table 3. It is not an exhaustive list but intended to provide guidance on the type of information needed. How the sources are used to create the TLYF-related products are discussed in each section.

Table 3. LYF Process Implementation—Inputs and Outputs Summary

Step	Sources (Inputs)	Products (Outputs)
Characterize the System and Mission	*System Aspects* • Architecture documents (e.g., Department of Defense Architecture Framework [DODAF]) • Supplier identification and product(s) • System details (e.g., design, interface requirements, system requirements) • System requirements (e.g., technical requirements document [TRD]) *Mission Aspects* • Mission operations-related work products (e.g., On-Orbit Handbook [OOH], concept of operations [CONOPS], operational constraints) • User operational requirements (e.g., operational requirements document [ORD], capability development document [CDD], functional requirements document [FRD]) • Mission details (e.g., design reference, timeline, phases, threads, products, services) • Any internal work products related to the mission/ products/services (e.g., use cases) • Threat assessments	Timeline of each mission phase and sequence of events List of mission-critical events/activities (nominal, first-time, critical, critical recurring, and contingency) List of mission characteristics for each mission-critical event/activity Associated objective for each mission-critical event/activity or set of events/activities Mission-critical events/activities grouped by mission phase (operational sequence)
Do MCFA	*Characterize the System and Mission* – step (inputs and outputs) Failure mode and effects analysis (FMEA)/failure mode and effects criticality analysis (FMECA) Fault management schema and contingency plans Reliability work products (e.g., single point failures, critical items list) Mission design and representative operational timelines for each phase of the mission (for ground system operations this may involve multiple activities operating in parallel to the mission)	List of mission-ending failure situations (including first-time and mission-critical events) covering all mission phases (fish head/tree tops) For each failure situation (includes successful completion of first-time/mission-critical events), the following is generated: • Fish head/tree top signifying the failure situation • Fish bones/branches for potential fault paths and contributors (including parallel contributors and processes) • Exoneration plan for each failure path • Exoneration methods for each failure situation (includes successful completion of first-time/mission-critical events) List of mission-critical events and fault paths not planned to be exonerated (risks)

Step	Sources (Inputs)	Products (Outputs)
Map Mission to LYF Tests (identify, assess, and allocate)	*Characterize the System and Mission* – step (inputs and outputs) • List of first-time events and corresponding mission characteristics • List of mission-critical events and corresponding mission characteristics • Success criteria for each mission-critical event/activity or set of events/activities *Do Mission-Critical Fault Analysis* – step (inputs and outputs) • List of mission-critical failure situations (including failure to execute first-time and mission-critical events), paths and potential contributors to failure Test program work products and information • Test program details (requirements, contractual products and guidance) • Test tools and resources • Test strategy documents (e.g., Test and Evaluation Master Plan [TEMP])	List of allocated LYF tests and their priorities For each allocated LYF test, the following is generated: • Test objective based on corresponding mission objective • Allocated pyramid level (supplier, system integration, and mission) • Allocated resources for test (operational and test resources) • LYF test exceptions list (high level) Allocated LYF test details documented (test plans and contractual documents) List of remaining unallocated LYF candidates tests with brief rationale List of first-time/mission-critical events with no planned validation
Architect LYF Tests	*Characterize the System and Mission* – step (inputs and outputs) *Do Mission-Critical Fault Analysis* – step (inputs and outputs) *Map Mission to LYF Tests* – step (inputs and outputs) *Perform Critical Fault Risk Management* – step (outputs) Supplier identification and responsibilities Program integrated master schedule (IMS) Details for resources available for allocated tests (i.e., fidelity, hardware, software)	Architected LYF tests For each architected LYF test, the following is generated: • Mission phase/event/failure situation coverage • Test plan (high level) • LYF test exceptions New candidate LYF tests (if applicable from *Perform Critical Fault Risk Management* – step)
Design LYF Tests	*Architect LYF Tests* – step (inputs and outputs)	Operationally realistic LYF tests *(each addresses potential mission-critical flaw paths and contributors)* For each LYF test, the following is generated: • Test plans (detailed with configurations and resources) • Test procedures • Entrance and exit criteria • Specific LYF test exceptions
Execute and Evaluate LYF Tests	*Design LYF Tests* – step (inputs and outputs)	For each LYF test executed, the following is generated: • As-run (redlined) test procedures • Test results (report/data) • Discrepancy reports (DRs) • Additional LYF test exceptions (if applicable) • Additional flaws/faults discovered (if applicable) • Retest plans and procedures (if applicable)

Step	Sources (Inputs)	Products (Outputs)
Perform Critical Fault Risk Management	*Execute and Evaluate LYF Tests* – step (outputs) *Do MISSION-CRITICAL FAULT ANALYSIS* – step (outputs) • List of mission-critical events and fault paths not planned to be exonerated via LYF tests *Map Mission to LYF Tests* – step (outputs) • List of LYF test exceptions linked to MCFA branches • List of first-time/mission-critical events with no planned validation	TLYF associated program risks: • List of LYF test exceptions linked to MCFA branches • List of first-time/mission-critical events with no planned validation LYF test exception handling • LYF test exception risk evaluation Risk mitigation plans • Proposed new LYF test candidates (if applicable)

4.2 Characterize the System and Mission

The first step of the process is to understand the system, its mission, core phases, and associated events and activities. Characterizing the system and mission is foundational. How the mission will be successfully executed is key to developing operationally realistic LYF tests. This fact is largely why the entire process starts with LYF rather than test. Because of this dependence on mission operations on LYF testing, the TLYF process has ramifications beyond the confines of a test organization. System and mission details and objectives provide the primary data for informing each LYF test.

Characterizing the system and mission requires four basic items:

1. System information (i.e., design, architecture, functional flows, interfaces)

2. Mission information (i.e., mission success definition, concept of operations, mission phases/events/sequence of execution, design reference mission)

3. Concept of how the system will be operated to support mission objectives—both as its own system and also in the context of the enterprise, if applicable

4. Knowledge of program resources which may interact with the system that is being built (i.e., support systems, legacy/heritage systems, user needs, tasker requirements)

How the mission will be executed must be understood before a test that emulates the conduction of the mission can be crafted. The questions of *who*, *what*, *where*, *when*, *why, and how* of the mission must be answered first.

The system aspects help to understand the "what" and "who" of the mission, i.e., what elements, people, resources and interfaces are involved. The mission aspects reveals the "how," "when," and "why" of system operation. The essence of mission operations is to execute a time-driven set of activities that occur between interacting elements and systems to accomplish the mission. It defines mission success and the details of its resulting services and/or products. Thus, the system and mission are tightly linked in the formulation of operationally realistic LYF tests.

This step in the process can begin as early as the pre-acquisition phase, after an initial system design and accompanying mission objectives and operations concepts have been defined. The acquisition team should make decisions that determine the extent to which the concepts are testable, the degree to which they are willing to include validation and risk-reduction testing of those concepts, and what items must be acquired or provided to enable those tests. For some acquisitions, the CONOPS will involve legacy support systems as well as tasking and user organizations. Those acquisitions will need to determine the systems of systems extent of the LYF tests.

4.2.1 Key Questions

The easiest way to extract the necessary mission information is to address the following questions:

- Why are we doing the mission? (*Why we fly/why we operate*)
- What is involved in executing the mission? (*What we fly/what we operate*)
- How is the mission accomplished? (*How we fly/how we operate*)

Why We Fly?

"Why we fly" must be considered when building LYF tests. This will allow priorities to be set and tests adequately designed in the context of mission success. The type of mission, such as product or service, will help answer this very important question. It is essential to know the primary mission contributors (e.g., photons in or messages in) and the expected mission results and products (e.g., data out or messages out).

Most unmanned space system missions can be put into one of two categories: product missions or service missions. Product missions use sensors to observe and collect data on a phenomenon (photons in) and/or return raw and/or processed data to the ground. The ground then performs additional processing of the data to turn it into products which are distributed to users. There are two primary service missions: communication and navigation. Communication missions receive owner- or user-generated signals and distribute them to subscriber/other user equipment. A navigation mission provides time and position data directly to user equipment. Most research/development (R&D) projects will also fall into these broad categories.

For each type of mission to be successful, a number of events must first be successfully executed. A typical product mission would include something like these events:

1. Looking at the correct phenomenon in the correct location at the correct time

2. Turning the collected information into raw and/or processed output data

3. Transmitting mission and health data to spacecraft data handling and telemetry subsystems for transmission to a ground system

4. Processing the received data into products and distributing those products to users

A typical communications service mission would include something like these events:

1. A ground-based satellite transmitter dish beams a signal to the satellite's receiving dish

2. The satellite processes the data and configures according to what is received from the ground

3. The satellite boosts the signal and sends it back down to Earth from its transmitter dish to a receiving dish somewhere else on Earth [4]

These events can be broken down into smaller pieces with more detailed steps. Each step contributes to the success of the higher-level event. By simply asking “why we fly,” we can assign criticality levels to each of the pieces that contribute to achieving a successful mission.

What We Fly?

“What we fly” and what we use to operate the mission form the basis for what must be tested. The history of spaceflight is littered with failed missions due to a lack of operability tests performed on the *actual* flight and ground systems. Tests done on engineering units and simulators may be useful, but they cannot reveal flaws that are in the flight/ground system or that can manifest during the mission. Tests done on one unit of a series will not reveal flaws on a similar, but not identical, unit. The phrase “similar in form, fit, and function” does not necessarily mean the same as flight.

Depending on the criticality of the item or interface, it may be important to characterize the deviations. Tests of the space element using test equipment for commanding and telemetry may not be an adequate substitute for testing the space element with the operational ground control element. Tests of the ground element face similar issues when tested with SV simulators. This is particularly true for more complex ground systems that dynamically respond to SV telemetry. The omission of tests that involve the total end-to-end operational chain will likely miss the flaws that only occur in the flight configuration with the actual operational elements [24]. In order to successfully perform LYF tests, operational support resources must be available. Historically, operations development (ground system availability, personnel training, operations procedure development) has lagged behind SV development. The program must track the development of those resources to ensure that they are available for LYF testing.

The TLYF process makes use of testing at lower integration levels where appropriate, for perceptivity and risk reduction. This allows testing of operational elements that is not feasible at higher levels (e.g., sensing payload subsystem validation). However, LYF testing culminates in using the final system in an end-to-end, mission execution fashion. This includes the complete mission chain—space segment, ground elements, operational HW and SW, associated databases, processes, procedures and personnel—exercising the entire system from operational planning processes through dissemination of data. For some missions, this will involve many elements, as shown for a notional operational space system in Figure 5.

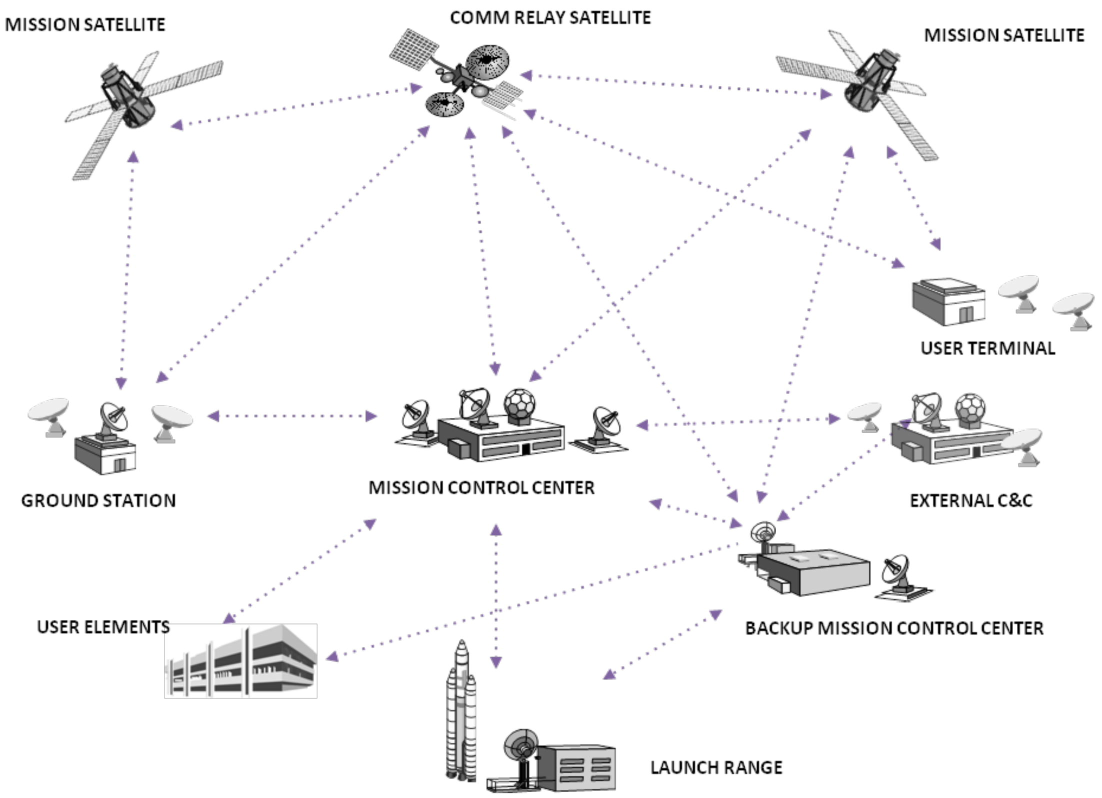


Figure 5. Notional operational space system.

Tests that reveal flaws generally lead to rework. The reworked system is a different entity than the original version. Technical considerations in deciding whether and how to retest include the extent of the most recent change and the kind of tests remaining. Late rework tends to have less rigorous review and control of procedures. Ad hoc rework is a frequent source of additional problems. Inadequate or no post-rework testing of the repaired item is considered a "test-what-you-fly" violation.

Also, it is important to be consciously aware of what is not a part of the operational system. Although a prime tenet of TLYF is to test *what you fly*, it is clear that this may not always be possible. Test-related resources (i.e., SV simulators, engineering units, testbeds, factory test HW, test SW and databases, test scripts, test procedures, and test personnel) are all necessary in SV development, but their use may hide flaws that only come from the integrated system. These items are not cost-neutral even if provided by the contractor. Their use may introduce flaws that have no relation to mission items. In addition, shared test resources (i.e., resources that are shared across several space systems) need to be carefully managed and characterized. Use of nonflight articles should be closely scrutinized and accounted for in evaluating test resources and their availability. (See section 4.6.1.5.)

Lesson: Test What You Fly

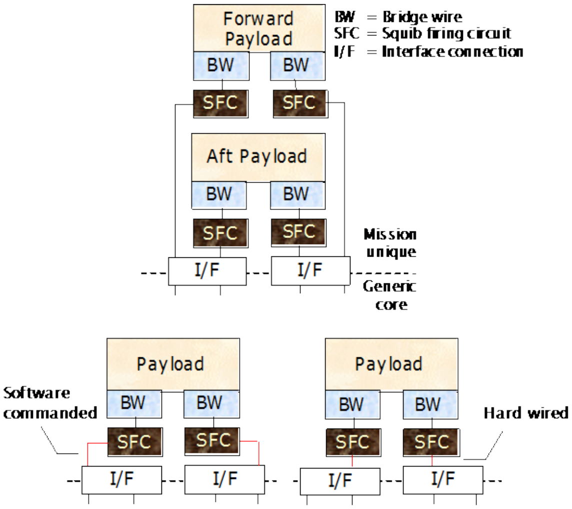
A payload fairing was designed to accommodate two satellites, but only one satellite flew on a particular mission.

The mission specification had the separation commands sent to the “forward” position. An engineer redlined the commands to “aft” to simplify wiring. Somehow this change was not incorporated in the final mission specification.

Not realizing that the informal redline had fallen through the cracks, the hardware group designed an incompatible harness. The drawings were released as a new baseline, making it difficult to detect crucial changes.

Systems engineering departments neglected to check the final design because the key mission specification was developed by software engineers and was not placed under systems engineering’s jurisdiction.

The mistake was not discovered on the ground because the generic systems test activated both positions, allowing the miswired ordnance verification unit to appear working. The satellite was stuck in the nosecone.



Separation Configuration

(Top) For two payloads
(Bottom) For the failed mission

<u>Lessons Learned</u>:

- Test the specific configuration that will be flown.
- Conduct tests and reviews to validate that the requirements are met, rather than that the drawings are correctly implemented.
- Actively involve systems engineers in software development activities, and formally control all system (including software) interfaces.

How We Fly?

“How we fly” addresses the mission execution details. The CONOPS begins the document chain by describing key aspects of the mission, operational context, system drivers and constraints, operational scenarios, and risks. This information is crucial because the manner of use of the spacecraft, payload, and ground system obviously drives the design of both HW and SW. The people, equipment, SW, HW, and facilities used to accomplish the mission should also be identified. This information may be formulated later in the system acquisition lifecycle development. Also, interactions between space and ground control, space and user equipment, tasking agencies to mission planning, ground control to other ground assets, new elements to legacy elements, and/or associated systems must all be described. The environments, both physical and operational, through which the mission must be flown, need to be understood.

4.2.2 Mission Characteristics

After developing a broad understanding of the mission, it is necessary to extract the mission characteristics so that they can be appropriately included or accounted for in LYF tests. Table 4 lists a representative set of mission characteristics, grouped into various classes along with possible test characteristics. The inclusion of the table is not provided to imply that a valid LYF test must incorporate all associated mission characteristics. However, the effects of mission characteristics must be considered for each test and included where appropriate. LYF tests must be designed to ensure that test characteristics do not mask important mission characteristics. The divergence between mission characteristics and test characteristics form the basis for the LYF test exception analysis, which will be discussed later.

Table 4. Mission and Test Characteristics

Mission Characteristic Classes	**Mission Characteristics**	**Test Characteristics**
End-to-end (integration or hierarchy) **level**	Space vehicle, space segment, launch vehicle, ground element, ground segment (user terminal + mission control center + backup mission control center), system (SV + ground control), end-to-end system (space segment + ground segment), systems of systems	Component, unit, subsystem, bus, bus flight software, payload, payload flight software, ground operations equipment, ground operations software
Configuration	SV (with configuration exceptions), space assets, ground SV control (uplink), ground PL control (uplink), ground and interfaces	Non-flight test article, test equipment, test software, facility, simulators, stimulators, test support functions
Time and timeline	Running clock, pre-programmed/invariant command sequence, fixed-duration activity, variable-duration activity, order-dependent activity, order-independent activity, initial conditions, first-time activity, simultaneous activities	Clock reset, test-driven function order, test initial conditions, test duration, test recurrence rate
Environments (internal, ascent, space, ground)	**Space**: Vibroacoustics, shock events, acceleration, booster RF, near-space RF, ascent temperature profile, ascent pressure profile, vacuum, atomic oxygen, eclipse duration, eclipse transitions, solar radiation, cosmic radiation, particle radiation, planet albedo **Ground**: Location (latitude, longitude, elevation), weather (temperature, precipitation, air pressure and cloud cover), climate (long-term weather conditions), system loading	Vibration, acoustic, shock, conducted/radiated EMI/EMC, ambient (room) pressure, ambient (room) temperature Ground simulators, user element models
Uplink (commands)	Bands/rates, contact command plans, time-tagged commanding (CMD uploads), realtime commanding, command options	Command hard line, test command script, test command database, autonomous commanding
Downlink (state of health, mission data, payload data)	Bands/rates, automated (ground) limit telemetry checking, stored telemetry playback, manual/realtime TLM evaluation, TLM trending and evaluation (tools), mission data production tools, databases, data type(s)	Telemetry hardline, STE limit checking, test TLM database, data formats, TLM processing, payload models, simulated payload data

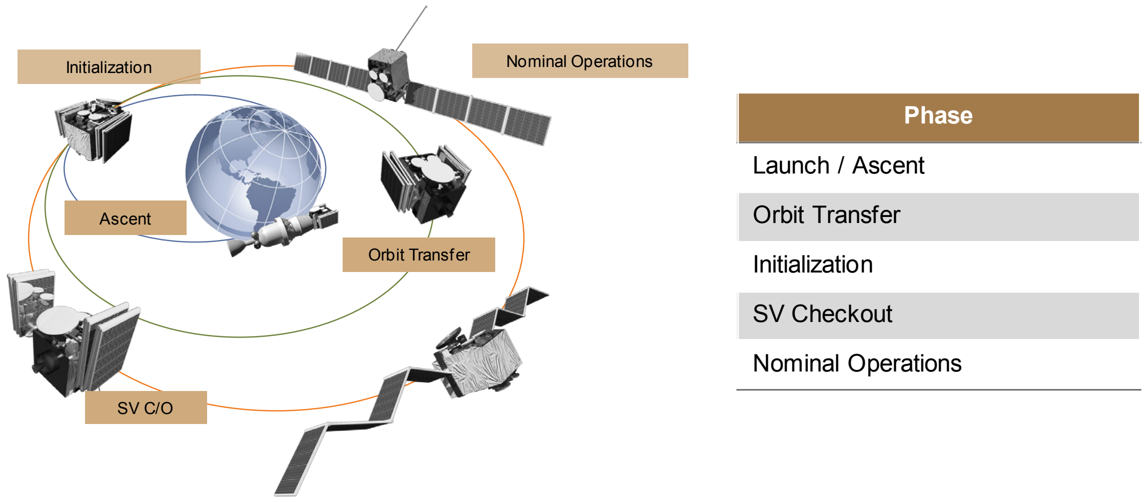
Mission Characteristic Classes	Mission Characteristics	Test Characteristics
Mission planning and operations	Payload planning, mission planning, mission personnel, mission procedures, mission processes, mission phases, operational modes, fault management crew training	Test procedures, test processes, test conductors, training simulators

Mission characteristics may change from phase to phase or during different activities. For instance, the data rates used for state of health (SOH) telemetry may be different during contingency operations than during normal operations. The SV attitude control mode and fault management mode may change depending on the activity or condition.

The time and timeline class are particularly important for LYF tests. Clock resets, time compression, events executed in isolation rather than concurrently, and external timing sources are all test artifacts that can mask serious faults.

4.2.3 Mission Timeline

A mission timeline is necessary for mapping mission phases, events, and modes according to order and timing. A mission phase for SVs may be distinguished by a discrete change in a key characteristic, such as acceleration, configuration, or operation. A mission phase test would include either a complete mission phase or a representative portion of a mission phase, depending on the duration of the phase. The test objective associated with a mission phase test is to demonstrate key mission phase objectives. Figure 6 depicts notional distinct mission phases beginning with ascent and leading to nominal operations.



Phase
Launch / Ascent
Orbit Transfer
Initialization
SV Checkout
Nominal Operations

Figure 6. Notional mission phases.

For ground systems, the phrase "operational mode" may better describe a discrete phase. Ground systems have the capacity/capability to simultaneously execute multiple operational modes. In addition to the mission phases for a SV, ground system modes may consist of the following: launch and early orbit, handover, test/exercise/rehearsal, maintenance, sustainment, transition, and survivable/endurable.

A LYF test based on the mission timeline would include a set of activities performed by the space asset and ground over the course of a few days. This would include:

- Automated spacecraft and payload activities
- Execution of the mission planning activities to produce a schedule and commands to be uploaded to the space asset
- Command uploads as appropriate to space asset contact times
- Insertion of external mission events (remote sensing targets, user interactions)
- Recovery of vehicle state of health and mission data
- Mission product generation and distribution

Figure 7 shows a notional timeline for launch and early operations. Timelines are also extremely helpful in identifying/mapping the unique ground and space segment activities (those being conducted independently and concurrently) required to achieve the mission. The LYF test is a powerful tool in exposing problems that stem from observing the different participants in an operational rhythm coming together. Figure 7 shows four distinct entities for the ground (ground station, mission control center, backup control center, and user) and two for space (payload and bus).

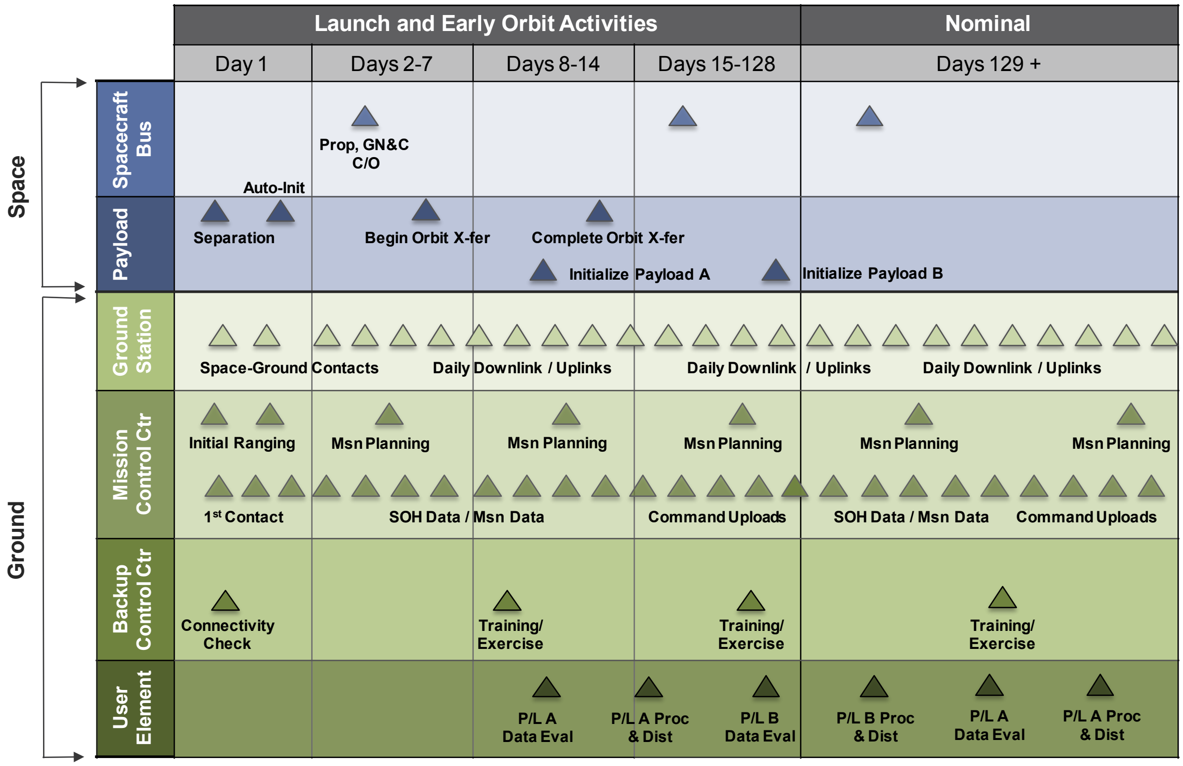


Figure 7. Notional mission timeline.

Each mission phase can be decomposed into a sequence of discrete events. Mission events are noteworthy happenings, whether planned or unplanned. Examples include first light through an optical system, loss of signal (unplanned), transition to safe mode, or transition to operational status. These events can be decomposed into a series of individual mission activities. Examples of mission activities include booster stage separation, collision avoidance, SV contact with ground station, SV entrance into umbra, solar array deployment, acquisition and tracking of a target, uplink of a command load, or broadcast of a timing signal. A naming convention for this decomposition is provided in section 4.3.

When exploring candidate LYF tests, each phase and transition between phases needs to be considered with special attention to those phases and transitions deemed critical. A mission phase should encompass all the events/activities that occur during the phase, as noted in Table 5. If orbit transfer activities and SV commissioning activities occur simultaneously, the orbit transfer phase must include the SV commissioning activities that will occur in the same time frame. A LYF test for a mission phase would aim to include key critical mission characteristics. Options for what is present during testing depend on the duration of the phase and the ability to emulate key mission phase objectives. It should be possible to include a test of the complete "ascent" phase. It should be feasible to perform—at some level—most, if not all, of an orbit transfer phase. Nominal operations, which comprise the bulk of the mission, should have an appropriately long (days, weeks) duration covering as many types of critical activities, scenarios, and timeline sections as feasible. Table 5 defines how each mission phase may have its own end-to-end configuration depending on the mission objective.

Table 5. Timeline and Configurations by Mission Phase

Mission Phase	Mission Objective	Timeline	End-to-End Configuration
Ascent	Successful ascent	T+0 through Launch Vehicle (LV) Separation	Space Vehicle (SV) + LV + Launch Vehicle Control (LC)
Automated Initialization	Perform automated initialization	LV Separation through first planned ground segment commanding	SV
Orbit Transfer	Transfer to Operational Orbit	First post separation orbit determination through park orbit	SV + Ground Segment (GS)
Commanded Initialization	Send initial command from GS to SV	First planned ground segment commanding through ready for ops	SV + GS
Normal Ops	Accomplish primary mission	Ready for ops through special ops or EOL	SV + GS
Normal Ops	Upgrade the Ground System software	Scheduled period of time on a non-interference basis	GS [Mission Control Center, Backup Control Center]
Fault & Contingency	Keep the vehicle safe and recover to normal ops	Any applicable mission phase	SV [if autonomous fault mgmt] or SV + GS

When reviewing mission timelines for LYF tests, it is important to consider fault and contingency operations (see section 2 for definitions*)*. While the other phases have a single expected timeline, the fault and contingency phases may have branches of activities based upon the particular anomaly that occurs. Faults and contingencies that are applicable in each of the regular mission phases will need separate candidate LYF tests for each phase.

A primary goal for the first step in the TLYF process is to identify a list of mission-critical events/activities and associated critical items needed for successful completion of the event/activity. Mission-critical events consist of nominal, first-time, critical, critical recurring, and contingency. A first-time activity is not only the literal first time a discrete activity is performed, but is also the first time a repetitive set of activities (e.g., nominal operations) is performed (e.g., ground system upgrades). Things tend to fail during these types of events. For example, shortly before the launch, faulty ground software failed to close a valve in the rocket's second stage pneumatic system. This system performs several actions, including operation and movements for the stage's steering engine. Investigators determined this ground software error before led to the failed rocket launch [25].

The first-time analysis is an iterative process that can be initially performed following the development of the mission concept of operations. It provides the basis for allocating activities to the LYF tests. Each first-time activity either needs to be included in an integrated test or accounted for in the LYF test exception analysis (i.e., mission-critical events that are not covered by test). This data will feed into both *Map Mission to LYF Tests* and *Perform Critical Fault Risk Management*.

Mission-critical activities must be successful in order to accomplish the primary mission objective(s). It's easier to formulate critical activities for space activities; however, there are ground activities that must be executed to support the SV mission-critical activities. In addition, ground critical activities include those necessary to sustain the health and safety of space assets or ensure continued operations of the space system. For SVs in early phases, almost every activity (for ground and space) is mission critical. In later phases, there may be first-time activities that are themselves not mission critical, but which must, nevertheless, be executed in order to provide the appropriate conditions for subsequent mission-critical activities. This may be the case for a ground system, where a critical upgrade is necessary while the system continues to conduct its mission or a new mission thread is added to the overall system.

4.2.4 Outputs

The output of *characterize the mission and system* depends on the system and mission maturity (how far along it is in the development process). Outputs at the RFP development phase should include a notional system concept and mission concept (including product or service descriptions; mission phases; a notional timeline; space assets; ground asset(s); and external interfaces, user systems and other ground external systems). As the system develops, outputs should expand to include identification of suppliers, system constraints, and operational tools (e.g., planning, analysis, and data processing and dissemination).

- Timeline of each mission phase and sequence of events
- List of mission-critical events/activities (nominal, first-time, critical, critical-recurring, and contingency)
- List of mission characteristics for each mission-critical event/activity
- Associated objective for each mission-critical event/activity or set of events/activities
- Mission-critical events/activities grouped by mission phase (operational sequence)

4.3 Do MCFA

Fault analysis is a technique typically used as part of a failure investigation, done AFTER the mission-critical situation has occurred. In a failure situation, the actual failure indication is known, e.g., no communication from the SV or SV in wrong orbit. The process herein includes a step to do mission-critical fault analysis. However, this analysis is done pre-mission, based on lessons learned from the back-to-back failures of the Mars Polar Lander (MPO) and Mars Climate Orbiter (MCO) in 1999. The program and mission management for the next project in line, Mars Odyssey, observed that the failure review teams were remarkably efficient at identifying a number of serious flaws that had escaped the normal review processes. The lead spacecraft engineer proposed that the Mars Odyssey engineering team meet as if they were a failure review board preflight, to hypothesize mission failures and identify possible contributors to those failures. Using a fault tree analysis (FTA) approach, they were able to identify a number of potential problems. These were investigated and mitigated as necessary. Other Mars projects have since used this technique as part of their design process [12]. The recognized utility and value of this technique in pre-positioning programs to effectively respond to off-nominal events makes this a best practice for use in the broader USG space enterprise.

Mars Polar Lander

Failure: The Mars Polar Lander crashed on the Martian surface. Its onboard HW and SW logic was supposed to sense touchdown and immediately shut down the descent engines. During development, an LYF test detected a HW problem. A design change was implemented; however, the test was not rerun. The failure investigation determined that the original HW problem had masked a second problem of HW/SW interaction. This logic was faulty and had never been tested in the flight configuration at the SV level because a decision was made not to retest after a repair.

Lessons Learned:

- Test *what* you fly. Ad hoc repairs are a frequent source of additional problems. Late repairs tend to have less rigorous review and control of procedures resulting in inadequate post-rework test of repaired items.
- Perform LYF tests across mode and phase transitions, as well as across the range of initial conditions.

Mars Climate Orbiter

Failure: The loss of Mars Climate Orbiter is a classic case of misunderstanding the criticality of ground planning tools. The vehicle impacted the surface of Mars rather than entering its orbit due to improper mid-course trajectory corrections. The vehicle expected commands based on metric calculations. One mission-planning tool, whose software was deemed not critical, used English units.

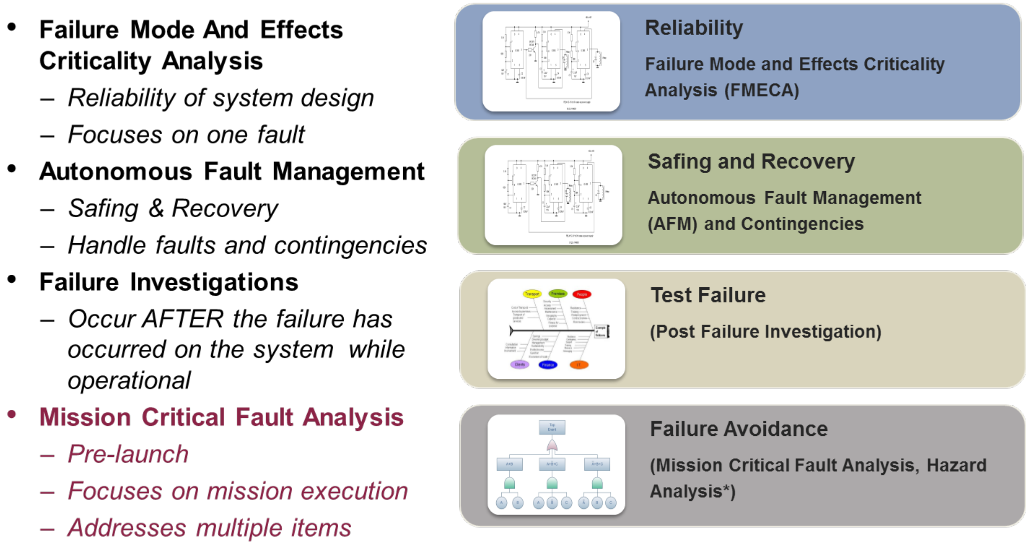
Lessons Learned:

- Critical failure analysis needs to include the entire system.
- Anything that touches/interacts with critical flight equipment and processes is itself critical.
- Conduct an end-to-end, operationally realistic test with mission configuration to find ground and flight interaction problems.

For the purpose of this guide, mission-critical fault analysis is defined as a pre-launch analysis that examines a system's operational timeline (discrete critical events and transitions) and focuses on what could go wrong with the mission (i.e., mission-critical failures). "It is a top-down analysis of potential failures that is performed before a system is fielded or flown" [12]. It addresses faults that may occur as the system executes its mission. Faults may involve multiple and concurrent items (hardware and software). It may use common fault analysis tools (e.g., fault trees, Ishikawa diagrams). It is an adjunct to a FMEA analysis or to a FMECA analysis.

4.3.1 Distinguishing Features

Besides performing the fault analysis prior to failure occurrence, MCFA can be distinguished from other failure methods. A summary of the key differences is highlighted in Figure 8.



- **Failure Mode And Effects Criticality Analysis**
  - *Reliability of system design*
  - *Focuses on one fault*
- **Autonomous Fault Management**
  - *Safing & Recovery*
  - *Handle faults and contingencies*
- **Failure Investigations**
  - *Occur AFTER the failure has occurred on the system while operational*
- **Mission Critical Fault Analysis**
  - *Pre-launch*
  - *Focuses on mission execution*
  - *Addresses multiple items*

Reliability

Failure Mode and Effects Criticality Analysis (FMECA)

Safing and Recovery

Autonomous Fault Management (AFM) and Contingencies

Test Failure

(Post Failure Investigation)

Failure Avoidance

(Mission Critical Fault Analysis, Hazard Analysis*)

Figure 8. MCFA and other fault analysis methods.

Most programs require a FMECA or FMEA as part of the design process. These provide formal and systematic approaches for identifying potential system failure modes, causes, and the effects of the failure mode's occurrence. FMECA/FMEA are commonly used as the basis for a HW reliability analysis and so do not address failure modes associated with system interactions (HW-HW, HW-SW, and SW-SW) or time-dependence. These are valuable bottoms-up design assessment techniques that systematically step through the HW design looking for downstream effects of component or part failure, particularly single-point failures. However, they are limited in their ability to assess system design because they do not consider the mix of HW, SW, process, procedure, timing, and human errors that may contribute to a failure. While there are forms of FMECA that can be created to address system-level flows, they are generally not required on NSS contracts.

Fault management (automated on the SV or ground-initiated) is not suitable for addressing defects related to architecture, design, manufacturing, assembly, or common cause [13]. When space assets are involved, ground contingencies are primarily oriented to vehicle safing and recovery than to fault identification/prevention.

A properly performed pre-failure (pre-mortem) mission-critical fault analysis provides a methodical roadmap for identifying fault paths and addressing, exonerating, or eliminating them before they occur. The systems engineering thinking that goes into automated fault management, contingency planning, and failure investigations can be applied to the MCFA. There is risk associated with the characteristics that are deemed critical to the mission that are not readily testable (i.e., it is an LYF test exception). If the obvious (but impractical or impossible) test cannot be performed, the risk of failing to detect mission-critical flaws increases. When an operationally realistic LYF test decreases use of mission aspects (articles, conditions) with nonflight/non-operational aspects, the likelihood of missing critical flaws increases. What cannot be exonerated or eliminated is a risk.

4.3.2 Mission Success and Critical Failures

Performing MCFA during the system design process provides many benefits for both LYF design and testing. A space system with complex flight and ground SW will have a large number of latent flaws and an almost infinite number of potential flaws. However, only some of the flaws will contribute to, or cause, mission-critical situations (i.e., when the right combination of attributes are present). Fault analysis helps identify which mission characteristics are critical and need to be present for the test, allowing potential critical defects to be exposed.

MCFA results are best derived by using root cause analysis methods like FTA or fishbones (also known as Ishikawa diagrams). Fishbones make it possible to visually construct and consider different items in parallel, but do not help identify or communicate how multiple bones jointly contribute. This becomes problematic when creating mission-ending scenarios where parallel processes occur. FTA, on the other hand, allows visual modelling of how various factors can combine to cause a failure. These techniques quickly focus on the most critical mission failures for the system and include a broader representation of failure contributors (i.e., accounting for all critical mission characteristics). This technique promotes a multi-disciplinary and interactive approach which allows subject matter experts (SMEs) to consider how the various parts of the system depend on each other.

In order to analyze what can go wrong, it is essential to be able to define what constitutes success. Recall that the output from the first step of the TLYF process includes a mission timeline and list of mission-critical events. These events are critical because they contribute to the overall success of the mission. Mission-critical failures flow from mission success goals. Mission goals may be divided into two categories [12]:

1. **Primary mission success**, which is the absolute minimum that the project must accomplish
2. **Full mission success**, which defines the baseline mission that the project is attempting to achieve

For the purpose of this paper, a mission-critical failure is defined as a condition that meets one or more of the following criteria:

- Failure leading to inability to meet/achieve mission objectives (e.g., payload, spacecraft bus, or ground element is no longer capable of supporting mission objectives)

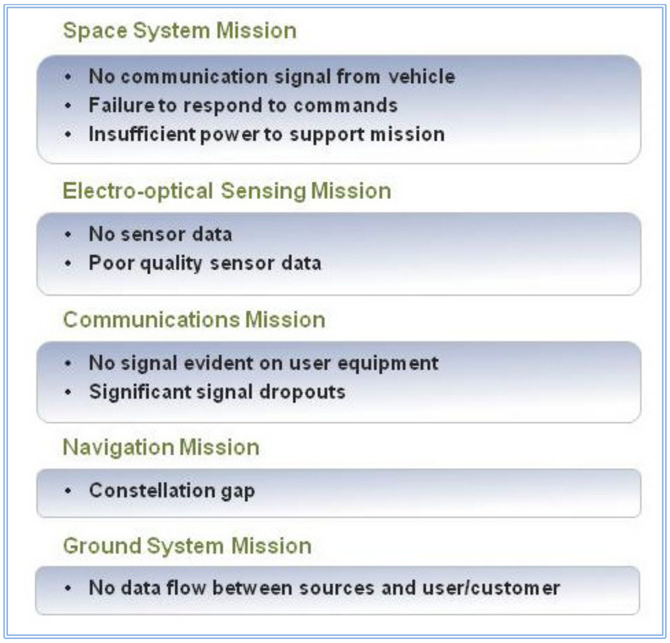
- Inability to meet minimum performance specifications

- Degrading condition whose trend indicates a loss of mission before mean mission duration (MMD) or design life

- Repetitive transient condition(s) that, if not corrected, would lead to an unacceptable loss of mission performance, data, or services (e.g., satellite with processor susceptibility to single-event upsets in orbit with mean-time-to-upset much less than mean-time-to-recovery-from-upset)

- Inability to correctly generate, process, and/or transmit data to requester (e.g., end user of the mission data)

Examples of mission-critical failures for the various types of missions are highlighted in Figure 9.



Space System Mission

- No communication signal from vehicle
- Failure to respond to commands
- Insufficient power to support mission

Electro-optical Sensing Mission

- No sensor data
- Poor quality sensor data

Communications Mission

- No signal evident on user equipment
- Significant signal dropouts

Navigation Mission

- Constellation gap

Ground System Mission

- No data flow between sources and user/customer

Figure 9. Examples of mission-critical failures by mission type.

4.3.3 Process Details

MCFA provides a mechanism for identifying mission risk unlike other failure analyses. It uses the system's operational timeline to identify mission-critical events and transitions. It starts at the broadest grouping of the mission timeline—mission phases—and decomposes it down to discrete supporting events/activities. This allows a closer look at the system down to the subsystem, component, actions, discrete events, and conditions, as necessary. The MCFA determines the fault paths and flaws that could prevent success for each of these critical events. If such a potential flaw exists, it must either be exonerated by LYF testing, discovered by some means and mitigated accordingly, or identified as a risk to mission success.

The following MCFA steps are based on the Mars Climate Orbiter and Mars Polar Lander lessons learned [12]:

1. **Assemble the team**. A diverse team of SMEs led by a systems engineer/senior engineer is important for success. Key representatives may include, but are not limited to, the following domains of expertise: mission design, navigation, operations, test, integration, spacecraft systems, spacecraft subsystems, software, payloads, and ground system. Different perspectives will facilitate the discovery of flaws caused by unintended interactions or dependencies.

2. **Establish a schedule**. Once the team is formed, MCFA is best executed within boundaries of time and schedule. Depending on the program, associated milestones, and constraints, it is helpful to determine expected outcomes up front.

3. **Conduct the failure investigation prior to launch/mission execution.** The team, or a subgroup of experts (depending on the fault focus) gather to execute the following:

    a. **Identify critical events and mission characteristics.** The team examines the system's operational timeline and identifies mission-critical events and transitions. For each critical event, all critical mission characteristics are identified (see section 4.2.2).

    b. **Identify mission-critical situations and potential mission-ending failures.** The team identifies and develops mission-critical situations and models failure conditions with a fault tree/event tree. The top event of the tree is usually a specific mission failure. The failure conditions are reflected in the basic events and logic gates of the fault tree.

    "How deep should the tree go? It is best to stop the analysis when the mitigations become the same. At some point all the faults below a certain fault event will have the same mitigation [12]."

    It is important to note that mission failure is not restricted to a spacecraft or payload failure. It can be something that disrupts operations to the point that the mission cannot be successfully executed:

        i. Mission-critical events and transition events, such as initialization, activation, data collection, data transfer, calibration, orbit insertion, or second stage ignition

        ii. Mission-ending situations, such as the inability to command or upload SW patch, no data to ground, or second stage failure

    c. **Identify potential contributors.** The team identifies flaws that can contribute to each condition and form fault paths to failure:

        i. Possible contributors to failure, e.g., items, processes, transactions, parallel activities, or interfaces

        ii. Types of contributors, e.g., single and multiple contributors, interconnections, transactions, or environments

        iii. Locations these failures are likely to occur, e.g., hardware, software, procedures, or processes

        iv. When these failures are likely to occur, e.g., what mission phase

        v. The root cause(s) of each mission failure

4. **Identify critical fault path exoneration methods.** Once fault trees or fishbone diagrams are defined, it can be determined what kind of evaluation or method is necessary to exonerate branches or bones (flaws), and thus lower the risk of the failure condition occurring. If mitigation is not possible, the risk should be added to the program's formal risk management process (see section 4.7).

4.3.3.1 Fault Tree Mechanics—Example

An example of a very high-level fault tree is shown below. The mission failure is defined as “space vehicle dead on arrival to orbit,” a critical event. The potential failure situations leading to the mission-ending event are identified at the next level of the tree. Once the failure situations are defined, the branches and leaves are populated with relevant mission characteristics. This example is not exhaustive but shows the development process for fault trees. The fault tree would need to be populated with “leaves” that capture lower-level critical mission characteristics that contribute to each failure situation.

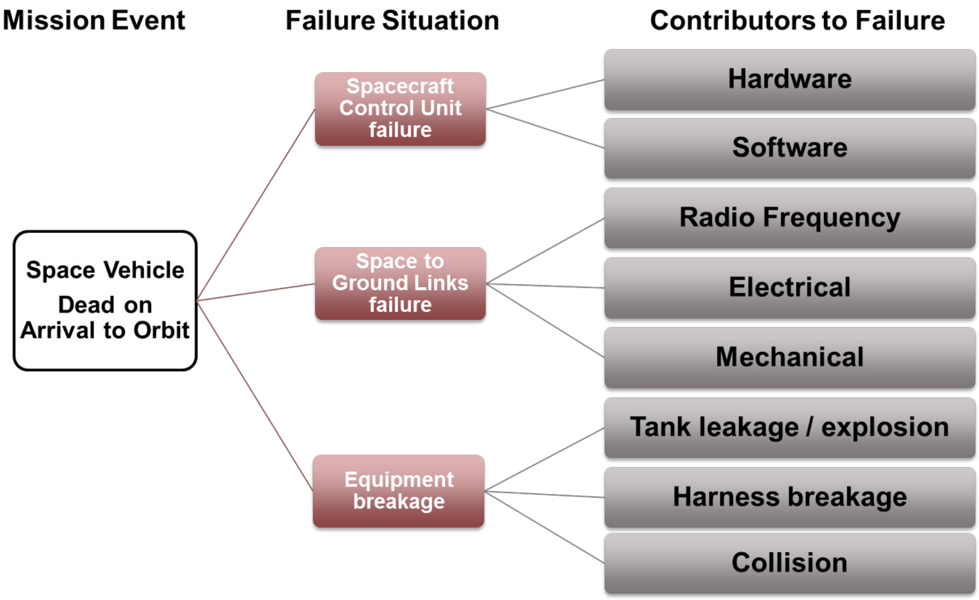


Figure 10. Sample: Mission event, failure situations, and contributors.

4.3.4 Outputs

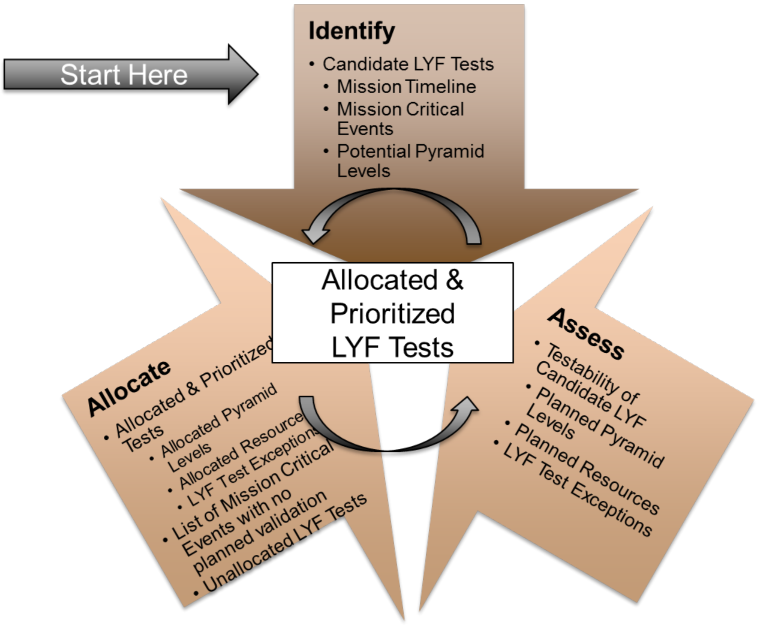
The result of conducting MCFA for LYF test development includes the following:

- List of mission-ending failure situations (including first-time and mission-critical events) covering all mission phases (fish head/tree tops)
- For each failure situation (includes successful completion of first-time/mission-critical events), the following is generated:
  - Fish head/tree top signifying the failure situation
  - Fish bones/branches for potential fault paths and contributors (including parallel contributors and processes)
  - Exoneration plan for each failure path (I, A, D, and T)
  - Exoneration methods for each failure situation (includes successful completion of first-time/mission-critical events) (I, A, D, and T)
- List of mission-critical events and fault paths not planned to be exonerated (these become risks)

In the context of the TLYF process, the MCFA provides the insight necessary to prioritize and scope LYF testing. It also provides the framework for evaluating LYF test exceptions and determining if those exceptions must be handled as risks.

4.4 Map Mission to LYF Tests

The next step in the process, *Map Mission to LYF Tests*, contains three key LYF test building activities (identify, assess, and allocate), as shown in Figure 11. The purpose for the distinction is to capture the cyclical interdependency that occurs among these actions leading up to the initial list of accepted (allocated and prioritized) LYF tests. The first-time and mission-critical events lists, the mission phases and their objectives, and the MCFA results form the basis to identify and ultimately prioritize candidate LYF tests.



Start Here
Identify
• Candidate LYF Tests
• Mission Timeline
• Mission Critical Events
• Potential Pyramid Levels
Allocated & Prioritized LYF Tests
Assess
• Testability of Candidate LYF
• Planned Pyramid Levels
• Planned Resources
• LYF Test Exceptions
Allocate
• Allocated & Prioritized Tests
• Allocated Pyramid Levels
• Allocated Resource
• LYF Test Exception
• List of Mission Critical Events with no planned validation
• Unallocated LYF Tests

Figure 11. *Map Mission to LYF Tests* activities.

The inputs for this step include all the necessary system and mission information extracted from the *Characterize the system and mission* and *Do mission-critical fault analysis* outputs. In addition, test program work products are used. Types of test program products include, but are not limited to, test program details (requirements, contractual products, and guidance), test tools and resources, and test strategy (e.g., test and evaluation master plan [TEMP]).

4.4.1 Process Details (Identify, Assess, Allocate)

A multi-disciplined team of systems engineering, mission engineering, and system integration and test personnel is needed to effectively map the mission to operationally realistic LYF tests. The team will need to extract mission objectives, identify candidate LYF tests, assess candidate LYF tests for testability, and allocate specific candidate LYF tests. During this activity, there are three distinct "pyramids" defined for the TLYF process that will be utilized. It is important to understand these pyramids before discussing the specific actions involved in mapping the mission to LYF tests.

4.4.1.1 Pyramid Approach

The TLYF process uses three pyramids: the system-integration pyramid, the provider/supplier pyramid, and the mission-level pyramid. Pyramid tops represent the highest complexity and pyramid bases represent the lowest level of simplicity. Each pyramid provides a context for designing operationally realistic LYF tests.

4.4.1.1.1 System-Integration Pyramid

Most are familiar with the system-integration pyramid. It is introduced in Chapter 1, *Test and Evaluation Handbook Overview*, Space Vehicle Test and Evaluation Handbook, 2nd Edition, Volume 1 [19]. It shows the progression of integration and testing for a space vehicle system (from lowest level to highest: part and subassembly, unit, subsystem or module, system, and in-orbit). Each level of testing has unique verification purposes.

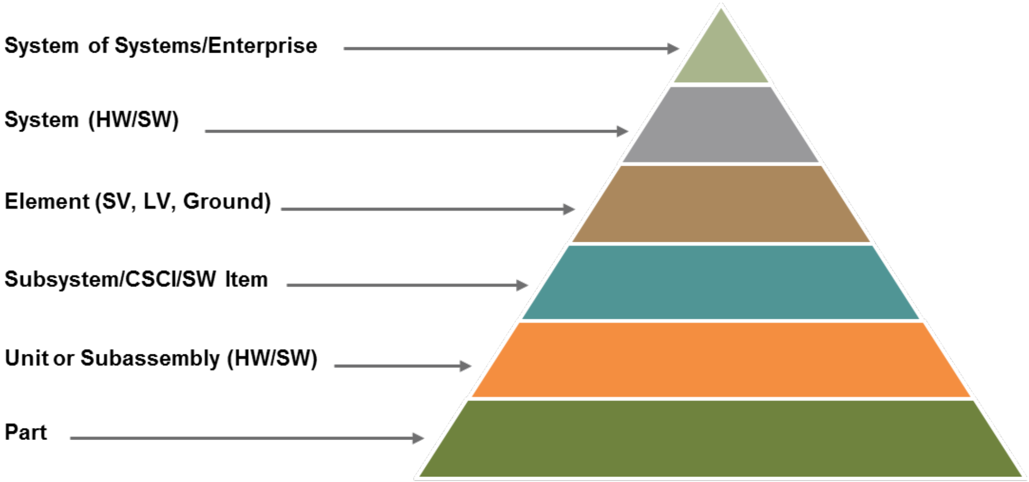


Figure 12. System-integration pyramid.

The top of the integration pyramid is typically the launched SV in its operational environment with the final flight HW and SW. The TLYF process utilizes this structure and extends it to include HW integration, SW integration, combined HW and SW integration, and up to the entire enterprise (see Figure 12). For ground systems, this may involve multiple ground and user elements in their operational arrangement. The enterprise level of integration addresses the merging of existing systems with new acquisitions in the same mission domain, such as legacy space vehicles with new blocks of space vehicles and ground system. It includes several mission domains working together to provide mission data or services. For instance, an enterprise test may include a low-Earth-orbiting set of SVs and that mission's dedicated ground control, plus the Air Force Satellite Control Network (AFSCN) and/or geosynchronous communications satellites. Suggested definitions for each of the integration levels for the system pyramid are provided in section 2.

The TLYF process leverages the fact that testing at different levels of a system's physical integration provides an opportunity to discover different types of mission-ending flaws. There are logical ties between each failure and particular mission characteristics. Table 6 provides a sample of unclassified mission-critical failures, the anomaly root cause, and the detectable level of the integration pyramid. Each anomaly listed in the table occurred after liftoff on a spacecraft or launch vehicle that had its requirements, functionality, and performance verified prior to launch. These vehicles were ready to go, had passed their "requirements-centric" tests, and been judged flightworthy, but failed anyway.

Table 6. Failing to Test Like You Fly along the Integration Pyramid

	Mission Critical Anomaly & Root Cause	TLYF Issue	Integration Level of Flaw Detectability	Applicable Mission Characteristics
Titan CT-2	Failure to separate SV. Miswire/numbering error for single payload	How & What We Fly	Integrated LV & SV	Timeline, sequence, configuration, command
Ariane V	Inertial Reference System disabled. "Dead code" inherited from Ariane IV	What We Fly	Integrated Flight SW & Control Subsystem	Sequence, end-to-end level, fault management
ESEX Arcjet	Battery explosion. "Heritage" battery & charging system not able to sustain unique charging scheme	How We Fly	Payload Power Subsystem	Duration
AV-009	Wrong orbit. Engine fuel inlet valve did not close fully at end of first burn, resulting in overboard fuel leak during coast phase	How We Fly	Valve Assembly	Duration, internal environment

4.4.1.1.2 Provider/Supplier Pyramid

The provider/supplier pyramid addresses the issue of LYF testing for product suppliers and integrators. A mission failed because a subassembly designed for continuous operations (100-percent duty cycle) was delivered to a system that intended to use it at a distinct duty cycle. The subassembly supplier noted that had they been informed of the mission usage, they would have provided a different subassembly. There are many other examples of mission failures that occurred as a result of suppliers not being aware of the usage of their products in the mission, a clear failure in requirements flowdown and subcontract management.

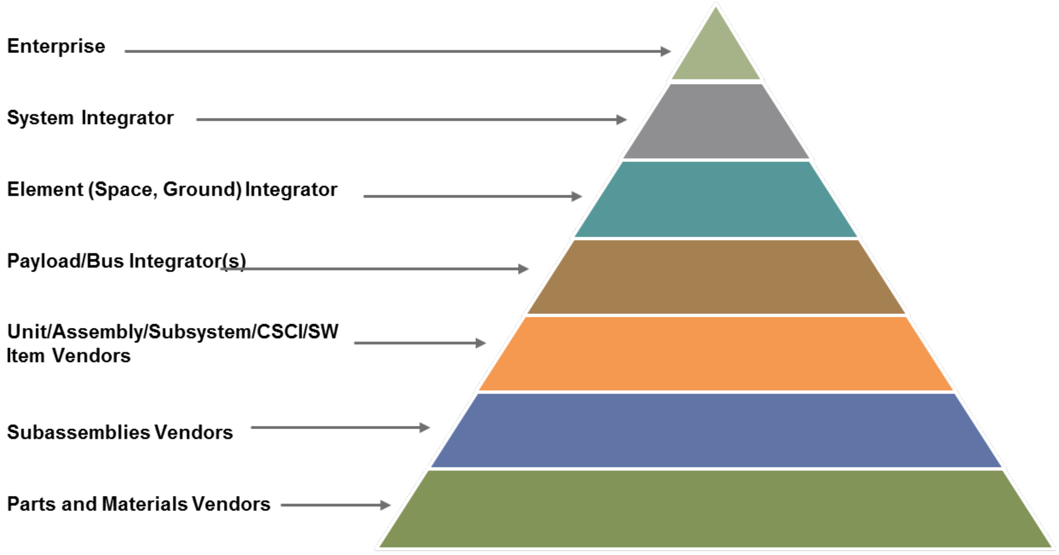

Enterprise
System Integrator
Element (Space, Ground) Integrator
Payload/Bus Integrator(s)
Unit/Assembly/Subsystem/CSCI/SW Item Vendors
Subassemblies Vendors
Parts and Materials Vendors

Figure 13. Provider/supplier pyramid.

The provider/supplier pyramid, shown in Figure 13, depicts a recommended progression for applying mission operations requirements down the supply chain with an associated validation of that usage. For example, a ground system may have something akin to the "payload" level to address core ground entities (e.g., command and control, mission management, mission data processing, and infrastructure). A LYF test for lower levels (i.e., parts and materials vendors) demonstrates readiness to conduct mission activities at the next higher level of integration. Tests along the supplier chain are necessary to reduce risk prior to the higher-level tests. These can provide the opportunity for certain kinds of flaws to be uncovered using the supplier test facilities and special test equipment.

4.4.1.1.3 Mission-Level Pyramid

Finally, the mission-level pyramid, shown in Figure 14, becomes necessary to show the progression of mission complexity. Complexity increases from basic mission activities, through mission threads, mission scenarios, and up to a full mission timeline. The pyramid may be used two ways: (1) from the top to decompose mission phases into executable events or (2) from the bottom to provide context for lower-level mission activities (i.e., how, where, and when they fit in the overall mission timeline).

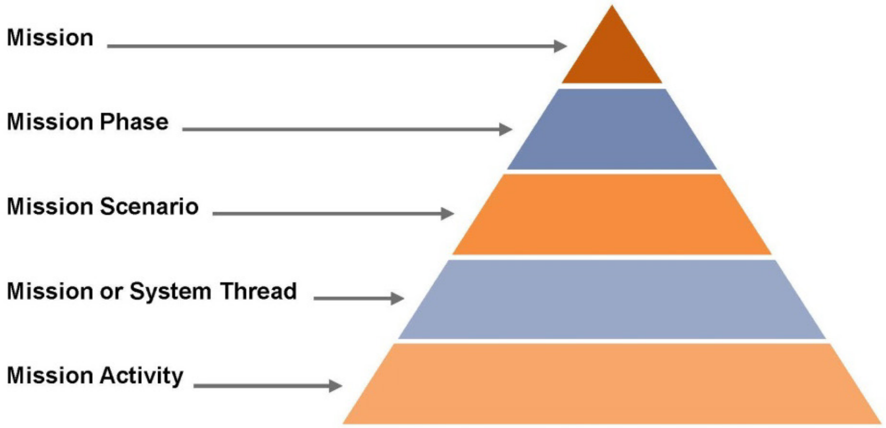


Figure 14. Mission-level pyramid.

The mission-level pyramid provides a framework for decomposing the mission timeline (pyramid top) down to a series of distinct activities/events (pyramid base). It highlights the need for developing tests that involve multiple threads and concurrent scenarios to more fully represent what occurs in executing a mission timeline. Activities, threads, and scenarios need to be accounted for in the test approach for risk reduction, but are not likely to be adequate representations of the operational complexities that occur in mission execution. Depending on the scenario or thread, the specific order of supporting mission activities may differ. That said, it may be that some LYF tests will need to be done at these lower levels of concurrence or duration, in which case particular attention must be paid to the types of flaws that may be missed by not being able to execute a more flight-like timeline.

For the process purpose, definitions of the terms used in the mission-level pyramid are provided below. Note that these are not industry-accepted definitions, but are included for describing the creation of operationally realistic LYF tests:

- **Mission/Mission Timeline:** A time-sequenced list of events of a satellite beginning at launch and extending to end-of-life.

- **Mission Phase:** A distinct division of the overall satellite mission. Typical mission phases include testing, launch and early orbit, initial, payload checkout, satellite in-orbit, and end-of-life.

- **Mission Scenario:** A mission scenario is an operational sequence within a given mission phase. The sequence consists of an ordered list of functions/activities required to achieve a distinct mission objective.

- **Mission or System Thread:** A time-sequenced slice of a mission scenario in which the system explicitly supports actions to be executed by the user. Each mission thread is a story composed of the events, actions, stimuli, information, and interactions applicable to utilizing the deployed system within the context of its place in the overall enterprise architecture. Mission threads include descriptions of the system's role in sample user missions and information regarding interactions with the user, interfaces to other systems, and any pertinent system states or modes. Mission threads also provide a framework for the system threads, mapping specific system threads to mission thread steps.

- **Mission Activity:** A discrete event within a mission phase. A sequence of mission activities may constitute a mission scenario, thread, or phase.

Figure 15 shows a notional mission thread for a satellite flight software (FSW) upload. Each box represents a discrete mission activity. LYF tests designed from this thread are used to verify the FSW upload capability and address questions such as whether the system is able to update on-board satellite SW subsystems and how the system will verify accuracy of image transfer.

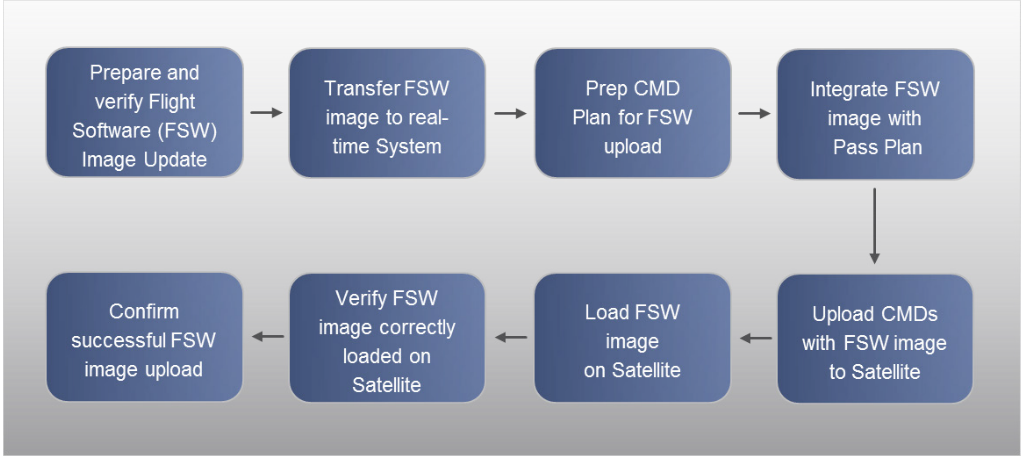

Prepare and verify Flight Software (FSW) Image Update
Transfer FSW image to real-time System
Prep CMD Plan for FSW upload
Integrate FSW image with Pass Plan
Upload CMDs with FSW image to Satellite
Load FSW image on Satellite
Verify FSW image correctly loaded on Satellite
Confirm successful FSW image upload

Figure 15. Notional mission thread.

A scenario is a higher-level construct than a thread. An example of a basic mission scenario for a remote sensing system is provided in Figure 16. In this example, the mission scenario is the detection, acquisition, and reporting of a missile launch with a subsequent return to the initial, quiescent stare mode. Each box represents a set of discrete mission activities required to accomplish the task.

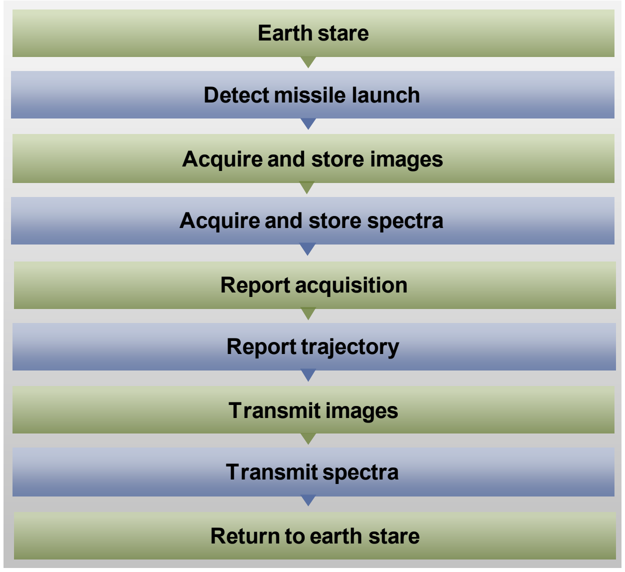


Figure 16. Notional mission scenario.

Using Mission-level Pyramid Example

The mission-level pyramid can be used to develop tests for satellite subsystems. Starting at the top of the pyramid and moving downward, the mission phase "Payload Initialization & Checkout" can be decomposed down to the mission scenario "Mode 1" and further down to a series of discrete activities performed by the payload (see Figure 17). The decomposition allows the test designer to replicate the detailed mission activities for the payload subsystem within the context of the mission timeline (entry and exit points).

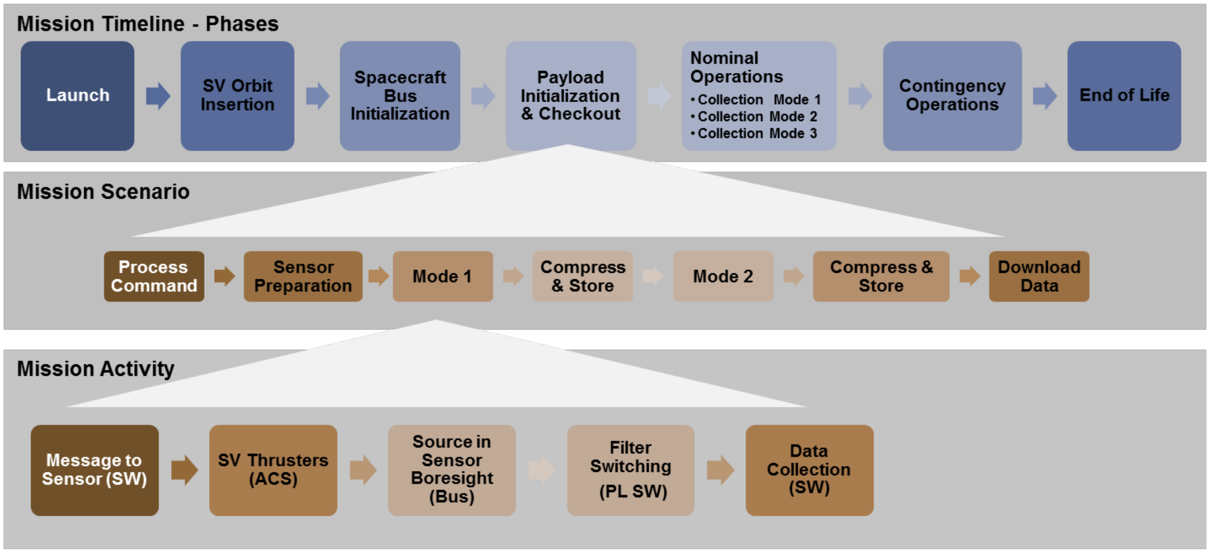


Figure 17. Notional mission decomposition (payload activities).

4.4.1.2 Identify Candidate LYF Tests

4.4.1.2.1 Mission-Critical Events and Mission Characteristics

A driving motivation for beginning the TLYF approach at the program start is to be able to allocate operationally realistic LYF tests and resources at the appropriate levels of development and integration. The systems engineering team evaluates the mission, selecting mission situations that are critical and should be subject to LYF tests.

By using the pyramids with an understanding of the system and its mission, mapping the mission to LYF tests can begin. The first activity in mapping the mission to LYF tests is to **identify** candidate tests. The critical mission events (e.g., first-time events and operations, recurring critical events, operational sequences and mission phases—including fault handling and contingencies) and associated characteristics provide the framework for building candidate LYF tests.

The process of identifying candidate tests is subsequently done by individual organizations, contractors, and integrated systems engineering teams at design milestones. It is refined again as operations concepts, requirements, and designs change and mature. System engineers, design engineers, and other mission staff should propose candidate LYF tests that reflect their concerns for adequate mission validation.

Candidate LYF tests will depend on the supplier-level perspective. Table 7 shows a notional sequence of critical events for SV ascent mission from the perspective of the (1) SV contractor, (2) LV contractor, and (3) launch control organization, demonstrating how the supplier perspective will affect what defines success. For example, the LV integrator will view the ascent and injection phases as their entire mission and will look for proper execution of engine-related events. A SV integrator with a SV that is active during ascent will view this phase as received data and properly respond to LV inputs and observe flight SW decision processes linked to elapsed time or sensed conditions. An integrator responsible for bringing LV, SV, ground system, and range assets will view the goal of this phase as requiring all elements to provide appropriate information at expected times and respond properly to information received.

Table 7. Notional Ascent First-Time and Mission-Critical Events

Ascent (Space Vehicle)	
Timeline	**Critical Event**
Prelaunch	Configure Spacecraft Computer Unit (SCU)
T – 10 hrs	SW patch and database upload
T – 4 hrs	Initialize configuration for launch
T – 10 min	Switch to internal power
T + 0	Launch signal for SCU timer
T + 613 sec	Turn on SGLS transmitters
T + 58 min	Separation
T + 2 hrs	Solar Array deploy and SV sun pointing attitude
T + 5 hrs	SV Telemetry (via S-Band)

Ascent (Launch Vehicle)	
Timeline	**Critical Event**
Prelaunch	Initialize configuration for launch
T + 0	Liftoff
T + 82.5 sec	Graphite epoxy motor (GEM) jettison
T + 264 sec	Main Engine Cutoff
T + 277.5 sec	Second Stage ignition
T + 281.5 sec	Fairing jettison
T + 685.1 sec	Second Stage Engine Cutoff
T + 58 min	Separation

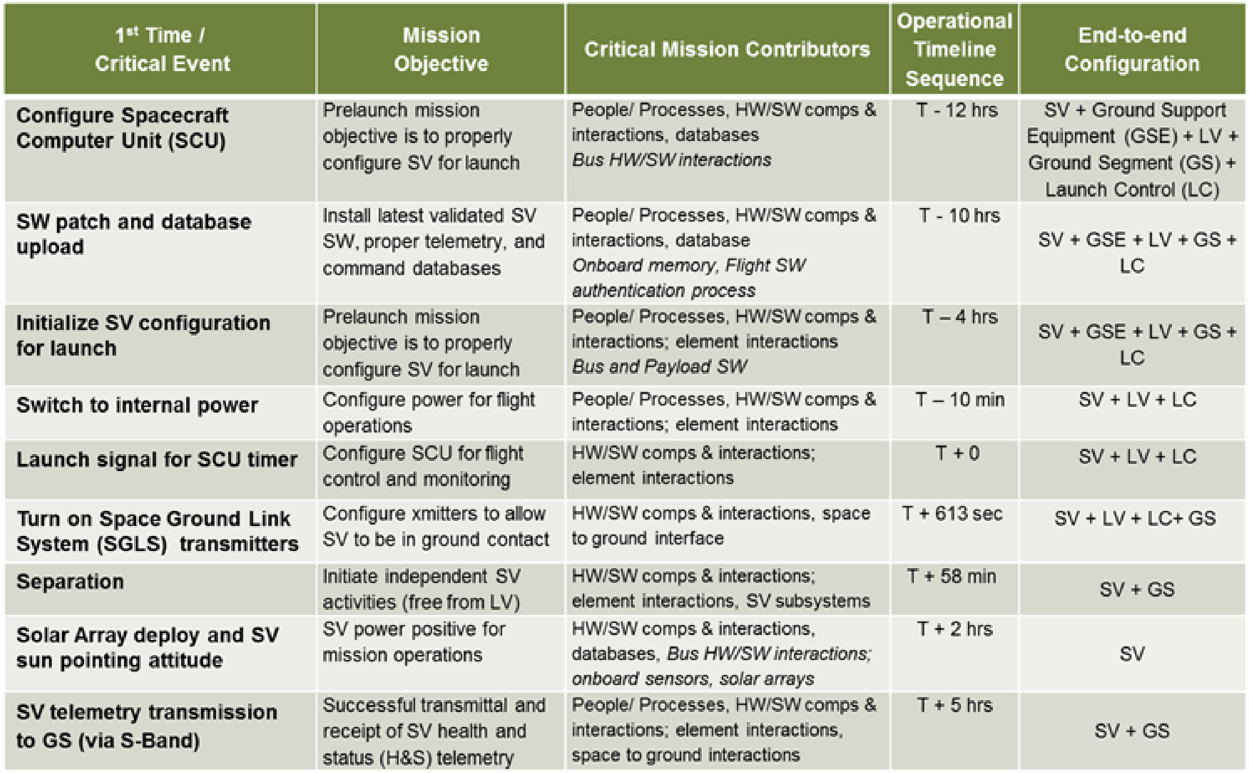
Ascent (Launch Control)	
Timeline	**Critical Event**
T – 5 min	LV Flight Termination System Check
T – 4 min	Receive weather briefing
T – 2 min	Polled for Go/No-go
T – 2 sec	LV engine thrust monitored
T + 0	LV committed to liftoff
T + 10 sec	Monitor LV ascent trajectory parameters
T + 65 min	Preliminary SV State Vector to SV Ground Segment

From the SV point of view, there are two critical events that occur quickly after launch: (1) receipt of the launch signal to start the SCU time and (2) turn-on of Space Ground Link System (SGLS) transmitters. In the background, the SV must check for health of powered units and implement automated fault management if warranted. The SV must successfully execute ascent activities (either autonomously or by command), and be ready to conduct SV activities after separation. The transition from the ascent phase to SV autonomy is critical. In the case above, the critical events continue, including SV solar array panel deployment facing the sun and SV transmission of telemetry to the ground. The LV, on the other hand, performs a complex series of actions during ascent and must achieve the mission orbit while maintaining the integrity of the payload. These mission-critical events at the SV/LV levels are excellent LYF test candidates. In parallel to the SV ascent are ground support activities: the LV ground crew is monitoring for safety (i.e., the critical assets to ensure successful orbital insertion) and the SV ground crew is monitoring the health of the SV.

Different contracts for the different providers will result in different LYF tests, with different objectives, even for the same mission phase. At the systems-of-systems level, different interactions between government agencies will result in different LYF tests. These considerations are discussed further in section 5.

Table 8 provides a notional example of the kinds of mission information that should be recorded to assist in generation of relevant candidate LYF tests. The matrix begins with listing mission-critical events and identifying associated details for each event (i.e., objective critical mission characteristics, duration, sequence, and configuration). Each candidate LYF test should address, at a minimum, the mission information highlighted in each column heading of the table. Portions of the design reference mission timeline may be included as well as components of the system (especially the components that represent the physical "ends" of the test).

Table 8. Sample of Mission Aspects for Candidate LYF Tests (Ascent Phase—SV Perspective)



1st Time / Critical Event	Mission Objective	Critical Mission Contributors	Operational Timeline Sequence	End-to-end Configuration
Configure Spacecraft Computer Unit (SCU)	Prelaunch mission objective is to properly configure SV for launch	People/ Processes, HW/SW comps & interactions, databases *Bus HW/SW interactions*	T - 12 hrs	SV + Ground Support Equipment (GSE) + LV + Ground Segment (GS) + Launch Control (LC)
SW patch and database upload	Install latest validated SV SW, proper telemetry, and command databases	People/ Processes, HW/SW comps & interactions, database *Onboard memory, Flight SW authentication process*	T - 10 hrs	SV + GSE + LV + GS + LC
Initialize SV configuration for launch	Prelaunch mission objective is to properly configure SV for launch	People/ Processes, HW/SW comps & interactions; element interactions *Bus and Payload SW*	T – 4 hrs	SV + GSE + LV + GS + LC
Switch to internal power	Configure power for flight operations	People/ Processes, HW/SW comps & interactions; element interactions	T – 10 min	SV + LV + LC
Launch signal for SCU timer	Configure SCU for flight control and monitoring	HW/SW comps & interactions; element interactions	T + 0	SV + LV + LC
Turn on Space Ground Link System (SGLS) transmitters	Configure xmitters to allow SV to be in ground contact	HW/SW comps & interactions, space to ground interface	T + 613 sec	SV + LV + LC+ GS
Separation	Initiate independent SV activities (free from LV)	HW/SW comps & interactions; element interactions, SV subsystems	T + 58 min	SV + GS
Solar Array deploy and SV sun pointing attitude	SV power positive for mission operations	HW/SW comps & interactions, databases, *Bus HW/SW interactions; onboard sensors, solar arrays*	T + 2 hrs	SV
SV telemetry transmission to GS (via S-Band)	Successful transmittal and receipt of SV health and status (H&S) telemetry	People/ Processes, HW/SW comps & interactions; element interactions, space to ground interactions	T + 5 hrs	SV + GS

4.4.1.2.2 Mission Aspects for Critical Events

Candidate LYF tests may be oriented to a complete mission phase, portions of a long timeline, specific mission activities/threads/scenarios, or to particular fault situations. The following activities, from the top of the mission-level pyramid, are likely candidate LYF tests:

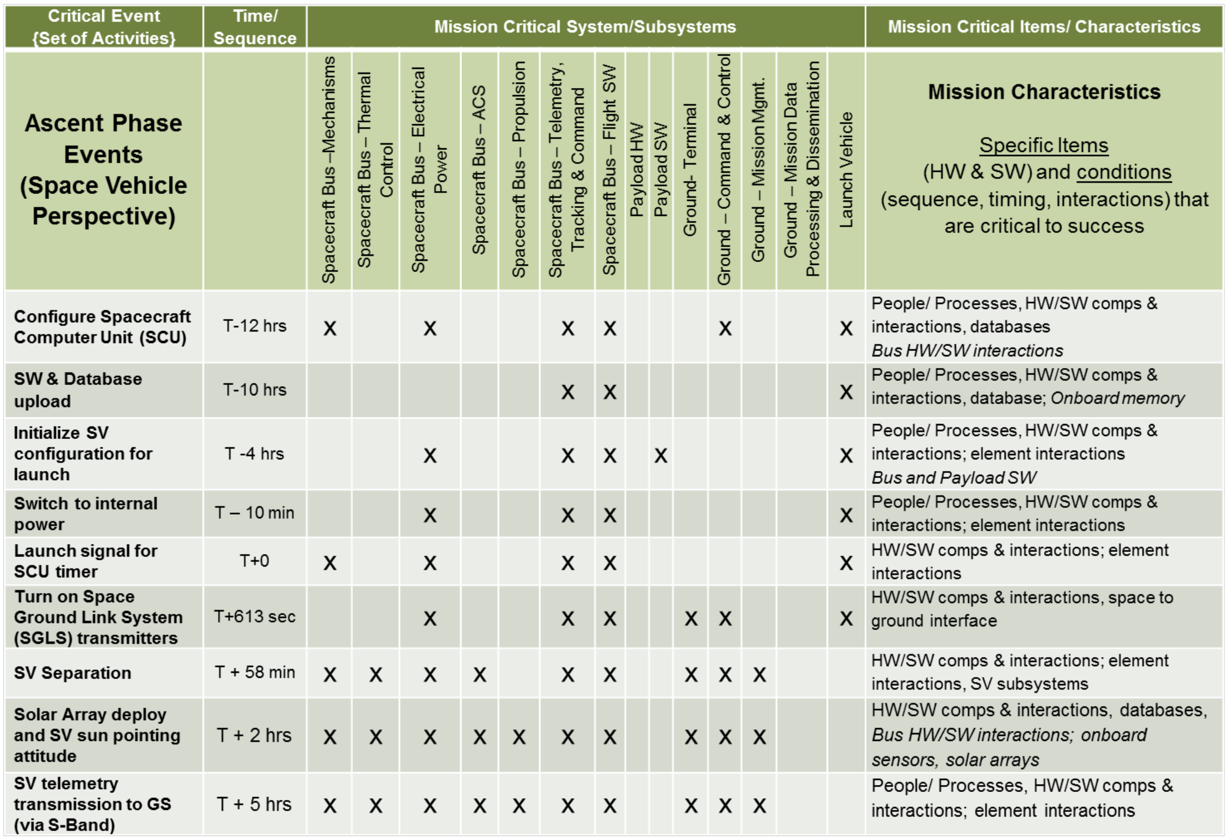
- A series of mission-critical events, including transitions (i.e., a week in the life)
- SV stressing operations where system capacity is exercised (i.e., a really bad day in the life)
- LV ascent phase
- SV orbit transfer
- SV automated initialization
- Payload initialization and checkout
- Critical simultaneous activities (i.e., nominal ground operations and ground system upgrade)
- Defined fault situations (programmed/planned)
  - Automated fault management
  - Redundancy management (execute backup component/subsystem and return to nominal)
  - Entry into/exit from safe mode
  - Error detection and correction
  - Operational planned response contingencies

It is also necessary to consider candidate LYF tests that cross mission phase boundaries (i.e., transitions between phases) and account for variations in boundary conditions for those transitions. Initial conditions for the test must be specified to reflect flight/operational conditions. Where specific events, such as solar array deployment, cannot be included at the higher integration level in vacuum, the critical event is assessed for testability at lower levels where it is feasible to replicate the critical characteristics in test.

The two versions of first-time activities (the literal first time of an activity as well as the first time a sustained activity is run for some duration) are both used as the basis for operationally realistic LYF tests. The second version is necessary to flush out accumulation and asynchronous timing errors. These errors may need more than a single occurrence to allow detection. For this type of candidate test, critical mission characteristics may include duration, number of repetitions in a mission-appropriate time period, system loading, and sequence of activities.

In addition to mission activities, it is necessary to identify what mission-critical contributors from the system details will need to be included in each candidate LYF test. Table 9 provides a sample template for pinpointing critical contributors. In addition to the system information, MCFA can bring significant value, especially when LYF test exceptions are identified.

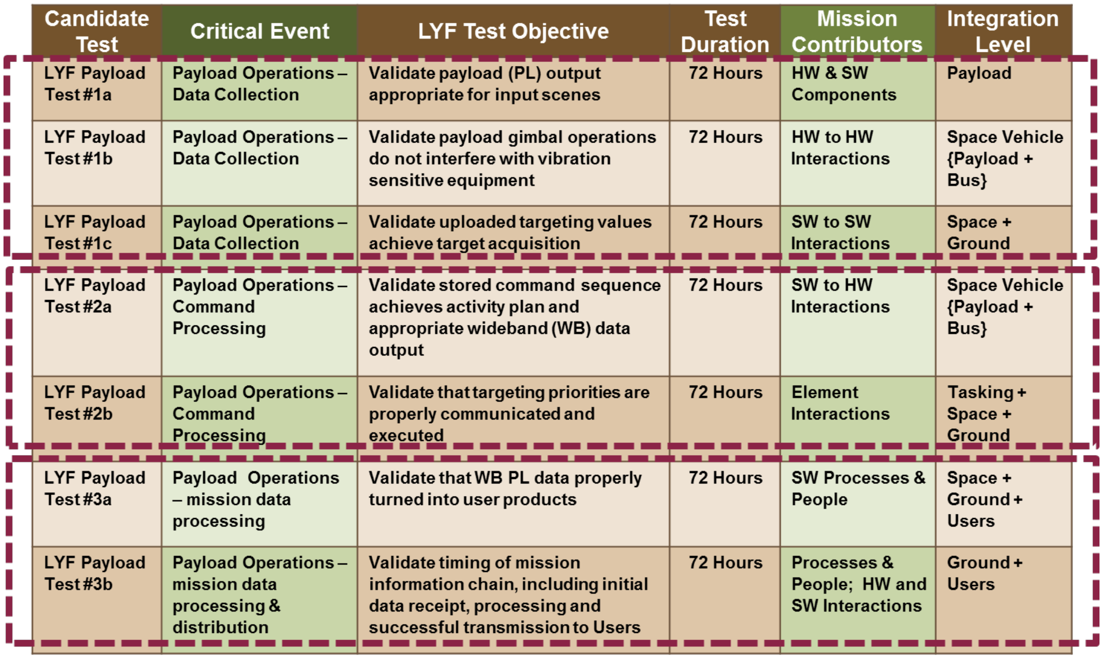
Table 9. Sample Template of Ascent (SV Perspective) Mission-Critical Contributors



Critical Event {Set of Activities}	Time/ Sequence	Mission Critical System/Subsystems														Mission Critical Items/ Characteristics
Ascent Phase Events (Space Vehicle Perspective)		Spacecraft Bus –Mechanisms	Spacecraft Bus – Thermal Control	Spacecraft Bus – Electrical Power	Spacecraft Bus – ACS	Spacecraft Bus – Propulsion	Spacecraft Bus – Telemetry, Tracking & Command	Spacecraft Bus – Flight SW	Payload HW	Payload SW	Ground- Terminal	Ground – Command & Control	Ground – Mission Mgmt.	Ground – Mission Data Processing & Dissemination	Launch Vehicle	**Mission Characteristics** Specific Items (HW & SW) and conditions (sequence, timing, interactions) that are critical to success
Configure Spacecraft Computer Unit (SCU)	T-12 hrs	X		X			X	X				X			X	People/ Processes, HW/SW comps & interactions, databases *Bus HW/SW interactions*
SW & Database upload	T-10 hrs						X	X							X	People/ Processes, HW/SW comps & interactions, database; *Onboard memory*
Initialize SV configuration for launch	T -4 hrs			X			X	X		X					X	People/ Processes, HW/SW comps & interactions; element interactions *Bus and Payload SW*
Switch to internal power	T – 10 min			X			X	X							X	People/ Processes, HW/SW comps & interactions; element interactions
Launch signal for SCU timer	T+0	X		X			X	X							X	HW/SW comps & interactions; element interactions
Turn on Space Ground Link System (SGLS) transmitters	T+613 sec			X			X	X			X	X			X	HW/SW comps & interactions, space to ground interface
SV Separation	T + 58 min	X	X	X	X		X	X			X	X	X			HW/SW comps & interactions; element interactions, SV subsystems
Solar Array deploy and SV sun pointing attitude	T + 2 hrs	X	X	X	X	X	X	X			X	X	X			HW/SW comps & interactions, databases, *Bus HW/SW interactions; onboard sensors, solar arrays*
SV telemetry transmission to GS (via S-Band)	T + 5 hrs	X	X	X	X	X	X	X			X	X	X			People/ Processes, HW/SW comps & interactions; element interactions

Once the key mission aspects are identified, associated columns can be added to identify test aspects. Different candidate LYF tests may be proposed to focus on one or more of these, even when the tests cover the same mission objective/phase/activity. The focus will be indicated by differing test objectives and can be proposed for different levels of integration. A program-established budget that allows for a three-day LYF test to be planned, designed, and executed should be considered. From that prerequisite, several candidate LYF tests spanning a three-day period can be envisioned for each contributor and integration level, as depicted in Table 10. In addition, keeping the mission aspects and test aspects side-by-side, makes it easier to identify differences between the mission and each test.

Table 10. Candidate LYF Tests for Payload Operations

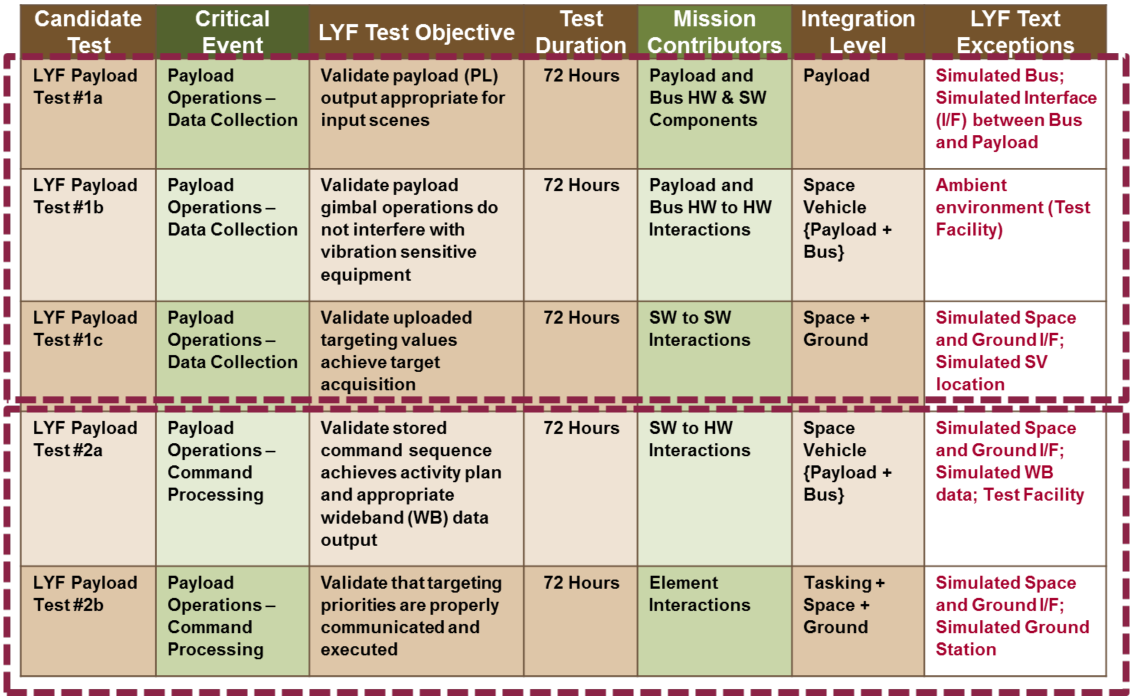


Candidate Test	Critical Event	LYF Test Objective	Test Duration	Mission Contributors	Integration Level
LYF Payload Test #1a	Payload Operations – Data Collection	Validate payload (PL) output appropriate for input scenes	72 Hours	HW & SW Components	Payload
LYF Payload Test #1b	Payload Operations – Data Collection	Validate payload gimbal operations do not interfere with vibration sensitive equipment	72 Hours	HW to HW Interactions	Space Vehicle {Payload + Bus}
LYF Payload Test #1c	Payload Operations – Data Collection	Validate uploaded targeting values achieve target acquisition	72 Hours	SW to SW Interactions	Space + Ground
LYF Payload Test #2a	Payload Operations – Command Processing	Validate stored command sequence achieves activity plan and appropriate wideband (WB) data output	72 Hours	SW to HW Interactions	Space Vehicle {Payload + Bus}
LYF Payload Test #2b	Payload Operations – Command Processing	Validate that targeting priorities are properly communicated and executed	72 Hours	Element Interactions	Tasking + Space + Ground
LYF Payload Test #3a	Payload Operations – mission data processing	Validate that WB PL data properly turned into user products	72 Hours	SW Processes & People	Space + Ground + Users
LYF Payload Test #3b	Payload Operations – mission data processing & distribution	Validate timing of mission information chain, including initial data receipt, processing and successful transmission to Users	72 Hours	Processes & People; HW and SW Interactions	Ground + Users

Those aspects that are not flight-like become the specific **LYF test exceptions**. Legitimate LYF test exceptions can arise from physics, engineering constraints, and programmatic decisions. This approach prevents the potential to overlook important deviations from the mission (i.e., timing and transitions).

LYF test exceptions are the specific test deviations from mission characteristics that apply to the mission activity under operationally realistic test. This phrase is used to convey the necessity for linking the exception to a specific flight-like or operationally realistic test. With mission-execution context, it is possible to explicitly identify all critical mission characteristics and distinguish what is and is not flight-like for testing. The significance of a LYF test exception is better characterized with the MCFA results. An example of a high-level list of LYF test exceptions for payload operations-focused candidate LYF tests is shown below (Table 11). Later in the test design process when test resources are allocated and test procedures defined, it is more evident what the associated LYF test exceptions will be.

Table 11. High-Level LYF Test Exceptions for Candidate LYF Tests



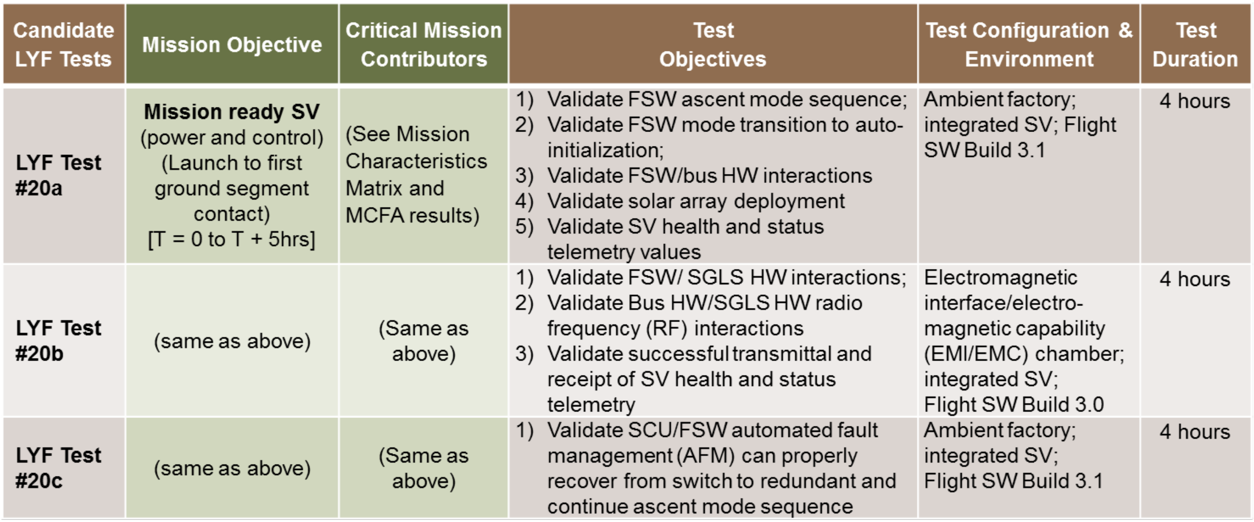
Candidate Test	Critical Event	LYF Test Objective	Test Duration	Mission Contributors	Integration Level	LYF Text Exceptions
LYF Payload Test #1a	Payload Operations – Data Collection	Validate payload (PL) output appropriate for input scenes	72 Hours	Payload and Bus HW & SW Components	Payload	Simulated Bus; Simulated Interface (I/F) between Bus and Payload
LYF Payload Test #1b	Payload Operations – Data Collection	Validate payload gimbal operations do not interfere with vibration sensitive equipment	72 Hours	Payload and Bus HW to HW Interactions	Space Vehicle {Payload + Bus}	Ambient environment (Test Facility)
LYF Payload Test #1c	Payload Operations – Data Collection	Validate uploaded targeting values achieve target acquisition	72 Hours	SW to SW Interactions	Space + Ground	Simulated Space and Ground I/F; Simulated SV location
LYF Payload Test #2a	Payload Operations – Command Processing	Validate stored command sequence achieves activity plan and appropriate wideband (WB) data output	72 Hours	SW to HW Interactions	Space Vehicle {Payload + Bus}	Simulated Space and Ground I/F; Simulated WB data; Test Facility
LYF Payload Test #2b	Payload Operations – Command Processing	Validate that targeting priorities are properly communicated and executed	72 Hours	Element Interactions	Tasking + Space + Ground	Simulated Space and Ground I/F; Simulated Ground Station

4.4.1.3 Assess Candidate LYF Tests

Once a candidate test list has been created, it is necessary to assess it for testability. Testability accounts for the three pyramids (i.e., mission-level, integration-level, and supplier-level), LYF test exceptions, and MCFA results. The concept of *testability* is key to building LYF tests. It provides a mechanism for determining the practical extent to which the literal (operational-like) mission can be replicated in test. Testability allows one to account for the physical and engineering limitations and balance what can be done in a flight-like manner with acceptable and understood risk and program constraints. The **testability** assessment is based on four factors: feasibility, practicality, perceptivity, and value added (i.e., programmatic value).

Consider a set of candidate LYF tests that are intended to address the SV-perspective ascent phase objectives. Pulling from the critical events list (Table 8) and associated matrix (Table 9) the following notional set of tests and test configurations are identified below in Table 12. Each test has a slightly different objective and associated test resource.

Table 12. SV Ascent Phase Mission Candidate LYF Tests



Candidate LYF Tests	Mission Objective	Critical Mission Contributors	Test Objectives	Test Configuration & Environment	Test Duration
LYF Test #20a	**Mission ready SV** (power and control) (Launch to first ground segment contact) [T = 0 to T + 5hrs]	(See Mission Characteristics Matrix and MCFA results)	1) Validate FSW ascent mode sequence; 2) Validate FSW mode transition to auto-initialization; 3) Validate FSW/bus HW interactions 4) Validate solar array deployment 5) Validate SV health and status telemetry values	Ambient factory; integrated SV; Flight SW Build 3.1	4 hours
LYF Test #20b	(same as above)	(Same as above)	1) Validate FSW/ SGLS HW interactions; 2) Validate Bus HW/SGLS HW radio frequency (RF) interactions 3) Validate successful transmittal and receipt of SV health and status telemetry	Electromagnetic interface/electro-magnetic capability (EMI/EMC) chamber; integrated SV; Flight SW Build 3.0	4 hours
LYF Test #20c	(same as above)	(Same as above)	1) Validate SCU/FSW automated fault management (AFM) can properly recover from switch to redundant and continue ascent mode sequence	Ambient factory; integrated SV; Flight SW Build 3.1	4 hours

LYF test candidates are considered for testability at lower levels of the integration, mission, and supplier pyramids as necessary for risk reduction, practicality, or better perceptivity. All assessments must be in context of mission objectives and associated applicable mission characteristics. As a result, the candidate LYF test list is refined into a set of realizable tests that the program can execute.

- **Feasibility**: A feasible test is one that is possible within the limitations of the physics. Ground testing of a space system always has limitations in creating all the concurrent physical and operational environments the system will experience during the mission. This is not to say that all tests must incorporate all such concurrent conditions. The assessment process is used to help identify those characteristics that prevent the execution of a candidate LYF test. Test objectives can be adjusted to be in concert with a feasible set of mission characteristics.

  A test is *feasible* if the physics allows it to be performed in the proposed test venue regardless of resource limitations.

- **Practicality**: Practicality implies resource limitations such as facility availability (or whether such a facility exist), risks to the vehicle or operations, confusion about test due to interactions between the test equipment and the items being tested, and necessary special equipment that may be difficult to design, obtain, or use. For SV safety reasons, there may be situations in which it is infeasible or too risky to induce the fault on the vehicle or to utilize a SV for ground segment testing. In these cases, a testbed with flight processors running mature flight SW with simulated subsystems may be the choice for LYF testing. The limitations of this arrangement (i.e., which critical mission characteristics are present and which ones are not) must be clearly understood and appropriate validation of the testbed conducted. Use of a testbed in lieu of the flight item would be noted as a LYF test exception, and also evaluated from a risk perspective.

  A test is *practical* if the resources are available and the engineering of the test configuration is such that the test components will not obscure or obfuscate the test results of the items under test.

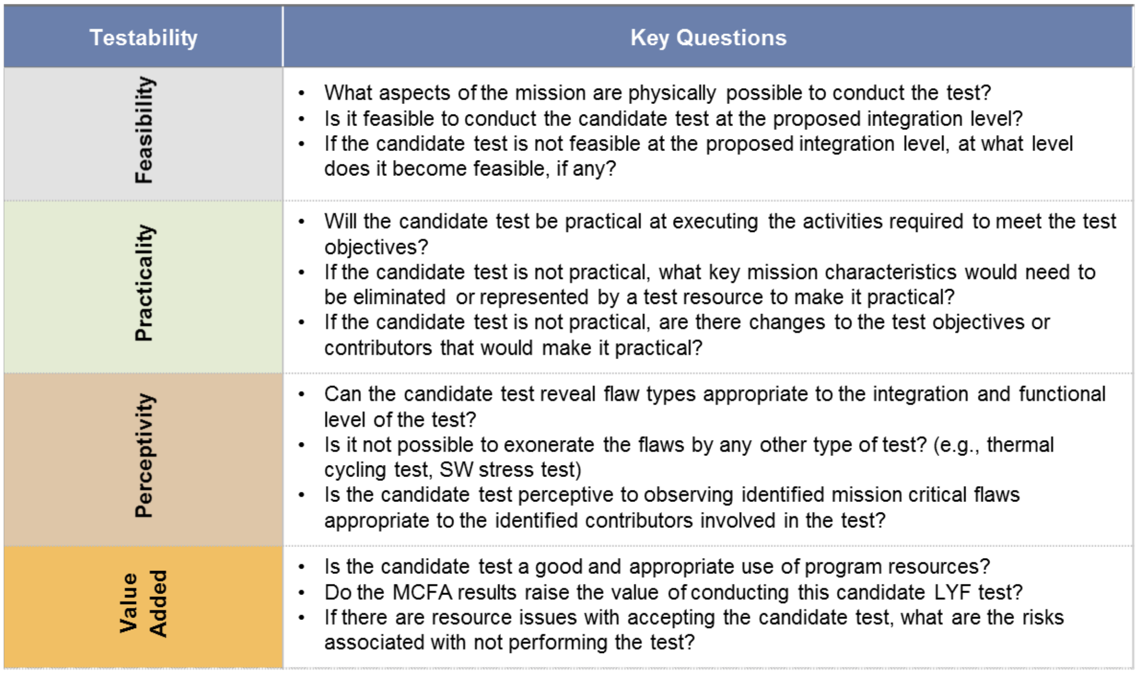
- **Perceptivity**: The perceptivity assessment must address the test's ability to validate the item's use in the mission.

A test is *perceptive* if it is sensitive to the parameters being measured for the test success criteria, or is sensitive to particular flaws being explored.

- **Value added**: Practicality and perceptivity are combined to form the basis for evaluating the programmatic value of the test. In terms of mission-critical flaws detected, the benefit relative to the cost and risk of conducting the test informs decisionmakers as to whether the LYF test (including identified integration and mission-level) should be conducted.

  The programmatic *value* of a test is evaluated in terms of its required resource needs and constraints (e.g., money, time, personnel, and equipment) versus the risks of not allocating those resources for a test.

The testability assessment is performed by asking a series of questions about each candidate test. A sample of the types of questions addressed is provided in Figure 18.

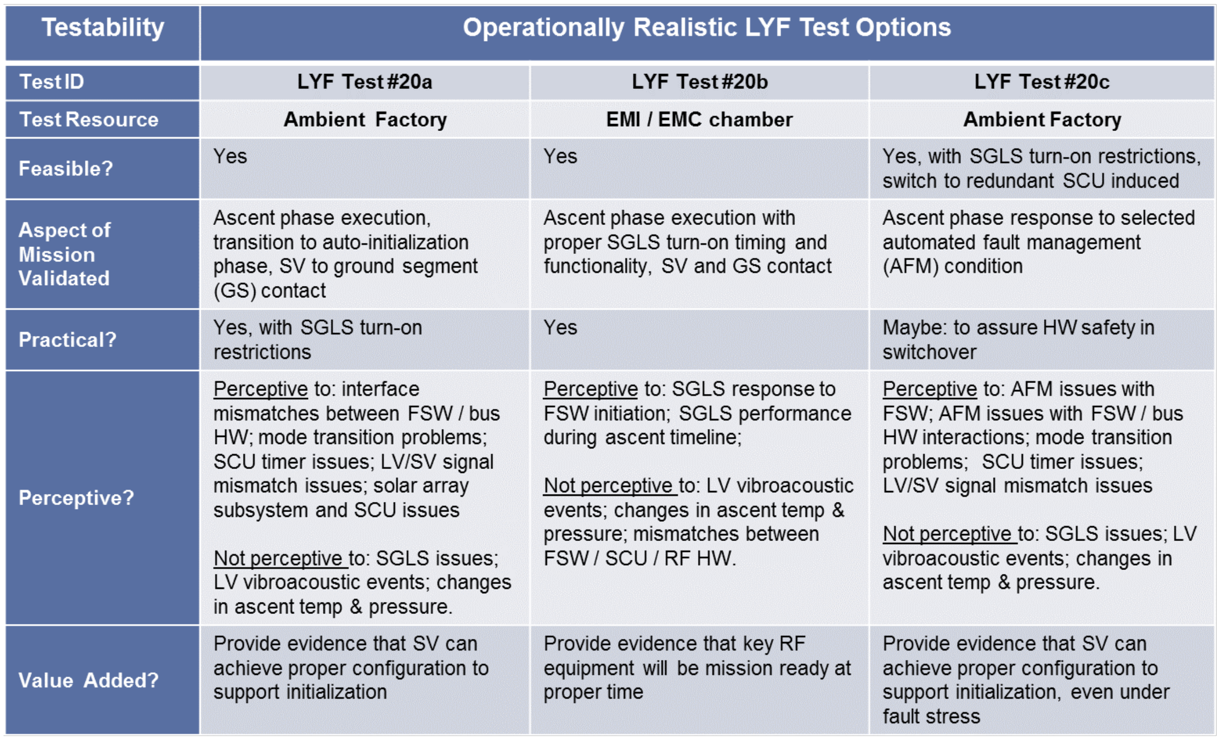


Testability	Key Questions
Feasibility	• What aspects of the mission are physically possible to conduct the test? • Is it feasible to conduct the candidate test at the proposed integration level? • If the candidate test is not feasible at the proposed integration level, at what level does it become feasible, if any?
Practicality	• Will the candidate test be practical at executing the activities required to meet the test objectives? • If the candidate test is not practical, what key mission characteristics would need to be eliminated or represented by a test resource to make it practical? • If the candidate test is not practical, are there changes to the test objectives or contributors that would make it practical?
Perceptivity	• Can the candidate test reveal flaw types appropriate to the integration and functional level of the test? • Is it not possible to exonerate the flaws by any other type of test? (e.g., thermal cycling test, SW stress test) • Is the candidate test perceptive to observing identified mission critical flaws appropriate to the identified contributors involved in the test?
Value Added	• Is the candidate test a good and appropriate use of program resources? • Do the MCFA results raise the value of conducting this candidate LYF test? • If there are resource issues with accepting the candidate test, what are the risks associated with not performing the test?

Figure 18. Sample of key questions for testability.

Table 13 shows an example of what that exercise may look like using the specific ascent-phase candidate LYF tests identified in Table 12. The options for consideration revolve around test resources which often have cost and schedule implications.

Table 13. Sample of Testability Assessment for Candidate LYF Tests (SV Ascent Phase)



Testability	Operationally Realistic LYF Test Options		
Test ID	LYF Test #20a	LYF Test #20b	LYF Test #20c
Test Resource	Ambient Factory	EMI / EMC chamber	Ambient Factory
Feasible?	Yes	Yes	Yes, with SGLS turn-on restrictions, switch to redundant SCU induced
Aspect of Mission Validated	Ascent phase execution, transition to auto-initialization phase, SV to ground segment (GS) contact	Ascent phase execution with proper SGLS turn-on timing and functionality, SV and GS contact	Ascent phase response to selected automated fault management (AFM) condition
Practical?	Yes, with SGLS turn-on restrictions	Yes	Maybe: to assure HW safety in switchover
Perceptive?	Perceptive to: interface mismatches between FSW / bus HW; mode transition problems; SCU timer issues; LV/SV signal mismatch issues; solar array subsystem and SCU issues Not perceptive to: SGLS issues; LV vibroacoustic events; changes in ascent temp & pressure.	Perceptive to: SGLS response to FSW initiation; SGLS performance during ascent timeline; Not perceptive to: LV vibroacoustic events; changes in ascent temp & pressure; mismatches between FSW / SCU / RF HW.	Perceptive to: AFM issues with FSW; AFM issues with FSW / bus HW interactions; mode transition problems; SCU timer issues; LV/SV signal mismatch issues Not perceptive to: SGLS issues; LV vibroacoustic events; changes in ascent temp & pressure.
Value Added?	Provide evidence that SV can achieve proper configuration to support initialization	Provide evidence that key RF equipment will be mission ready at proper time	Provide evidence that SV can achieve proper configuration to support initialization, even under fault stress

The following examples illustrate different aspects of this testability assessment.

Suppose the actual radio frequency (RF) environment is critical to performance for a given mission. Including all the contributors to that environment in appropriate facilities may not be possible in conjunction with other flight-like environments (e.g., vacuum, thermal control or decreasing ascent pressure). A LYF test of the normal operations timeline using active transmitters in the RF environment may be adequately perceptive to RF-related flaws that have the potential to seriously cripple the mission if it can run for the relevant duration of the timeline. Where it may not be possible to have all RF contributors/receivers, or have them operated for their mission duty cycle, further exceptions analysis may be necessary.

The ascent portion of a mission involves flight computers, sensors, and RF equipment operating in a rapidly changing vibroacoustics environment induced by launch and separation. So far, the system integrator's point of view has been discussed; however, the system integrator may levy LYF testing requirements on the suppliers. The system integrator must communicate those requirements and make sure suppliers understand the operational use for what they are building in order to account for those conditions in test. At lower levels of integration, the testability assessment trades fidelity to mission characteristics with practicality, leading to the identification of a selected subset of key parameters that best exercises the item for use in the mission.

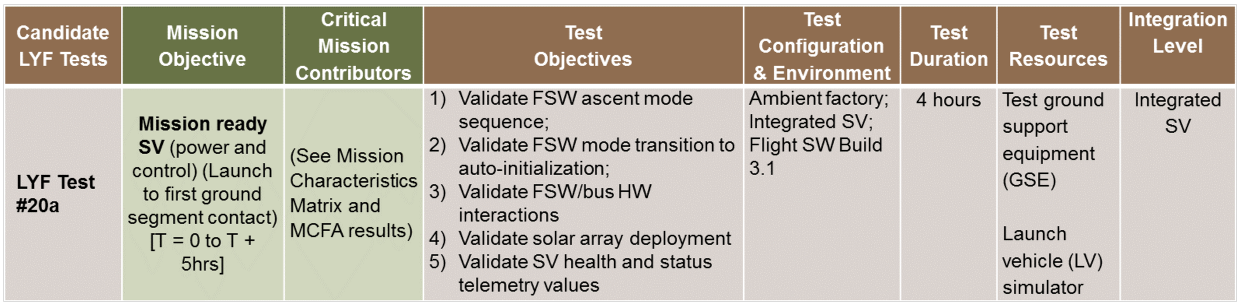
Consider the ultimate preflight execution of a LV mission (see Table 8). It would involve all subsystems executing according to the ascent timeline. A rocket stage, fully fueled, tied down on a test stand for a "hot fire" test is not only practical, but has been done. What is probably not feasible is to perform a hot fire test in a chamber that dynamically adjusts temperature and pressure to emulate those environments over the stage timeline. For a single-stage rocket, all that would be needed to be a more practical LYF test would be the addition of a payload and fairing for demonstrating separation events. It would probably be feasible, but not practical, to actually eject the fairing and payload.

Even at the first high-level pass-through in identifying candidate tests, it will be obvious that some objectives, elements, aspects, or key mission characteristics will not be able to be included in a test. These will constitute the initial LYF test exceptions. However, LYF test exceptions are only meaningful when tied to specific LYF tests. This will be highlighted later in the process.

The output of this step is a prioritized set of feasible, practical, perceptive, and programmatically valuable candidate tests. The priorities established initially due to mission criticality will be adjusted for these testability factors. The output will also include the list of critical and first-time events that have not been assigned to a test at an appropriate level of integration.

4.4.1.4 Allocate Candidate LYF Tests

The allocate LYF tests apportions tests to specific pyramid levels (integration, mission-level, and supplier-level) and test resources. Using candidate LYF Test #20a from Table 10 (shown in Figure 19) as an example helps show the allocation actions.



Candidate LYF Tests	Mission Objective	Critical Mission Contributors	Test Objectives	Test Configuration & Environment	Test Duration	Test Resources	Integration Level
LYF Test #20a	**Mission ready SV** (power and control) (Launch to first ground segment contact) [T = 0 to T + 5hrs]	(See Mission Characteristics Matrix and MCFA results)	1) Validate FSW ascent mode sequence; 2) Validate FSW mode transition to auto-initialization; 3) Validate FSW/bus HW interactions 4) Validate solar array deployment 5) Validate SV health and status telemetry values	Ambient factory; Integrated SV; Flight SW Build 3.1	4 hours	Test ground support equipment (GSE) Launch vehicle (LV) simulator	Integrated SV

Figure 19. Candidate LYF tests and allocations: LYF Test #20a.

All three pyramids are used simultaneously. The objective for the test makes sense for executing it at the SV level of the integration pyramid. Figure 20 depicts the candidate LYF Test #20a allocations. With the payload integrated onto the bus (integration level), there is ample access to the items of interest. The test follows a mission scenario (mission level) for payload operations with a specific focus on the movement of the payload gimbal. The interactions of the gimbal (HW to HW) would replicate those used for data collections on orbit. The payload integrators (supplier level) would be interested in this test to make sure the payload is correctly incorporated. Lastly, it is important to note the test resources that were used to make measurements and drive gimbal movement.

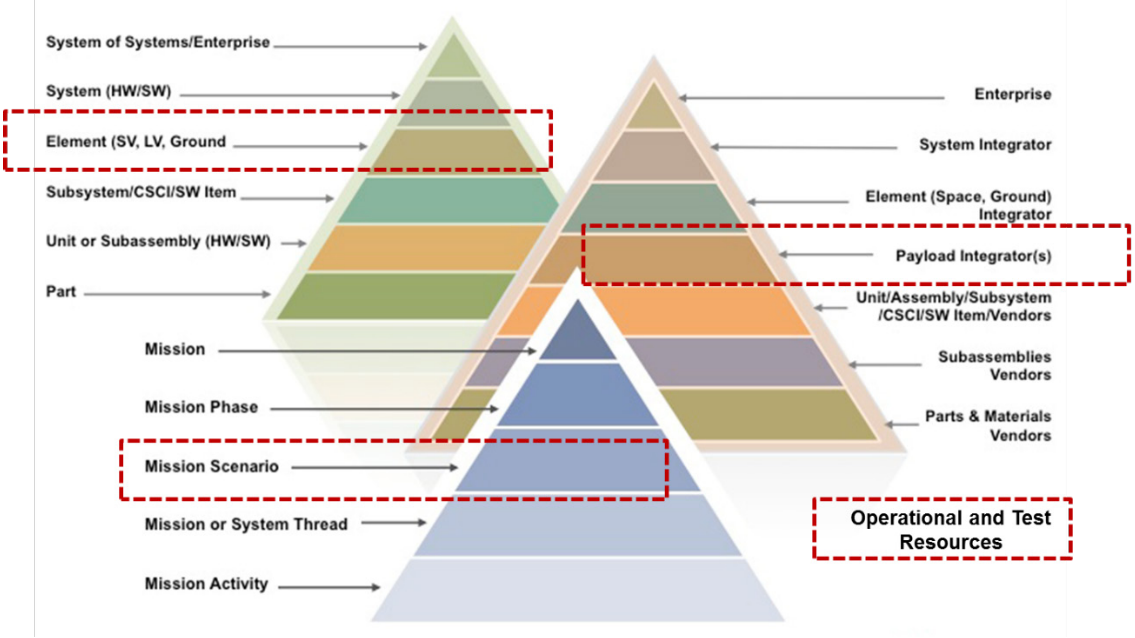


Figure 20. Allocate candidate tests across the pyramids.

LYF testing at lower levels of the integration and mission pyramids can reduce risk by catching problems early when the costs of remediation are lower. Even if the final mission validation will be provided with a high-level test, the program should evaluate the cost/benefit of operationally realistic LYF testing at lower levels or earlier stages of development.

A LYF test at higher levels of integration should include as many critical events in sequence to cover a mission phase or several mission phases, including transitions. A LYF test at lower levels of integration should execute a portion of a mission appropriate to the item and activity under test. A LYF test for ground systems should examine multiple streams of activities within a given phase because activities are often performed in parallel to achieve mission success.

At the lower levels of the supplier pyramid, the government may not have direct contractual relationships with the suppliers. That is where early planning can assist in making sure the test requirements flow down from prime or integrating contractors to lower-level supplier accounts for the mission.

There are reasons to do operationally realistic LYF tests at lower levels of integration. First, as powerful as the end-to-end configuration is at revealing system interaction flaws, there will always be mission-critical and first-time events that cannot be adequately or safely performed at this level of integration. Second, there are flaws that cannot be perceived at higher levels of integration. The third reason is to find flaws that are embedded at the lower levels as early as possible in development, when mitigation is easiest and less costly.

One successful TLYF risk-reduction approach has been to integrate the space and ground segments very early in development, and use the ground system as the command and control test support equipment during payload and SV integration and test. Although both the SV and ground system may have only partial functionality when the integration initially occurs, this approach provides an early indicator of interaction problems. Once those initial problems are solved, this approach provides early validation that the ground system can fully command the vehicle and process telemetry. Particular mission threads and scenarios can be executed as soon as development permits. This process enables discrepancies between the ground and space segments and ground and external systems to be discovered and fixed much earlier than in the more traditional approach where the space and ground segments do not see each other until both are nearly complete.

It may be possible to combine operationally realistic LYF tests with other tests to effectively accomplish multiple objectives. For instance, a SV thermal vacuum test could be extended to be more LYF by sending operationally realistic payload command sequences as they would be generated by ground operators.

The test item(s) and operational timeline to be used for the test should be clearly defined. For instance, in the case of a LYF test of the normal operations of an optical sensor, the test item might include not only the SV and ground command system, but also mission data processing and mission tasking. The operational timeline would include all the activities that would occur during a continuous, several-day period of operations. These activities might include:

- Routine SV activities (e.g., command schedule uploads, state of health checks, stored state of health downloads)
- Routine ground activities (e.g., maintenance, crew handover, system test and exercises)
- Mission operations based on long-term tasking (e.g., long-term schedule uploads, mission data downloads, mission data processing and dissemination)
- Interleaved periods of high-priority, quick response tasking

Example: Solar Array Deployment

Some activities done during a particular mission phase may not be practical or safe to do fully in the context of the mission timeline. The following shows the considerations involved in allocating mission activities to a practical integration level.

A common activity done after SV separation from the booster is an autonomous deployment of its solar array, orienting the array to the sun, and establishing a positive power condition. The highest level of integration for this first-time activity is the SV level. There may be critical flaws that only manifest under certain thermal conditions. Some flaws will only manifest when all components have been installed, including thermal blankets. Some flaws will only be observable in a vacuum. Missions have been lost by solar array problems caused by unexpected thermal gradients driven by the combination of vacuum and solar radiation, generally not used in U.S. test facilities. However, for most deployable solar arrays, it would not be feasible to execute a deployment within the confines of a thermal vacuum chamber. It should be possible (practical) to deploy installed solar arrays in an ambient environment, decoupled from vacuum. It may be prudent to allocate LYF risk-reduction tests to the subassembly (solar array) level and to the subsystem (power distribution and control, including flight SW) level. If the solar array uses a new technology or material for the cells or panels, it may be necessary to allocate a test of a part or subassembly to a combined environment (thermal, vacuum, radiation) test.

An equivalent thought process is used for allocating LYF tests to the mission activity pyramid. If the deployment of a solar array is part of an autonomous initialization phase, then a LYF test should be allocated to a mission phase test. It may be prudent to do a mission timeline LYF test that only includes the series of activities (e.g., solar array deployment through orienting the array to the sun ending with verification of power distribution from the arrays to the power subsystem). Lower-level risk-reduction tests may be allocated to an array deployment scenario, a scenario for orienting the array (and SV) to the sun, and a thread to follow incident energy on the solar array through to the power distribution system.

How are the mission-level pyramid allocations combined with the integration-level pyramid? It may not be feasible to execute that entire mission phase with the actual vehicle. It may be feasible to do a mission timeline LYF test that only includes a limited series of activities (e.g., solar array deployment through orienting the array to the sun). If that is not feasible, or if it is prudent to perform a lower-level risk-reduction test, a deployment thread (command through first physical motion of the solar array assembly) could be planned.

There are key characteristics from the mission that are not available in a factory environment and activities that are neither feasible nor practical on the flight equipment at any level of integration. This is a primary consideration necessary to utilize and allocate candidate LYF tests to test resources. These departures from mission characteristics and the use of non-flight items must all be treated as LYF test exceptions. Potential flaws that will be missed because of these departures are then the impetus to determine alternative perceptive methods to probe for the existence or absence of those flaws. For instance, it may not be feasible to completely deploy large solar arrays in test as they would be deployed during flight. The MCFA might include flaws concerning critical clearances, broken items, interference from cabling or thermal blankets, etc. All such potential flaws should be addressed in the risk portion of the TLYF process. It may be that some of these flaws may be amendable to assessment by models or tests on smaller portions of the array.

At some point in a test program of an autonomous system, certain contingencies should be exercised. In the case of the solar array deployment example, a failure to deploy can be a mission-ending situation that may be recovered by quick and appropriate action by the ground control team. A solar array deployment failure contingency test will need to be allocated to a combination of actual HW, testbeds, and simulators. The allocation also has to consider whether it will be done with the actual ground control equipment, processes, and personnel. There might need to be a risk-reduction test allocated to the ground segment and operations personnel.

Example: Ground System

Consider the case for testing a ground system's ability to receive, process, and transmit payload sensor data. The operational path would begin with "photons in" through a number of intermediary elements and end with "messages out" to the users/customers. Given factory and operations constraints, it may be impractical to stimulate the sensor in a mission-adequate manner that would ultimately result in a representative set of values in the products at the end of the operations chain. A common decision is to generate representative payload data in its own test venue; repackage that data, if necessary, as it would be generated through spacecraft systems; and provide that simulated data stream to the test or operational ground system. A ground system element or data processing thread test would be a reasonable alternative at lower levels of physical and mission integration to validate the ground system's ability to convert an input stream of mission data into viable products for users. Each of these alternatives will have its own set of LYF test exceptions, including those involved in the generation of the input data stream.

These tests are formally documented as appropriate to the stage of program development in the RFP, the TEMP, vendor contracts, and/or test plans. Each allocated test will include the test objectives, a list of the resources required (operational and test) to execute the test, some key aspects of the test, worst-impact LYF test exceptions or reference to more extensive list of exceptions, and test pass/fail criteria. Any implied schedule constraints, for instance by using flight HW or SW, operations procedures, etc., should be noted as well.

A second output of this step is a list of unallocated tests. The reasons for not accepting and allocating these tests should be documented. Risks associated with not performing each unallocated test should also be captured. The unallocated test list should be periodically reviewed as part of the *Perform Critical Fault*

Risk Management step (discussed later in this report) to determine if any relevant conditions have changed to warrant a change in allocation for a particular test.

4.4.1.5 Mission Readiness Test

It is now possible to define high-level mission readiness tests (MRTs). MRTs are operationally realistic tests that demonstrate the system's ability to conduct its mission by accounting for all elements and aspects of the operational chain before flight. They are typically SV-centric with the spacecraft in the factory run by the ground system. They are executed according to a mission-equivalent timeline using as many elements of the operational chain as is feasible. They include information and data flows, as well as transactions between interfaces in the operational chain. They are most effective at uncovering flaws when conducted over a period of time (i.e., days to weeks) and as close to launch/system activation with operational flight and ground configurations (i.e., HW, SW, and interfaces) as can be reasonably achieved to exercise the complete operations chain and cover critical events and transitions.

The MRT might be referred to as an end-to-end days-in-the-life (DITL) test. The use of the word "days" and not "day" is intentional. This highlights the value of executing an MRT over a period of days (as with the Mission Demonstration example used earlier). Also, the distinction here is that end-to-end tests can be done in ways that do not emulate the mission (i.e., functional or compatibility tests). The MRT is not just about the ends, but about other mission characteristics that are necessary to accomplish the mission in that configuration.

A notional concept of what would be involved in an MRT for a remote sensing mission is shown in Figure 21. The end-to-end chain begins with inputs from payload and spacecraft stimuli called "photons in." Some missions are taskable by the customer and will have inputs from tasking organizations. Outputs from the chain will include either mission products (e.g., images and data) or mission services (e.g., communications, navigation) called "messages out." The MRT goes beyond functional and interface testing by focusing on the specific command formats, data and message content processing.

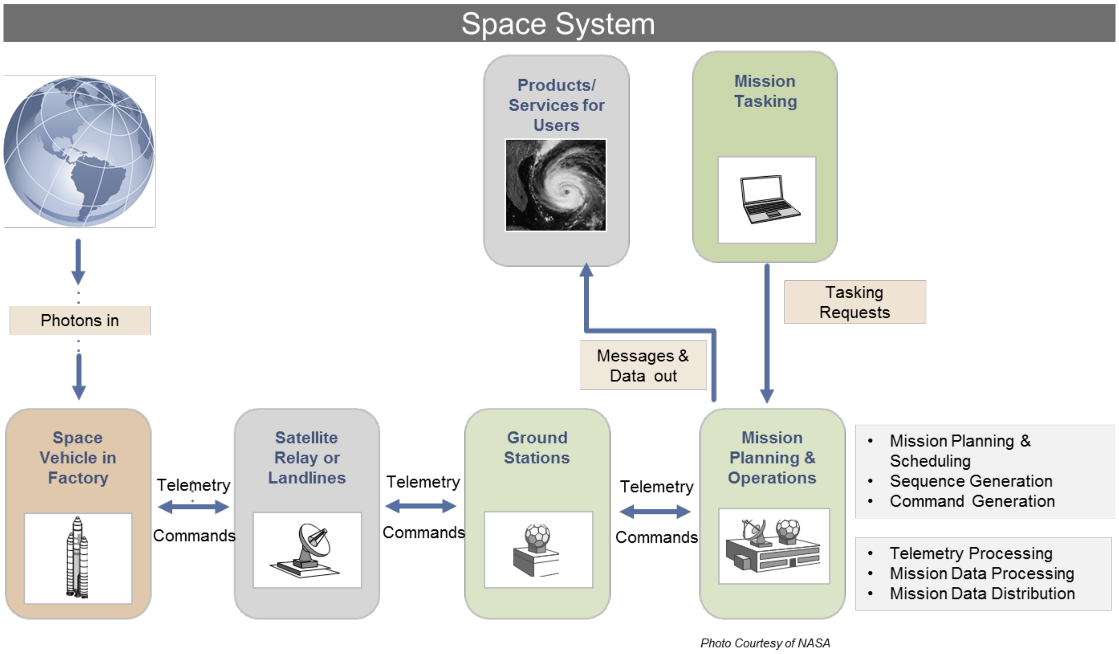


Figure 21. Mission readiness test (space system example).

Figure 22, Figure 23, and Figure 24 provide other examples of MRTs for ground, launch, and payload systems, respectively. If all elements are accounted for and mission ground assets are in control, this test is referred to as a mission readiness test. If only a subset of the system is represented, this type of test is an operationally realistic LYF test.

A ground system MRT (Figure 22) could represent the mission case where the primary mission control center goes offline and the backup center is required to ensure continuous operations support. When possible, the test should be performed on the integrated HW and SW installed in an operational system with connections to actual interfaces. Also when possible, these tests should be conducted at the target sites with operational personnel. The tests should use actual mission-operations configurations and conditions (to the maximum practical extent). The tests should account for the full operational configuration, including the space constellation, with operational timelines and data loads. External systems that could affect the operations of the system under test should be operated or simulated to replicate conditions expected during operations. The test should include focus on external interfaces, the use of operational databases, operational scenarios, and system requirements from a mission operations perspective.

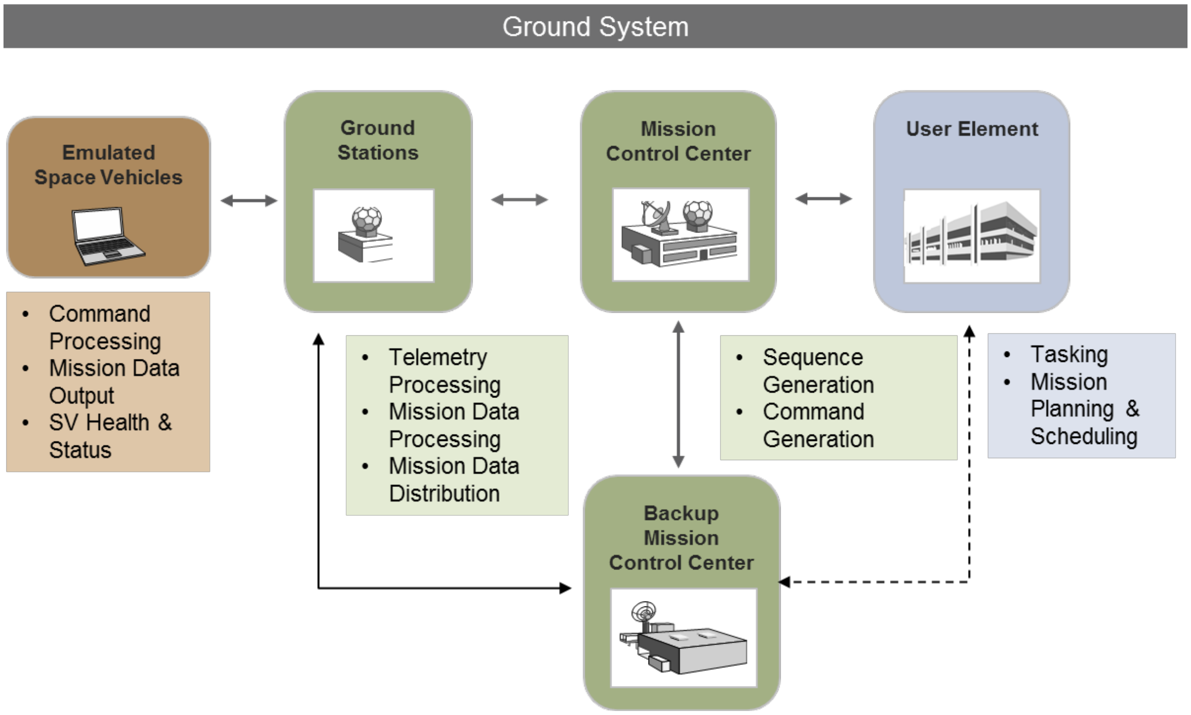


Figure 22. Mission readiness test (ground system example).

A launch system MRT (Figure 23) should exercise mission-specific items, including unique mission profiles, final databases, and SW. This test should also include general launch system processes, including communications and range interactions. It may be that a launch system MRT will need to be done in several venues, as the ascent profile may need to be done in a simulation environment.

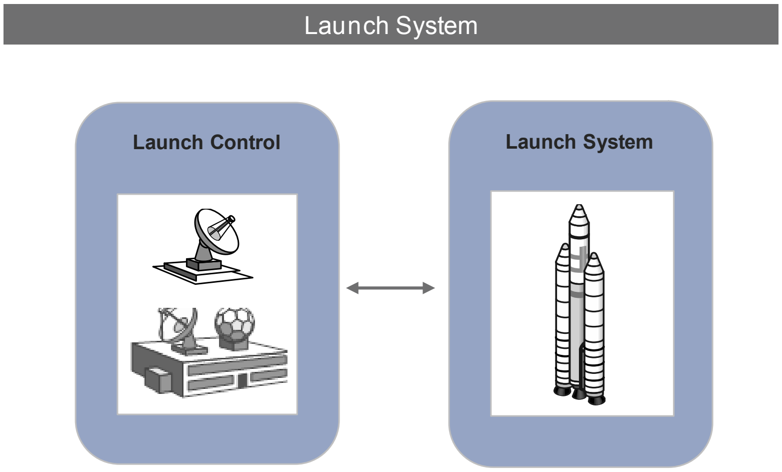


Figure 23. Mission readiness test (launch system example).

Lastly, a payload systems MRT (Figure 24) should demonstrate the payload's ability to provide either data or services for an end user. The flight HW and SW for the payload are in the factory and is being stimulated by source data. The payload unique commands are generated and sent to the SV for processing. The spacecraft bus supporting payload operation is emulated, and the payload data is sent to an emulated ground system. The key interfaces are examined and the final data product evaluated.

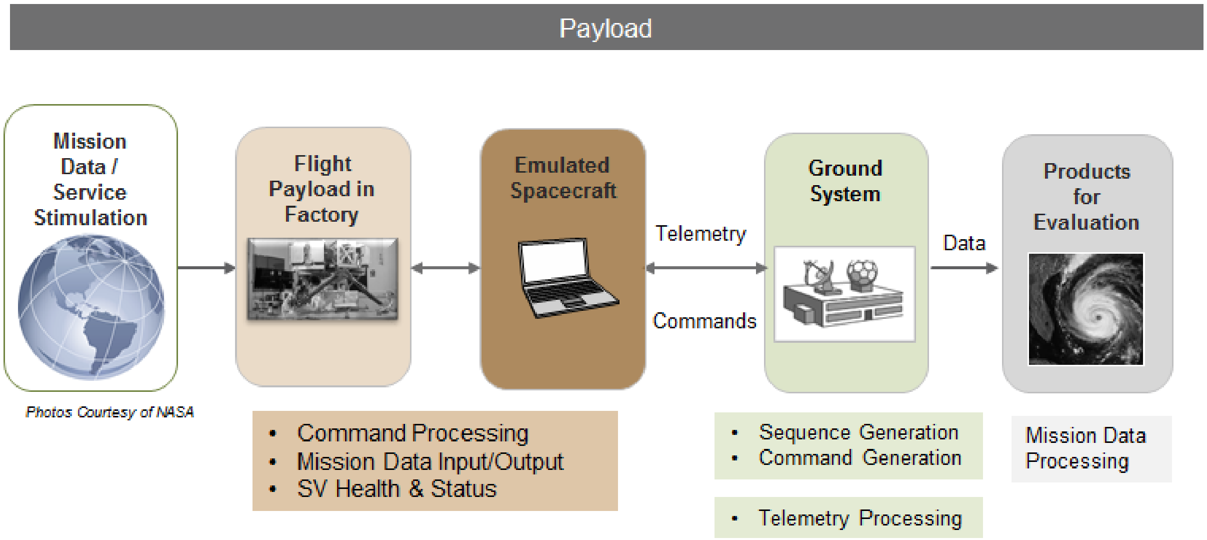


Figure 24. Mission readiness test (payload example).

For the payload MRT example shown in Figure 24, it is likely that a scene simulator will not be able to fully replicate all the various targets. Shortfalls in that replication would be noted as LYF test exceptions. The use of a payload internal stimulation source will be different from the ultimate targets to be observed. Those differences are noted as LYF test exceptions. All operationally realistic LYF tests will have deviations from the mission. Those deviations may be benign or critical. Each LYF test exception will ultimately be evaluated for its place on that scale between benign and critical. The results from the Perform MCFA step will help in the evaluation. LYF test exceptions will be discussed in more detail later in this report.

The MRT should reflect key aspects of the mission or service for each mission phase/key event, as defined in the CONOPS or equivalent source material. Key events for ground systems may include—but are not limited to—test, exercise, rehearsal, maintenance, upgrade, sustainment, transition, handover, normal operations, and contingency operations. For space systems, it also includes the SV in the factory under control of ground station assets being flown in the flight-like manner to the extent feasible. For ground systems, it may include multiple data sources being processed simultaneously across the ground system. The MRT is the bridge between the SV in the factory controlled from the ground station simulator and the flight CONOPS that indicates how the mission will be flown. Organizations requiring an end-to-end test have experienced that this test finds defects that are not detected by any other testing method. This type of test is already required by National Aeronautics and Space Administration (NASA) General Environmental Verification Specification (GEVS) and the European Space Agency (ESA) [9]. It is a natural follow-on to limited functional end-to-end tests, such as radio frequency (RF) compatibility tests.

ESA provides specific guidance for operationally realistic end-to-end MRT testing:

- Two end-to-end tests, at six months and three months prior to launch, must be run with the SV linked to the ground station, thus meeting the "test *what* you fly" dictate.

- Each test must be between one and two weeks in duration on the critical path, using "realistic operations sequences," with much of the test done on a mission timeline.

- The execution and passing of the end-to-end test is a fundamental program requirement to be verified. This verifies the (often unwritten) prime directive that says it has to work.

Depending on time and resources available, an MRT may need to be broken up into manageable pieces (referred to as operationally realistic LYF tests) to adequately represent the entire mission (phases/key events) and their associated distinct end-to-end configurations and time durations.

4.4.2 Outputs

As a result of *Mapping Mission to LYF Tests*, a list of **allocated** LYF tests and their priorities is generated, along with a list of remaining unallocated LYF candidate tests with brief rationale and a list of first-time/mission-critical events with no planned validation.

For each **allocated** LYF test, the following is generated:

- Test objective based on corresponding mission objective
- Allocated pyramid level (supplier, system integration, and mission)
- Allocated resources for test (operational and test resources)
- LYF test exceptions list (high-level)

Allocated LYF test high-level details are also documented (e.g., test plans and contractual documents).

4.5 Architect LYF Tests

Once a list of allocated LYF tests exists, the specific plans within the constraints identified are developed. During this step, key architectural elements for each test are formulated. At this point there is also an opportunity to explore a program's existing test plans, resources, and strategies for opportunities to incorporate or combine LYF tests with other existing tests.

4.5.1 LYF Test Architect

The architect (consisting of a team or a person) is responsible for planning, designing, and overseeing the construction of LYF tests. Early involvement in the TLYF process is key. The architect must be completely familiar with the system, its missions, and associated critical and first-time events. The architect must have cognizance and influence over the purposes, flow, and rhythm of the mission.

After allocated LYF tests have been determined, the architect assumes leadership for assigning roles and responsibilities to involved organizations to create high-level test plans for the flow of operationally realistic LYF tests involving all the participating elements. In some cases it will make sense to include lower-level tests as well.

4.5.1.1 Decision Authority

The architect is the primary decision authority for the framework of each allocated LYF test. The acquisition program office needs to decide which organization is responsible for architecting system and higher-level tests. Usually, an architect comes from the organization requiring the LYF tests. Early involvement in the process and an awareness of the specific system acquisition is helpful.

The architect leads the effort in identifying the set of architectural elements, organizations, and participants for each test. This requires the ability and authority to make priority, resource, and risk decisions concerning what LYF tests will be further developed and executed.

The architect oversees the LYF test development to ensure test design represents the mission execution and critical mission characteristics.

4.5.1.2 Collaborations

The architect works with LYF test designers to ensure the LYF test meets the intent of the test (i.e., test objectives are tied to mission objectives) and that test entry and exit criteria are adequately defined. The architect has the ability to adjust test scope as necessary to respond to test design considerations (e.g., test coverage of applicable mission characteristics), and evaluate the resources needed (operational or test) and LYF test exceptions that result from test design decisions.

4.5.1.3 Involvement in the TLYF Process

The architect contributes to LYF test development during the *Map Mission to LYF Tests* step. The architect also participates in doing mission-critical fault analysis and critical fault risk management. The architect provides input to and documents architectural decisions (at the appropriate level) to bound LYF test design; develops and updates LYF test architecture based on scope and resources agreed upon during the *Map Mission to LYF Tests* step; and identifies trade-offs between the mission objectives, test objectives, the architectural elements, and testability, returning to the *Map Mission to LYF Tests* team as necessary.

The architecting process drives decisions concerning which operational elements to include in the test within the framework of program resources. For each test, the architectural aspects (who, what, where, when, and how) shown in Figure 25 must be defined. Architectural decisions for an operationally realistic LYF test depend on whether it is a standalone test or is being developed to meet several objectives, one of which is to validate mission operability. In the course of architecting LYF tests, associated LYF test exceptions must be addressed—especially those tied to fault paths. These will be evaluated for mission risk and the exceptions deemed critical will be tracked (see section 4.7).

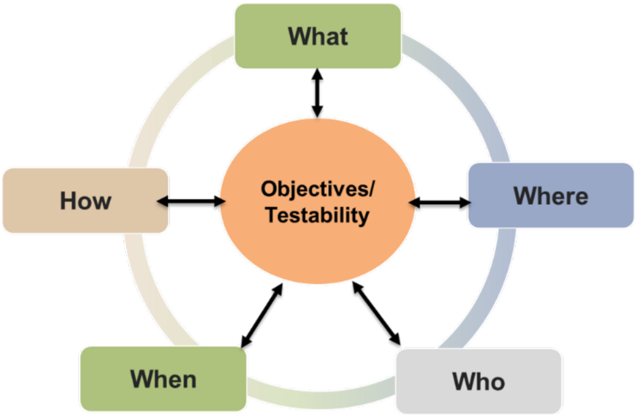


Figure 25. Interdependent architectural aspects.

4.5.1.4 Architectural Decisions

While considering objectives and testability, the architect must address the following architectural aspects of the test:

- **Who** should participate? (e.g., organizations, operational crew, user, test team, acquisition team, system/subsystem subject matter experts, customer)

- **What** should be used in the test? (e.g., test versus operational resources, supporting equipment)

- **Where** should each "what" reside? (e.g., factory, development setting, special test facility)

- **When** must each "item" be available? (e.g., resource availability, program schedule, programmatic constraints)

- **How** can the test event happen? (e.g., connectivity and interdependence of tests in overall test flow, logistics, multiple owners of test assets)

4.5.2 Outputs

The output, documented in the initial test plans, is the set of architectural details for each LYF test that the program will perform. The tests will be incorporated into the development schedule and be consistent with the test program.

Each architected LYF test will include:

- Details from the *Map Mission to LYF Tests* step
- Mission phase/event/situation coverage
- Test plan
- LYF test exceptions
- New candidate LYF tests (if applicable)

It is also possible for new LYF tests to be formed.

4.6 Design LYF Tests

The *Design LYF Tests* step takes an architected test through the test design elements, considerations, trades, and challenges. Unlike other tests, LYF tests are focused on demonstrating the system's ability to successfully execute mission objectives. The test design needs to account for the presence of critical mission characteristics and the opportunity for mission-critical flaw detection. The process must address the specific details of how tests will incorporate mission characteristics and use mission equipment, processes, and procedures as defined in the previous steps. It must also specify the use of test processes, equipment, and procedures to either emulate the mission and/or provide for the safety of the items under test.

4.6.1 Process Details

LYF test design involves defining a test approach and making sure it is successfully implemented. The test design team must create the detailed content for each architected LYF test. The test designer must make decisions about test objectives (i.e., what specifically needs to be accomplished by the test), mission characteristics (i.e., which ones need to be included in the test to satisfy those objectives), and test characteristics (i.e., which ones must be included to make the test executable). The test designer must also understand the fidelity of test characteristics and processes as compared to the operational mission characteristics and processes. Some aspects of fidelity may have been dictated or constrained in earlier steps, but the remaining aspects must be dealt with here.

LYF test design will be constrained or enhanced by decisions relating to test resources (i.e., simulators, stimulators, emulators, testbeds, and other test support equipment). As test procedures are developed and test resources identified, more critical LYF test exceptions will surface. These must be documented for use in the *Perform Critical Fault Risk Management* step.

4.6.1.1 Test Objectives

Test objectives for a LYF test must relate to the mission. This distinguishes a LYF test from other tests. The prime objectives are to provide evidence that the mission-critical activities included in the test can be successfully accomplished as they will occur during the mission and to provide evidence of the presence, or absence of, critical flaws.

An example from the Space Test Experiments Platform (STEP) Mission 1 illustrates how different test objectives influence the test results. A one-orbit design reference timeline scenario, as depicted in Figure 26, had been created early as a basis for time-/orbit-based design considerations. Traditional test design seeks to hold all but one independent variable constant, so that the effects of changing one (independent) variable can clearly be observed. A primary objective of the SV thermal vacuum test is to identify vehicle behavior that changes as a function of temperature in vacuum. To achieve that objective, it is necessary to run the same functional tests, including the reference orbit timeline, at different temperatures. The contractor identified the design reference orbit, from perigee at T=0 through the next perigee at T=102 minutes, as a LYF aspect of their thermal vacuum (TVAC) test. This reference orbit was run many times in TVAC, each time resetting the clock to “0” after each 102-minute simulated orbit.

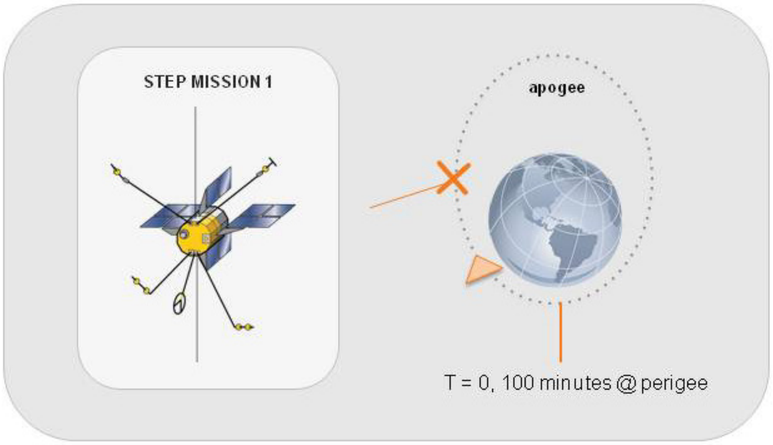


Figure 26. Example—STEP mission 1.

While this is a valid approach, it is different from the objective to find critical flaws during SV on-orbit operations. This second objective requires a longer timeline to probe accumulation or timing errors and other problems associated with a more complete set of mission activities. One key activity not included in the reference orbit was mission-equivalent ground commanding and data retrieval. This program had already done a factory compatibility test (FCT) demonstrating that sample commands of each type could be sent by the ground station and properly received by the vehicle. The FCT also demonstrated that the vehicle could send representative telemetry of each type and have that properly interpreted by the ground system. This test only demonstrated that the ground could command and receive (requirements-centric). It did not successfully demonstrate the system's ability to run the mission (mission-centric).

Original test plans assumed that performing the FCT—not involving any representation of the mission timeline, and exercising the one-orbit reference timeline, which does not involve interactive commanding and payload telemetry—would validate the ability to perform the mission. This conclusion was based on insufficient evidence to show that the system could support normal, sustained operations. Even though this was a "Class C" development (low monetary value, non-operational), the lack of evidence that the system could consistently provide useful primary payload data was compelling enough for program management to authorize an additional 12-hour mission readiness test, using a representative activity timeline performed in vacuum.

This additional test surfaced a mission-critical flaw approximately 20 minutes past the second perigee, as indicated by the arrow in Figure 26. The flaw was in the onboard data management SW, which allowed data from the secondary payloads to overwrite the primary payload's data. The objective for the mission was to obtain the primary payload's data. Undiscovered in test, this flaw could not have been corrected on-orbit, effectively precluding getting results from the primary mission.

4.6.1.2 Mission Characteristics and Cases

The test designer must decide which detailed characteristics to include in the test to reflect the architecture and to support test objectives (see Figure 27). The grouping of mission characteristic classes and associated mission characteristics has been provided above (see Table 4).

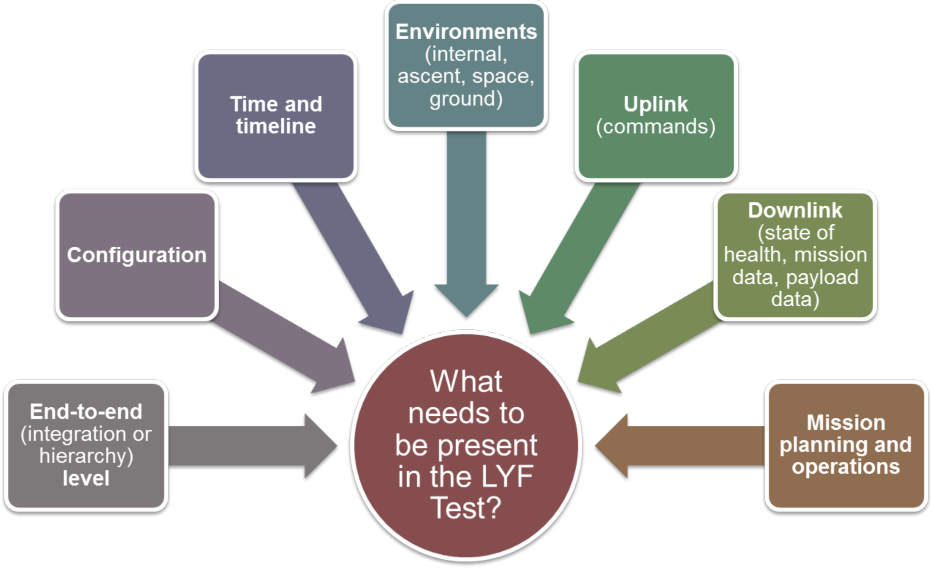

Environments (internal, ascent, space, ground)
Time and timeline
Uplink (commands)
Configuration
Downlink (state of health, mission data, payload data)
End-to-end (integration or hierarchy) level
What needs to be present in the LYF Test?
Mission planning and operations

Figure 27. Mission characteristics and LYF tests.

The mission aspects to be covered in a LYF test may happen in more than one manner during the mission or may have a mix of characteristics. Some may happen in parallel, as in the case for ground operations. The test designer must evaluate the mission situations to be validated and consider which test cases best capture those variants. For LYF tests, the organization of the mission will influence the identification of test cases. The system design will influence the inclusion of other test cases (e.g., interactions between elements, use of primary and redundant elements). For DITL/WITL tests, how many different kinds of days and cycles there are need to be identified. A mission that has a weekly planning cycle, a daily planning cycle, and a priority path for special or emergency operations may give rise to the need for multiple test cases. Test cases should specifically include fault and contingency conditions. Faults or contingencies that lead to the usage of backup or redundant items make effective test cases that can validate the use of those backup/redundant items. The number of test cases will also be influenced by the number or range of conditions which can initiate the activity (initial conditions) and the different transitions into and out of the activity.

For each test case, the test designer must identify the independent and dependent variables where appropriate, and define the test points to be monitored. For MRTs, it is important that there are test points with appropriate perceptivity. For instance, simply checking to see that an output file has been written or received is generally inadequate. The content of the file must be verified for correctness. In too many cases, spacecraft problems are detected late in AI&T. Later investigation shows evidence that the fault was captured in telemetry from an earlier test—but the telemetry was never assessed.

4.6.1.3 Initial, Transition, and End Conditions

Some mission activities may have a highly constrained set of initial conditions and limited alterative execution paths. That may lead to a small number of test cases. It may be feasible and valuable to do them all. Other mission activities will have a relatively wide set of options for initial conditions and execution. These require more test cases, especially if fault conditions and recovery could result in additional execution paths.

A timeline test of the ascent mission phase usually has a fairly constrained initial condition. An obvious test case is to run the system from the initial condition (T+0 or some prelaunch known configuration point) through satellite separation or the collision-avoidance maneuver. For systems that have active fault management in this phase, it would be prudent to inject some fault cases. One obvious case is the fail over from a primary to backup flight computer to validate that the mission can succeed in the case of early redundancy selection. Another case may consist of switching between the primary mission control center (MCC) to a backup control center (BCC).

The reason to include transition conditions as a test case consideration is to avoid the premature conclusion that mission phases or activities are independent and have no influence on the phases or activities that either precede or follow them. A test case that induces a transition to a safe mode may also need to include a recovery from safe mode to ensure no flaws exist in either event.

4.6.1.4 Test Design Roles and Responsibilities

Roles and responsibilities for LYF test design are likely to be distributed across a team representing the disciplines and elements involved. The team will solidify the details developed during the *Architect LYF Tests* step.

In addition to refining the details for architected LYF tests (i.e., objective, mission coverage, pyramid allocation, test resources), the test team is responsible for:

- Identifying parallel events/activities (i.e., ground systems) covered by the test
- Potential flaw paths intended to be exonerated or discovered in the test
- Defining the test success criteria (entrance and exit)
- Identifying test configuration (HW, SW, interfaces)
- Assigning, reviewing, and managing the development of the test procedures
- Dry-running and executing test procedures and capturing any updates made during the test (e.g., as-run test procedures)
- Capturing the deviations from the mission (LYF test exceptions) and process departures notable in the test (learning how the items under test actually operate versus how it was conjectured that they operate)
- Identifying and coordinating availability of any operational assets needed for the test

4.6.1.5 Test Equipment and Resources

The designer for a LYF test must draw on particular kinds of test equipment and resources to create executable tests. Table 14 includes a few types of equipment and tools needed to make a viable LYF test.

Table 14. Test Equipment Terms and Descriptions

Term	Description
Emulator	A system whose main function is the reproduction of a combined HW and SW simulation so as to perform as a surrogate for said system. An emulator simulates HW characteristics. Example: A GPS 1 pulse per second (PPS) emulator would drive a physical pulse signal into another electronics box. A GPS 1 PPS simulator may just write to the appropriate register of SW.
Electrical Ground Support Equipment (EGSE)	Electrical non-flight equipment whose purpose is to support or augment the interface to an item under test. In particular, to provide interfaces, functions, power, or signals required for ground operations/testing prior to flight. This is also referred to as special test equipment (STE).
Engineering Model (EM) HW	A non-flight version of a flight HW unit that utilizes flight design and flight-like components and processes in its manufacturing. This is also referred to as an engineering development unit (EDU).
HW-in-the-Loop (HITL)	A test configuration in which SW and HW are integrated together, including required simulators, to perform a set of dynamics scenarios, often involving state feedback and control.
Integrated Space Vehicle Testbed	This testbed type is a mating of an integrated space vehicle with a dynamics simulator to support closed-loop testing. The integrated space vehicle testbed requires components of the assembly, integration, and test (AI&T) environment
Simulation	The executable implementation of a model hosted on a simulator.
Simulator	A system whose main function is the execution of a set of behaviors that simulate systems or environments not present in the test configuration. Simulators for LYF tests are either HW devices or SW that represent mission elements that are not available in the test environment. Common HW simulators are the use of ground (factory) power in lieu of illuminating a deployed solar array and using representative input signals for launch vehicle inputs. A common SW simulator is used to provide input to guidance, navigation, and control (GN&C) subsystem items to induce the GN&C subsystem to respond as if it were at orbital altitudes. Another common HW or SW simulator is a type used to provide command responses and representative telemetry to ground command-and-control systems for their verification efforts. Simulators are frequently ascribed a fidelity level (high, medium, low), but the industry has no standard criteria for characterizing the fidelity of a simulator.
Stimulator	A stimulator is a device that provides appropriate input to sensors. These may be in the form of photon generators at appropriate frequencies or scene generators for sensors expected to process multi-dimensional scenes. These provide the “photons in” part of the end-to-end setup and may be appropriate both for payload sensors and bus subsystem sensors such as sun sensors or star trackers.
System/Subsystem Testbed (STB)	This testbed is a combination of EMs, flight units of the space vehicle, payload subsystems, and may include a dynamics simulator that simulates other flight subsystems as well as the orbital and attitude dynamics and the environment. The implementation includes all the electrical ground support equipment required to provide subsystem interfaces including a ground console to provide a command and telemetry interface.

The designer must also determine required fidelity levels for each particular test equipment and resource. Therefore, the specific aspects of models, simulators, simulations, and/or testbeds to be utilized must be well understood. In the *Space Vehicle Testbeds and Simulators Taxonomy and Development Guide* [4] three aspects of fidelity are addressed and are included here for reference: interface fidelity, hardware fidelity, and simulation software fidelity.

Interface fidelity levels (see Table 15) describe how closely the interface associated with the simulator or testbed matches the actual flight interface. They vary from physical interfaces over electrical connections between actual hardware components to non-physical interfaces used to support simulator and testbeds. Hardware fidelity levels (see Table 16) describe how closely the simulator or testbed hardware matches the actual flight vehicle hardware. Finally, simulation software model fidelity levels (see Table 17) describe how closely the dynamics simulator on testbed hardware matches the actual flight vehicle [4].

Table 15. Testbeds and Simulators Interface Fidelity Levels

Interface Fidelity Level (0-5)	Summary Description
(0) No Interface capability	No interface is provided between two end items
(1) Software Interfaces	Shared memory or other method to connect to systems/components together using software rather than electrical representation of the interface
(2) Simulated Interfaces	An interface is provided that adequately represents the data and general temporal characteristics of the interface
(3) Non Flight-Like Electrical Interface	A commercial equivalent emulation of the flight electrical interface
(4) Flight-Like Electrical Interface	An equivalent electrical interface, but not using flight qualified parts and cables
(5) Flight Electrical Interface	Actual flight electrical interface exists between end items

Table 16. Testbeds and Simulators Hardware Fidelity Levels

Hardware Fidelity Level (0-5)	Summary Description
(0) No Hardware capability	No hardware capability is provided
(1) Non flight-like Hardware	Commercial hardware capability that has no direct correlation to the flight hardware
(2) Emulated Hardware	Typically a commercial equivalent that has different performance
(3) Flight-Like Hardware	Usually described as an EM present in a testbed. Typically uses non radiation-hardened parts, but has similar performance to the flight hardware present in a testbed or simulator.
(4) Flight Hardware	Flight hardware present in the testbed

Table 17. Testbeds and Simulators Software Model Fidelity Levels

Simulator Software Model Fidelity Level (0-3)	Summary Description
(0) No Simulation Software capability	No simulation software model capability is provided
(1) Simple Model	A static representation (e.g. fixed data) of the output of the end item being modeled
(2) Simple Dynamics Model	A simple dynamics model representing input and output of the end item being modeled. This model's input and output data changes during execution, including simple responses to input parameters.
(3) Dynamics Model	A more complex dynamics model representing all required input and output data of the end item being modeled. Dynamics behavior is modeled (usually with algorithms) with ties to other models and the surrounding environment.

Equipment and resource usage during LYF testing will influence the ability to uncover potential mission-critical flaws. For example, if it is determined that HW and SW interfaces are critical to mission success, it may drive the requirement to include high-fidelity or actual flight/ground interfaces in the total operations chain.

4.6.1.6 Other Considerations

In addition to the elements covered, the following may be considered for LYF test design:

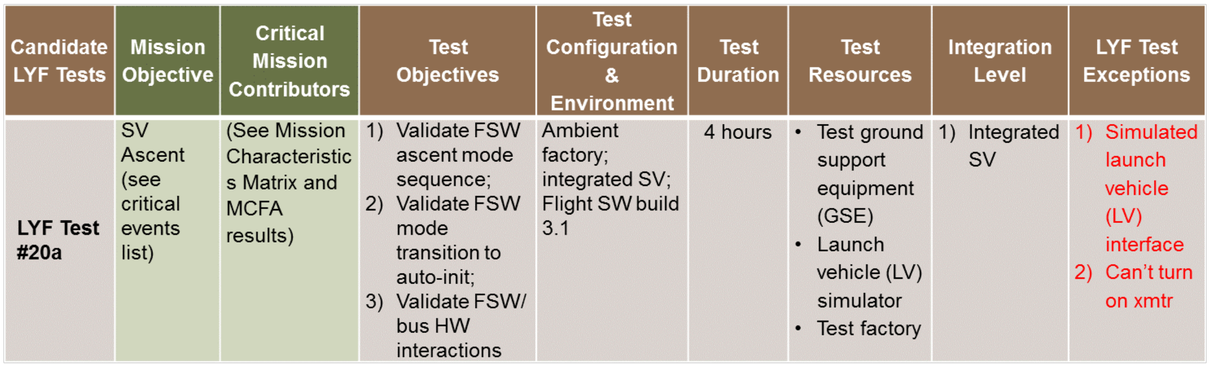
- Test outputs and data collection methods
- Operational tools and processes that will be used
- Test tools, configurations, and processes to be used in planning, executing, and evaluating the test
- Test databases
- Test scripts and sequence of execution
- Test constraints/limitations

4.6.2 "Like You Fly" Test Exceptions

Prior to this step, only high-level LYF test exceptions have been identified. Doing tests exactly LYF/ operate is usually not possible. However, it *is* possible to explicitly identify what is and is not flight-like or operational-like. Legitimate LYF test exceptions can arise from assessing the testability, as discussed in the *Map Mission to LYF Tests* step. Because there are likely to be a number of LYF test exceptions for each mission test, this approach focuses attention on those LYF test exceptions that are tied to potential mission-critical failure situations (fault paths and contributors) and ensures these are mitigated or managed.

Test characteristics can also be the basis for LYF test exceptions. A test characteristic is something that must be in place for a successful test, but is not present during the mission/flight. Factory power used as input to the spacecraft power system is a test characteristic that is not part of the flight. Diagnostic test instrumentation may also introduce distinct characteristics not found during operations. Be aware that the deviations between the test-provided item and the flight method/item can mask flaws.

Using the allocations for LYF candidate test #20a, LYF test exceptions are identified as shown in Figure 28. It is a simple example to show how test resources help detect LYF test exceptions. By expanding the list of critical mission contributors on a worksheet, more LYF test exceptions will be revealed, which will prove helpful for the remaining steps of the TLYF process.



Candidate LYF Tests	Mission Objective	Critical Mission Contributors	Test Objectives	Test Configuration & Environment	Test Duration	Test Resources	Integration Level	LYF Test Exceptions
LYF Test #20a	SV Ascent (see critical events list)	(See Mission Characteristics Matrix and MCFA results)	1) Validate FSW ascent mode sequence; 2) Validate FSW mode transition to auto-init; 3) Validate FSW/ bus HW interactions	Ambient factory; integrated SV; Flight SW build 3.1	4 hours	• Test ground support equipment (GSE) • Launch vehicle (LV) simulator • Test factory	1) Integrated SV	1) Simulated launch vehicle (LV) interface 2) Can't turn on xmtr

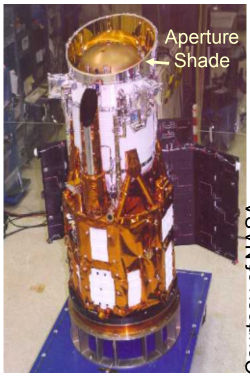
Figure 28. Candidate LYF test allocations and LYF test exceptions: LYF Test #20a.

As an example, the flaw that led to the payload failure on the Wide-Field Infrared Explorer (WIRE) satellite was masked by the use of ground power, rather than spacecraft power.

Lesson: WIRE (Wide-Field Infrared Explorer) Mission Failure

Incident Summary

NASA's WIRE mission failed immediately following its March 4, 1999 launch after a premature ejection of its telescope cover vented all cryogen. The mishap was caused by an unexpected power-on transient from a field-programmable gate array (FPGA), which was not caught on the ground due to a low-fidelity test setup.



Courtesy of NASA

Equipment Summary

WIRE was a 250-kg spacecraft equipped with an IR telescope cooled by solid hydrogen.

Three events had to occur in sequence to fire any initiator: pyro unit power-on, an ARM command to the non-latching arming relays, and a FIRE command to the power field-effect transistor (FET) switches.

Cause of Failure

A start-up transient in the Actel 1020 FPGA ruined the mission. This device has a built-in charge pump that, upon power-on, disconnects all of the device programming circuits from the functional logic cells briefly making the integrated circuit completely undefined and permitting the output pins to reset.

Within 14 milliseconds of the +5 volt application to the WIRE pyro unit, the FPGA closed all arming relays and enabled all output FET switches (Actel 1020's spurious output). Momentarily thereafter, the FPGA stabilized, and the outputs assumed their as-intended state, but the damage was already done.

Cause of Verification Escape

This start-up quirk is transitory. The Actel FPGA will not act uncontrollably again unless it is allowed to sit unpowered for a few hours.

Although power cycled many times during component testing, it was never unpowered long enough to reveal the problem. Even if this transient fire pulse had been noticed, any attempt to reproduce the observation would not have been successful without powering down the pyro unit for hours. In any event, the spurious output pulse was not observed during unit-level testing because a slow-rise power supply was used. During the transient period there was never enough voltage to close the arming relays.

The glitch was not noticed during system testing. The pyro simulator was very sensitive and was therefore fitted with a 23-millisecond load delay to avoid stressing the relay contacts. Because the FPGA's spurious trigger disappeared within 14 milliseconds, the load delay prevented the glitch from drawing current and from being recorded.

Manifestation of Failure

At launch, the chip had been powered down for weeks, making conditions for the transient ripe. Right after orbit insertion, the pyro unit was switched on. The FPGA was supposed to be safed and initialized at the direction of an oscillator clock, a process that took 55 milliseconds. Unfortunately, within 14 milliseconds the transient took place.

By then, sufficient voltage had already built up—during flight, power to the pyro box was applied via a fast relay—to close the arming relay. *The FIRE FETs, commanded by the same controller and therefore not truly independent, set off as well.* The aperture door opened, whereupon cryogen rushed out, causing the spacecraft to tumble and ended the mission.

Management Lessons Learned

- Perform high-fidelity system validation tests for pyrotechnics.

Technical Lessons Learned

- Applicable mission/interface characteristics (timing, initial conditions) must be emulated or evaluated for differences between mission and test items.
- Beware that many programmable devices do not follow their truth tables at power-on—see http://www.klabs.org/ for more information.
- Make sure sequential safety devices operate independently.

There are dozens of mission characteristics that would need to be applied to first-time activities and fault situations. Test design requires noting the deviations from mission as LYF test exceptions. Disparities between test and mission characteristics hold the potential to mask flaws that would not be detected during the LYF test under design. Therefore it is necessary to determine whether or not the exception has a possible connection to any critical fault conditions. The intent of identifying LYF test exceptions is to raise the question: By not including this mission characteristic or its effective approximation, will the test miss the opportunity to detect an associated mission-critical flaw? Although it is difficult to know these conditions prior to test, the *Do MCFA* step provides a mechanism for identifying potential fault paths and contributors. It is most effective to focus on LYF test exceptions that "potentially" contribute to critical fault conditions.

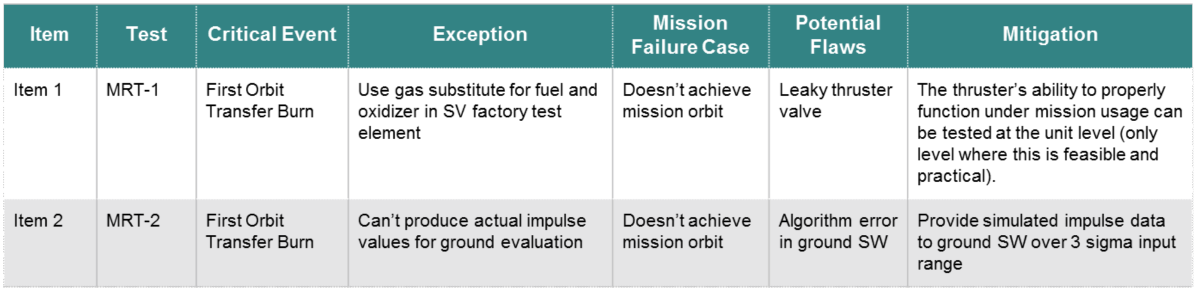
The following should be done for each LYF test:

- Identify deviations from mission characteristics that are critical to mission success based on MCFA*
- Identify what types of flaws could be missed due to this exception
- Define the worst possible outcome of each potential flaw
- Address the possibility that a flaw that will only be revealed in combination with another mission characteristic (or set of characteristics)
- Exonerate exceptions that intersect with critical flaw paths from the MCFA

**If a MCFA has not been done, conduct the analysis in conjunction with test design to determine fault paths and contributors to mission failure*

An example of a productive approach for tracking LYF test exceptions and associated mitigation plans is shown in Table 18. The format should be constructed to ensure that exceptions are addressed in a unique manner, i.e., that the exception is tied to a single activity or flaw path. Each instance of an activity can be different in detail, so it is best to resist any impulse to tie an exception to a wide range of tests or to a class of activities (e.g., orbit raising burns). *Exceptions that cannot be credibly tied to a critical mission event or flaw do not need to be captured.* Many exceptions will have little or no opportunity to mask a fatal flaw. The purpose of accounting for exceptions is to assess the risk of mission loss, and mitigate those that do have the potential to seriously affect the mission. Mitigation here answers the question: "If a potentially fatal flaw cannot be exonerated with a LYF test at this level, what else can be done to determine the presence or absence of the postulated flaw?"

Table 18. LYF Test Exceptions Tracking Example



Item	Test	Critical Event	Exception	Mission Failure Case	Potential Flaws	Mitigation
Item 1	MRT-1	First Orbit Transfer Burn	Use gas substitute for fuel and oxidizer in SV factory test element	Doesn't achieve mission orbit	Leaky thruster valve	The thruster's ability to properly function under mission usage can be tested at the unit level (only level where this is feasible and practical).
Item 2	MRT-2	First Orbit Transfer Burn	Can't produce actual impulse values for ground evaluation	Doesn't achieve mission orbit	Algorithm error in ground SW	Provide simulated impulse data to ground SW over 3 sigma input range

4.6.3 Outputs

LYF tests design results in a set of executable tests based on guidance from the *Architect LYF Test* step. For each designed LYF test, in addition to what has been previously identified, the following information is produced:

Operationally realistic LYF tests (*Each addresses potential mission-critical flaw paths and contributors)*
For each LYF test, the following is generated:

- Test plans (detailed with configurations and resources)
- Test procedures
- Entrance and exit criteria
- Specific LYF test exceptions (i.e., detailed test deviations from mission, including impact from test equipment and resources)

4.7 Execute and Evaluate LYF Tests

Execution of a LYF test is fundamentally the same as the execution of any portion of a space system test. The difference is that the test process itself should emulate the set of mission activities included in the test. When a LYF test is conducted with operational personnel using mission procedures on ground station equipment, with mission-like data routed to users, detected problems will indicate problems with mission execution. In contrast, when more traditional SV and system tests are executed in a factory or ground station setting with AI&T personnel, test SW and test equipment anomalies will occur related to the test resources. These anomalies do not reveal anything about the ability of the SV or system to perform the mission.

Therefore, during execution of a LYF test, comprehensive aspects of the test activity must be captured and evaluated. Deviations from the baseline test plan must not only be noted, but must be evaluated for the introduction of additional LYF test exceptions and for feedback into how the system will be operated during the mission.

Test procedures and associated resources developed and identified in the previous step, *Design LYF Tests*, become the input for execution and evaluation of LYF tests. Also the program's general test and development processes are folded in, such as a test readiness review, discrepancy review process, failure control board, and configuration control process.

4.7.1 Process Details

Because the test item for some LYF tests is more encompassing than for other tests, especially for an MRT, some LYF tests may require a higher degree of coordination among organizations prior to and during test execution. In particular, the process for the identification, handling, and tracking of discrepancies uncovered during test execution must be in place prior to test execution. Failure review boards (FRBs) should have representation from all elements participating in the test. Final authority for discrepancy resolution must be clearly identified. Also, if possible, a TLYF process-minded representative should be present during test procedure development and updates.

4.7.1.1 Execution of LYF Tests

Preliminary execution of a test procedure (dry run) is frequently done to validate its correctness. Dry runs also frequently detect problems with the items under test. A LYF test procedure may be closely related to, or derived from, an operational procedure. Therefore, the temptation to fix the problem and move on with the dry run, rather than using a more formal method of discrepancy handling, should be resisted. Anomalies detected in dry runs must be included in the contractor’s formal failure review and correction process. Failure to do this results in a serious undercount of the anomalies found in the test. Lessons learned from dry running the LYF test procedure should be communicated to the operations and design organizations for proper configuration control. Changes made during testing may also impact operational documentation.

The LYF testing, including dry runs, may be conducted for risk reduction prior to the operational products being brought under formal configuration control. Following a more formal methodology will ensure that all participating organizations are aware of the system’s responses, as well as the proposed solutions—solutions that may affect the way in which the mission is conducted. During formal test execution, traditional testing will have protocols for stopping a test, troubleshooting discrepancies, or continuing the test when such anomalies occur. During a mission, different protocols are likely to be used, as it is not possible to take some paths that are available preflight or prior to going operational. The LYF test team must decide what kinds of discrepancies need to be handled by which protocols.

4.7.1.2 Evaluation of LYF Tests

The evaluation of a LYF test may be made on the basis of inputs, outputs, final products, processes, and performance as it would be during the mission. The evaluation process includes examination of realtime and stored telemetry, data products, and mission services. Anomalies should be identified from the following:

- Procedure execution (i.e., as-run test procedures)
- Instantaneous and trended state-of-health and performance data
- Examination of logs and alarms/warnings
- Examination of mission data or mission service performance
- Mission process problems

For the case of examining mission products, it may be necessary to evaluate the mission service performance to characterize flaws that cannot be detected directly from mission data. That data should be examined for correctness. It is insufficient to simply check the sizes of mission data files, as is frequently done. Data processing steps need to be evaluated for each part of the data flow, from creation to consumption.

LYF tests used to demonstrate readiness to proceed to the next milestone need to be evaluated by those responsible for architecting the test and those responsible for the activities of the next milestone. In addition, LYF tests used for risk reduction (i.e., addressing mission-critical event coverage) need to be evaluated by those involved in the program risk-management process.

Most major tests require high-visibility discussions about the need to perform a retest following potential removal/replacement/revision of HW, SW, or procedures. A LYF test must be examined using the test-what-you-fly principle. Changes to the system as a result of test revelations may result in unintended consequences at the integrated level that cannot be detected in lower-level tests. Changes that fix the first problem may surface a flaw that could not be seen until the first problem was eliminated. These lessons from earlier mission failures should drive retest considerations.

A program must establish a methodology to determine the need for retest and what LYF aspects must be present in the retest, as a result of changes to HW, SW, personnel, processes, or procedures. The methodology must also address and account for the risks in not retesting.

Lesson: Thoroughly Verify All SW Changes, Including Ground SW [15]

A proven launch vehicle was used to launch a SV at a particular time.

Accordingly, the time variable in the SW was changed from *Reference Time* to *Fixed Time*.

Numerous updates to the ground SW had to be made, including one that controlled a valve regulating the ground-supplied nitrogen and, indirectly, an attitude-control engine. This valve should have been closed shortly before liftoff.

Since the *Reference Time* no longer applied, an existing command, “If the state is Abort (or the state is Nominal and *Reference Time* is T-105 sec), close Valve X.” should have been updated to: “If the state is Abort (or the state is Nominal and *Fixed Time* is T-105 sec), close Valve X.”

Unfortunately, the conditional statement in the parenthesis was omitted, and the command became “If the state is Abort, close valve X.” Hence, the valve stayed open, the engine malfunctioned, and the mission failed.

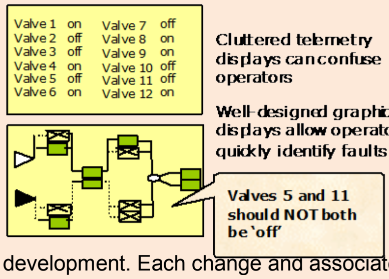
The error went undetected because the change notice included several unrelated items, failed to explain why the control code was changed, and did not compare the “was/is” algorithms. In addition, not all logic paths, displays, and output commands were verified.

Lessons Learned:

- A small SW error can have catastrophic mission impacts.

The failed launch was rehearsed three times, during which the console operators could have spotted the open valve but missed it.

Graphical displays, summarized telemetry data, and error checking should be provided to allow operators to identify and diagnose faults.



Valve 1	on	Valve 7	off
Valve 2	off	Valve 8	on
Valve 3	off	Valve 9	on
Valve 4	on	Valve 10	off
Valve 5	off	Valve 11	off
Valve 6	on	Valve 12	on

Cluttered telemetry displays can confuse operators

Well-designed graphical displays allow operators to quickly identify faults

Valves 5 and 11 should NOT both be ‘off’

- SW change processes require the same degree of rigor as the original development. Each change and associated rationale must be individually approved.
- ***Retest and regression testing should be formal and thorough. All logic paths affected by changes must be verified, and all results must be checked.***
- Operational status, particularly off-nominal indicators, must be displayed effectively.

4.7.2 Outputs

The outputs from this step are the usual test artifacts: as-run test procedures, discrepancy reports, test results, and recommendations for retest. In addition, any unplanned LYF test exceptions that occurred in the course of the test execution must be noted and evaluated. There may be recommendations for changes to operations concepts, constraints, processes, and procedures based on test results.

- For each LYF test executed, the following is generated:
  - As-run (redlined) test procedures
  - Test results (report/data) for LYF tests
  - Discrepancy reports (DRs)
  - Additional LYF test exceptions (if applicable)
  - Additional flaws/faults discovered (if applicable)
  - Retest plans and procedures (if applicable)

4.8 Perform Critical Fault Risk Management

As mentioned earlier, critical fault risk management is a part of the systems engineering process, and although it is not in the test domain, it is essential to the TLYF process. Critical fault risk management is executed in parallel with the TLYF process to monitor and manage the risk of critical, yet undetected faults. For any identified critical fault, risk management will track the fault and the LYF test exceptions to their ultimate disposition: exonerated, revealed, and repaired; or accepted by the program’s risk-decision authority (RDA).

The TLYF process helps systems engineering codify mission risk, as shown in Figure 29. When the results from the MCFA (flaw paths and contributors) are determined to be on the LYF test exceptions list, these present mission risk that need to be addressed (evaluated for criticality, mitigated, exonerated, or not). Even those exceptions that are included in a "standard set" are not necessarily low-risk and should be considered accordingly.

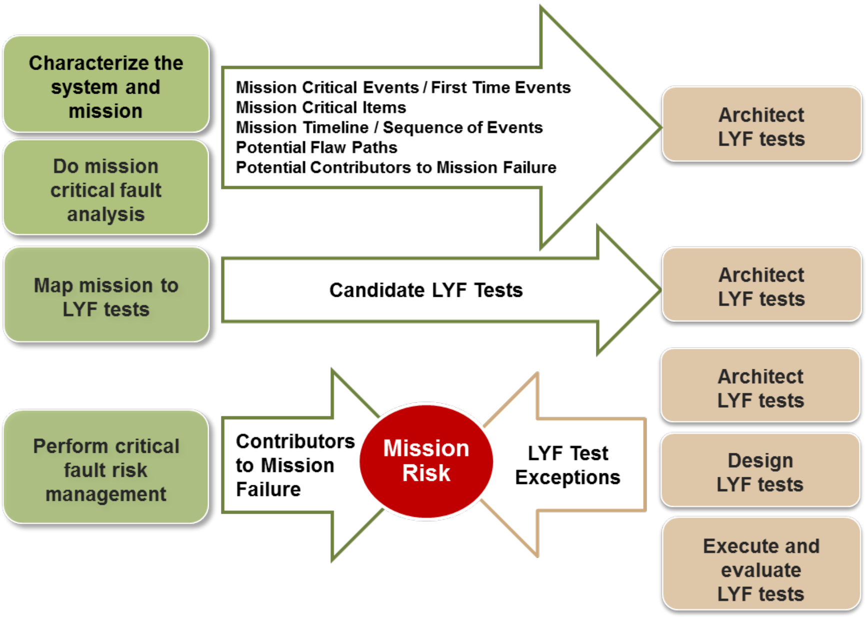

Characterize the system and mission
Do mission critical fault analysis
Map mission to LYF tests
Perform critical fault risk management
Mission Critical Events / First Time Events
Mission Critical Items
Mission Timeline / Sequence of Events
Potential Flaw Paths
Potential Contributors to Mission Failure
Candidate LYF Tests
Contributors to Mission Failure
Mission Risk
LYF Test Exceptions
Architect LYF tests
Architect LYF tests
Architect LYF tests
Design LYF tests
Execute and evaluate LYF tests

Figure 29. TLYF process and mission risk identification.

4.8.1 Process Details

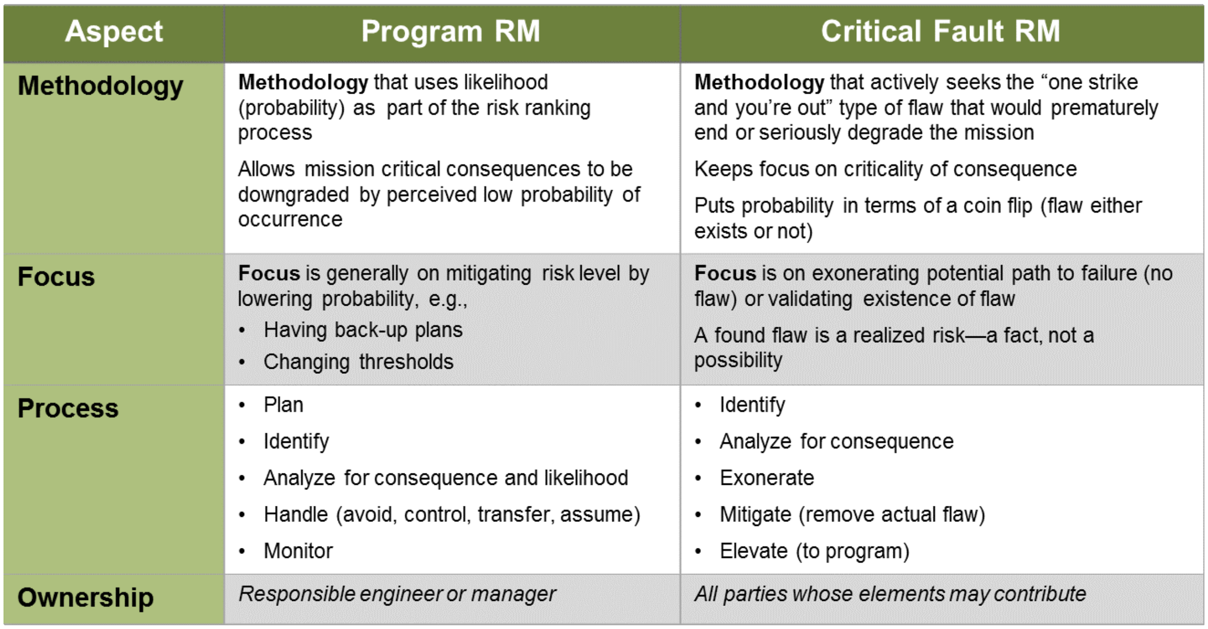
Performing Critical Fault Risk Management encompasses identification, analysis, mitigation planning and implementation, monitoring, and elevation of critical fault-related risks. These risks are based on potential mission-failure flaw paths and contributors identified during the *Do Mission-Critical Fault Analysis* step and the LYF test design process. This step includes performing additional analysis for identified LYF test exceptions by addressing any possible critical flaw that could have been missed. The goal is to exonerate each potential path to mission failure. Included in this goal is the mitigation of discovered flaws. It is essential to the process that critical fault-related risks that cannot be exonerated are identified and elevated.

4.8.1.1 Distinguishing Critical Fault Risk Management from Program Risk Management

The domains of critical fault risk management and program risk management are fundamentally different. Critical fault risk management has a narrow, technical focus. It addresses only those things that prevent operational success.

Program risk-management methodology tracks each program impacting risk by determining worst-case consequence and probability of occurrence. Few program-level risks are related to potential flaws in the system. The key differences between these two management approaches in terms of methodology, focus, process, and ownership are shown in Table 19.

Table 19. Distinctions between Program Risk Management and Critical Fault Risk Management



Aspect	Program RM	Critical Fault RM
Methodology	**Methodology** that uses likelihood (probability) as part of the risk ranking process Allows mission critical consequences to be downgraded by perceived low probability of occurrence	**Methodology** that actively seeks the "one strike and you're out" type of flaw that would prematurely end or seriously degrade the mission Keeps focus on criticality of consequence Puts probability in terms of a coin flip (flaw either exists or not)
Focus	**Focus** is generally on mitigating risk level by lowering probability, e.g., • Having back-up plans • Changing thresholds	**Focus** is on exonerating potential path to failure (no flaw) or validating existence of flaw A found flaw is a realized risk—a fact, not a possibility
Process	• Plan • Identify • Analyze for consequence and likelihood • Handle (avoid, control, transfer, assume) • Monitor	• Identify • Analyze for consequence • Exonerate • Mitigate (remove actual flaw) • Elevate (to program)
Ownership	*Responsible engineer or manager*	*All parties whose elements may contribute*

Since consequences rated critical (red) can cause mission failure, it is crucial to consider how to exonerate or reveal such flaws, even though the perceived probability of the flaw is low. The flaws shown in Table 5 would all have been identified as a low-likelihood probability of occurrence prior to launch, but they happened anyway. It is not appropriate to use the "likelihood of occurrence" probability in the critical fault risk assessment. The existence of a critical flaw has only two possibilities: either the flaw exists or it does not. The basis of estimate for the probability of all potential critical flaws is then 50 percent. If a program wishes to retain the probability assignment for each identified risk, this is the appropriate value for risks associated with potential critical flaws.

All flight anomalies were unexpected. If they were expected, they would have been mitigated before flight.

The key to critical fault risk management is to concentrate first on criticality and then on exoneration. LYF testing is one method by which a given flaw may be revealed or exonerated, but it is not the only method that can be used. If the risk is exonerated by tests that are not LYF, careful analysis is necessary to ensure that the LYF exceptions do not invalidate the test results.

When a given flaw cannot be exonerated within the available program resources, the risk should be elevated to the program risk-management process (see Figure 30). This provides the opportunity to use additional resources to exonerate the risk, or to decide to accept the risk with a full understanding of the criticality and impact of the potential consequences.

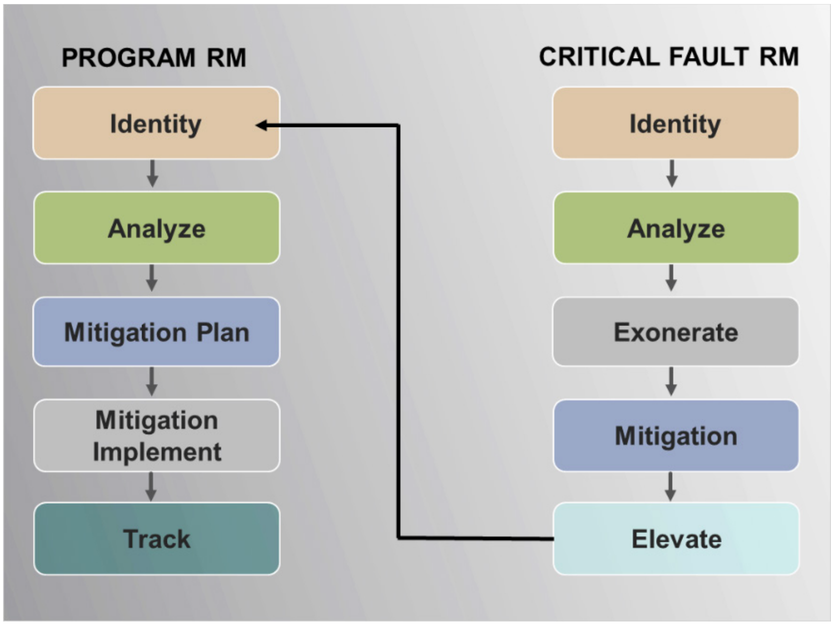


Figure 30. Program risk management versus critical fault risk management.

4.8.1.2 Exonerating Paths to Failure

It is not enough to identify known paths to failure. Methods for exonerating the potential critical flaws must be devised. Where the exoneration method depends on a LYF test, that test must be added to the list of LYF test candidates. It may make sense to exonerate the flaw by including specific mission characteristics into a non-LYF test. In deciding how to handle potential flaws it is important to consider the following:

- TLYF generally implies interactions among and concurrency of mission characteristics
  - Fault analysis must address these
  - Exoneration must apply to **all** concurrent contributors
- It is not enough to show that HW matches the schematic or that there is a redundant component
  - There must be evidence that there are no flaws affecting the critical as-built HW and SW interactions and dependencies
  - There must be evidence of transition and usage of critical redundant components

4.8.2 Outputs

Outcomes from this step include:

- TLYF-associated program risks
  - List of LYF test exceptions linked to MCFA fault branches
  - List of first-time/mission-critical events with no planned validation
- LYF test exception handling
  - LYF test exception risk evaluation

- Risk mitigation plans
  - Proposed new LYF test candidates (if applicable)

4.9 TLYF Process Events Schedule

Table 20 refers to the tasks associated with performing the TLYF process. It shows the iterative process for each of the steps and when to expect updates according to the system's lifecycle phase. It is intended to provide a framework for task expectations.

Table 20. TLYF Process Steps and Events Summary

Pre-Acquisition		Technology and Development			Engineering and Manufacturing		
TLYF Step	**RFP**	**SRR**	**SDR**	**PDR**	**CDR**	**Test Readiness Reviews (TRRs)**	**System Verification Review (SVR)**
Characterize the System and Mission	Used to scope TLYF process effort (initial list provided based on TRD and CONOPS)	List of mission-critical events/activities (nominal, first-time, critical, critical recurring, and contingency)	Updated	Updated	Updated		
Do Mission-Critical Fault Analysis	Define mission success (included phases and transitions)	List of mission-ending failure situations (including first-time and mission-critical events) covering all mission phases (fish head/tree tops	• Fish head/tree top signifying the failure situation • Fish bones/ branches for potential fault paths and contributors (including parallel contributors and processes) • Exoneration plan for each failure path	Initial analysis	Updated	Updated to account for redesign/test execution results	Updated based on test results
Map Mission to LYF Tests (Identify, Assess, Allocate)	• TLYF process effort described in RFP (SOW and CDRLs) • MOAs identified for associated resources and test activities	• Operationally realistic LYF tests identified, including allocation for MRT (linked to critical events, phases, and transitions) • Initial plan for test allocation (levels of integration/ mission/supplier)	List of prioritized allocated LYF tests (objectives, allocation of resources, high-level LYF test exceptions)	• Updated allocated LYF tests (MCFA results and pyramid allocations) • List of unallocated LYF tests • List of first-time/ mission-critical events with no planned validation	(See Architected LYF tests) Updates to list of unallocated LYF tests and list of first-time/ mission-critical events with no planned validation	(See Design LYF tests)	(See Design LYF tests)

Pre-Acquisition		Technology and Development			Engineering and Manufacturing		
TLYF Step	**RFP**	**SRR**	**SDR**	**PDR**	**CDR**	**Test Readiness Reviews (TRRs)**	**System Verification Review (SVR)**
Architect LYF Tests	If applicable, assign gov't architect to scope MRTs for contracts	High-level plans (MRT – gov't lead, other tests, contractor)	List of architected LYF tests with details (mission coverage, objectives, LYF test exceptions)	For each architected LYF test: • Mission phase/event/ failure situation coverage • Test plan • LYF test exceptions	Updated	(See Design LYF tests) Final plans (operationally realistic tests and MRTs)	(See Design LYF tests) Final plans (operationally realistic tests and MRTs)
Design LYF Tests	Identify design team for multiple contractor/agency tests	High-level plans (place holders for tests)	(See Architect LYF Tests)	(See Architect LYF Tests)	Final plans (operationally realistic tests and MRTs), including test plans, configuration, procedures, entrance/exit criteria, LYF test exceptions	Final plans updated for any redesign	
Execute and Evaluation LYF Tests				Review low-level tests	Updated as conducted	Updated as conducted	Updated as conducted
Perform Critical Fault Risk Management	Identify plans and associated teams	Initial plan based on LYF test list	TLYF associated program risks: • List of LYF test exceptions linked to MCFA branches • List of first-time/ mission-critical events with no planned validation	• Updated plans (identify new and more detail) • LYF test exception handling • LYF test exception risk evaluation and assessment • Risk mitigation plans • Proposed new LYF test candidates (if applicable)	Updated	Updated	Updated

5. Applying TLYF across the Lifecycle

The TLYF process is best incorporated at the beginning of the acquisition lifecycle. However, it is possible to introduce it at any time (even after CDR). The key is being able to incorporate key systems engineering practices (i.e., MCFA) that improve the test program's ability to reduce risk to mission (i.e., build/add operationally realistic LYF tests that address potential mission-ending faults).

Programs that are still in the design phase have the opportunity to fully implement the TLYF process. What is likely to suffer is that specific resources (e.g., testbeds or simulators) to support LYF tests may not be available within program constraints at this and subsequent phases. This is likely to lead to fewer LYF tests, missed opportunities for LYF development tests, missed risk-reduction LYF tests of lower levels of HW, lower fidelity of mission simulation leading to a higher number of LYF test exceptions, and a missed opportunity to align ground and space segment development for common equipment and SW. It may not be feasible to acquire the funding and resources needed to architect and design a dedicated LYF test. The alternative for injecting LYF testing is to propose modifications for existing system-level tests where resources are already scheduled and in place.

For space systems, it will likely not be possible to allocate LYF tests to the lowest levels for perceptivity and risk reduction after CDR, unless there are additional vehicles. These missed opportunities must be examined carefully for potential critical flaw escapes. There will be fewer program resources available for LYF tests. There are likely to be severe constraints on running LYF tests due to the lack of development of supporting equipment and SW. For low-volume production items, as is the case with space systems, it is best to carefully scrutinize each follow-on vehicle for new or added capabilities. Ground systems do not have this consideration under most architectures, which allows for units and other items to be removed for evaluation and test at almost any time in the lifecycle. Each new capability or upgrade should undergo the TLYF process.

5.1 TLYF Process Element Flow

The majority of this paper has focused on describing TLYF process implementation for a new acquisition. However, there are additional TLYF elements that directly relate to the process. These elements provide a mechanism for updating a program's systems engineering and test activities to improve those products and/or processes as they pertain to building perceptive operationally realistic LYF tests.

5.1.1 TLYF Process Planning

TLYF process planning allows the TLYF process to be formally placed on contract. It begins early in the acquisition lifecycle (i.e., DODI 5000.02, Pre-Phase A) due to much of the planning activities pertaining to the acquirer/buyer. There are a number of tasks associated with TLYF planning. The primary activities for the government/contractor team are:

1. Maintain awareness of TLYF-related policy and guidance

2. Support the establishment of an acquisition strategy by doing the following:

    a. Incorporate TLYF process language into the RFP package. Key customer program documents (i.e., statement of work, test and evaluation master plan, contract data requirements lists) are developed with recommended content for including TLYF-related tasks on the contract.

b. Ensure TLYF activities allotted for the program are consistent with the program's acquisition plan

c. Promote delivery of mission operations documentation and equipment in time to support operationally realistic LYF test planning, design, and execution (e.g., concept of operations, mission requirements, mission descriptions) by providing need dates

d. Plan and effectively schedule TLYF-related activities (systems engineering and test development) to fit into the program's integrated master schedule (IMS)

3. Assist in definition and identification of assets and resources needed to support operationally realistic LYF tests and document in the appropriate test strategy document. Ensure the following is addressed:

    a. Facilities or equipment to adequately stimulate or emulate mission payloads and vehicle systems (e.g., operational and test resources—simulators, emulators, testbeds, ground terminals, and specialized test facilities)

    b. User equipment to support tests

    c. Test data items (e.g., properly formatted data files, command data, telemetry database) to support independent space and ground element and segment operability tests

    d. Identify necessary participation from external organizations

    e. Identify necessary participation from trained operations and mission personnel

4. Create a high-level initial list of the following:

    a. First-time events, mission-critical events, situations, and associated key mission characteristics extracted from mission-related documentation

    b. A proposed list of operationally realistic LYF tests

        i. Mission event coverage (first-time and/or critical)

        ii. Critical mission characteristics required to achieve mission objective(s)

    c. The critical mission characteristics for each first-time and/or critical event that are not planned or cannot be tested in an operational manner prior to launch or fielding (LYF test exceptions)

    d. An evaluation of the mission risk for identified LYF test exceptions

    e. LYF development tests driven by the need to validate new operational concepts, new technologies in the context of the mission, or components used in new operational contexts

5. Define a high-level test plan for pre-ship and pre-launch mission readiness test(s) and include the mission timeline coverage (i.e., phases, sequence of events), estimated duration(s), and resources needed (operational and test)

6. Ensure the following are defined to promote realistic estimates from potential bidders:

    a. Description of acquisition scope that addresses requirements from mission-tasking organizations, end-user needs and requirements, and system-of-systems considerations

    b. Description of mission products, services, and desired outcomes

    c. Detailed concept of how the system will be operated that addresses mission requirements, system external interfaces, operations interactions, operational constraints, mission phases, and a preliminary operational timeline

    d. Resources that will interact with the system, including operational tools, operator planning and analysis tools, legacy systems, and heritage systems

5.1.2 TLYF Assessments

The TLYF assessment ensures that critical mission characteristics (i.e., hardware and software components, conditions, processes, interactions, transactions, and environments) are addressed and that associated risks are mitigated or managed. A program's test approach will likely not replicate all planned mission activities in test due to resource limitations. However, the decision not to address specific critical mission characteristics via an operationally realistic LYF test should be made from a risk perspective.

The value of the TLYF assessment lies in determining a program's tailored technical baseline of tests compared to recommended TLYF best practices in order to identify gaps and/or weaknesses in the program's execution approach. Communicating assessment findings early in a program's lifecycle can enhance the test approach and increase confidence in mission success.

5.1.2.1 Assessment Steps

The assessment follows a top-down approach, and consists of six steps (four directly linked with the TLYF process), as shown in Figure 31. It begins with program office coordination, addresses the steps in the TLYF process, and closes with a summary of findings, impacts and recommendations for addressing gaps.

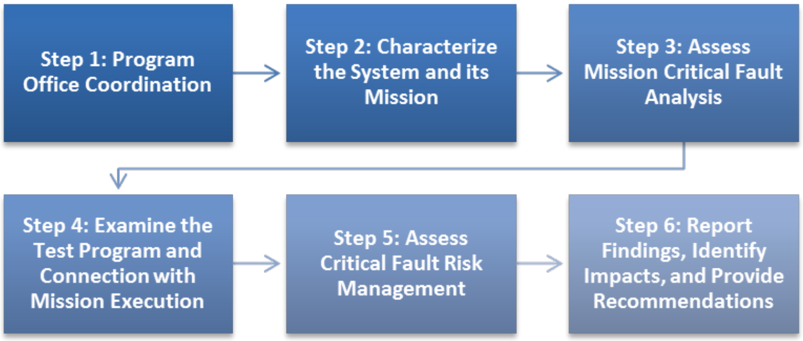

Step 1: Program Office Coordination
Step 2: Characterize the System and its Mission
Step 3: Assess Mission Critical Fault Analysis
Step 4: Examine the Test Program and Connection with Mission Execution
Step 5: Assess Critical Fault Risk Management
Step 6: Report Findings, Identify Impacts, and Provide Recommendations

Figure 31. TLYF assessment steps.

The assessment focuses on reviewing specific types of information/data. Sources may come from government documents, contractor work products, and/or interviews. Program work products vary based on contractual agreements and the contractor's internal approach to implementing their version of the TLYF process; therefore, a deeper level of looking may be required. The best approach is to be aware of the TLYF process, the types of information that provide insight, and the probable sources for that information. In most cases, a program's work products provide adequate source material to perform an assessment. If documented sources are not sufficient, it may be necessary to conduct specific interviews to get a clear story.

5.1.2.2 Assessment Criteria

The TLYF assessment focuses on documentation, people, processes, and products. Table 21 describes the criteria for all TLYF process assessments. Each assessment step is evaluated against criteria that reflect lessons incorporated into the TLYF process. Beneath each high-level criterion, a series of sub-criteria exist that inform the evaluation. Measurement against the TLYF process provides one dimension of the assessment. The other dimension involves the nature of identified gaps and the potential impacts for not addressing those gaps.

Table 21. Criteria for TLYF Assessment

TLYF Process Step	Criteria 1	Criteria 2	Criteria 3
Characterize the System	Clear, current, accessible, and complete system source material is available for all levels of supplier pyramid	Clear, current, accessible, and complete system source material is available for all levels of the integration pyramid (i.e., functional flows and interfaces decomposed)	A configuration management process exists to establish, update, inform, and disseminate related documents and subsequent work products (based on system-related HW/SW changes) (Inform across all pyramids)
Characterize the Mission	Clear, current, accessible, and complete mission source material is available for all levels of mission pyramid	Clear, current, accessible, and complete mission/operational source material is available for the "item" under review	Mission-related work products (first-time and mission-critical events) exist or can be easily derived from existing work products
Assess Mission-Critical Fault Analysis	Clear method (work products describing methods) for identifying mission-critical faults (mission failures and failure situations)	For each mission failure, a clear method for identifying potential contributors to the failure from a mission execution perspective (flaw paths)	Clear method for linking potential fault paths and contributors with an exoneration plan (I, A, D, T)
Evaluate Test Program and Operationally Realistic Tests	Clear method for identifying, assessing, and allocating operationally realistic tests (objectives, sequence, methodology tied to the mission)	Clear characterization of fidelity and configuration of allocated test resources used for operationally realistic LYF tests Test resources clearly defined and fidelity-specified for use in testing (testbeds, simulations, models, and any other item that substitutes for a mission characteristic) Work products that indicate test resources used for operationally realistic LYF tests are verified and validated	MCFA results (i.e., fault paths and contributors) and related work products are used as a basis for creating and scoping operationally realistic LYF tests
Assess Critical Fault Risk Management	Clear method for tracking and evaluating LYF test exceptions that are tied to critical flaw paths	Clear method for tracking evidence of exoneration for fault paths and contributors	Evident linkage from LYF test exceptions' risks and program risk management

5.1.2.3 Summarize Findings, Impacts and Recommendations

Once the data from work products and interviews has been provided and reviewed, the team develops a list of their findings. Findings include a list of items where TLYF assessment criteria is met or not met (i.e., specific gaps in systems engineering and/or test development). The summary focuses on the enterprise/system/subsystem under review. It includes a list of compliance items and gaps. For each gap, potential impacts and recommendations for mitigation are given. This offers decisionmakers the opportunity to make their own assessment of the associated risks of inaction.

There are three potential outcomes for decisionmakers:

1. Change the existing test(s) if areas of elevated risk are identified. Evaluate the testability (feasibility, practicality, perceptivity, and value-added) for conducting operationally realistic LYF tests to mitigate the elevated risk.

2. Define additional tests to be performed. The program can determine if additional resources such as equipment, personnel, facilities, or schedule are needed.

3. Assess the risk of not augmenting the test program to include adequate MCFA and operationally realistic LYF tests.

5.1.2.4 Lifecycle Consideration

A TLYF assessment can be performed on any program at any time in the acquisition lifecycle. However, there are nuances to be aware of when scheduling an assessment. Assessment objectives will shift as a program moves from the "early" phase to the "late" phase. Table 22 summarizes the distinction in objectives per program development phase.

Table 22. Program Phase and TLYF Process Assessment Objective

Program Phase	TLYF Process Assessment Objective	Outcome
Early	Evaluate government/acquirer TLYF-related work products, linkages, definitions, processes, plans, and decisions.	Recommended modifications to the contract are provided (e.g., tasks related to the TLYF process—MCFA, addition of operationally realistic tests and test resources, tasks that link MCFA results to test program, clarifications on LYF test exceptions and definitions of what is operationally realistic)
Mid	Evaluate contractor systems engineering and test development processes and work products	A summary of the findings which identify TLYF gaps and associated impacts for the enterprise/ system/subsystem under review, with specific impacts for each gap outlined and recommendations made for closing/addressing each gap
Late	Perform independent TLYF risk assessment for readiness for launch of SVs and LVs, as well as the readiness of associated ground support equipment	Risk assessment (LYF test exceptions, mission-critical fault escapes)

5.2 Acquisition Implications

In an increasingly budget-constrained environment, alternatives for applying the TLYF process need to be considered based on the type of system developed. Each has unique aspects and will be discussed here.

Aspects consist of contractual elements that require attention, potential limitations with implementing the TLYF process, and prospective opportunities to improve test programs to uncover mission-critical flaws.

5.2.1 Considerations

To the extent that the acquisition entity intends to include TLYF, documentation, necessary interactions among HW, SW, and databases, and procedures must be harmonized in official schedules. These are programmatic factors that need to be well thought out early in the lifecycle. These programmatic considerations are identified in section 5.1.1.

Programs may want an independent assessment to discover deficits and review risks in terms of TLYF implementation. This is considered a TLYF process assessment as discussed in section 5.1.2.

5.2.2 Space

There are many potential acquisition variations for the space element. Each variation will have associated considerations for applying the TLYF process. The customer (acquirer of the system) may buy single or multiple integrated space vehicles (ISVs—the spacecraft bus plus payloads). In the case of buying space vehicles, the customer will have a contract with an integration contractor to perform or oversee the design, integration, verification, and validation of the ISV. That contractor may build or subcontract the development and delivery of the spacecraft and payloads. In turn, those who are building the spacecraft and payloads may farm out various responsibilities to vendors.

Although the customer is only buying the space element, it is in their best interest (for mission assurance) to connect this acquisition to concurrent applicable acquisitions (e.g., a ground system); existing space systems (e.g., legacy vehicles) for the same mission; or other critical external systems/segments, if those aspects are not under the current customer's portfolio.

A customer may acquire one payload or multiple payloads, with the strategy to have those payloads integrated via another organization's space system acquisition (i.e., the host vehicle). Alternately, a customer may acquire a payload directly from one provider for integration by another contractor (government-furnished equipment), where the customer has both contracts.

The above-mentioned examples have considerations for propagating the TLYF process because the customer does not have direct control over processes at lower levels. In the case of a contract with an integration contractor who subcontracts for a spacecraft bus, a payload, one or more subsystems, or one or more units or assemblies, the acquisition customer should specify how the TLYF process should flow down to those subcontractors. The customer should also require that the integration contractor include subcontractor TLYF process workflow products into the final products for the acquisition. At some point in the acquisition, either the integrating contractor or their subcontractors will buy parts, materials, and subassemblies from vendors who are unaware of their ultimate usage. The acquisition customer must specify the aspects of the TLYF process that need to be applied all the way down the chain to ensure testability (see section 4.4.1.3) of the purchases for use in the application.

5.2.3 Ground

A ground system is usually designed to handle more than one space asset, whether multiple copies or versions of the same vehicle or several completely different satellites/missions. Ground systems are usually intended to be sustained for decades.

A new ground system development may or may not be done concurrently with the space or launch system(s) it is meant to control. Concurrent developments should have system-level considerations as described in section 5.2.5. One aspect that may be different between ground and space, even in concurrent acquisitions, is the scope of the “first-time” events for the ground system. The first-time analysis for space and launch elements are usually tied to the launch and orbital timelines. Ground systems being introduced to an operational environment may have significant “first-time” events well ahead of their concurrent space or launch counterparts. In some cases, ground systems are not built with a direct capability to “control” space assets, but are built to provide data/information between one source and another. Ground systems being set up without ties to a specific space or launch system development will have a very different set of first-time events that may not have anything to do with a launch or orbital timeline. Ground system acquirers and developers may need to look carefully at product and process introductions to adequately determine an appropriate set of first-time events.

Identifying mission success criteria for each phase of launch and space systems seems very obvious: deliver the space asset to the intended orbit, have the spacecraft bus systems operating per specifications to support payload operations, or provide quality products to users on time. For ground systems, first-time operational events are not necessarily tied to T-0, launch. Ground systems must have the capacity to simultaneously support multiple operational modes. The “mission success criteria,” necessary to do a viable MCFA, is a term not generally used in the ground systems community. However, customers for ground systems must address such criteria to fully accomplish the TLYF process.

New acquisitions addressing sustainment activities for existing ground systems may be faced with the challenge of introducing the TLYF process into a well-established operational structure. Some systems that are decades old may not have extant appropriate documentation or even adequate expertise for understanding the basic architecture and design of the system. Customers who want to proceed with using the TLYF process for such systems may need to allocate resources to reverse engineer or do other investigative engineering processes to obtain fundamental products that identify as-designed and/or as-modified aspects of the system.

5.2.4 Launch

As mentioned in section 3, the fundamental definition of the TLYF process is a pre-launch systems engineering and test activity. For space and ground elements, there is an assumption that LYF tests, especially a mission readiness test, can and should be conducted prior to “shipping” the articles from the factory. A space vehicle can be tested as a vehicle in the factory or at the launch site. A ground system can emulate mission operations before being installed in an operational environment. However, a launch vehicle generally is not a launch vehicle until it is integrated on its launch pad.

Launch vehicle segments will have some pre-ship testing, even pre-ship LYF tests, but those are not necessarily representative of a full vehicle LYF test. A customer must acknowledge that integrated LYF tests will be done either on test stands that can accommodate an integrated vehicle or on the actual launch pad. The latter is a test conducted very late in the launch flow compared to other acquisitions.

Most space vehicles and payloads will have items installed for flight that cannot be tested directly in a system-level LYF test. Generally the vehicle can be tested as an integrated unit for many LYF test objectives. The reverse is usually true of a launch vehicle. Unless the vehicle is specifically designed to be tested as an integrated entity, it may be that very little of the integrated system can be truly tested LYF prior to launch. This puts the burden of the process on doing a very thorough critical mission fault analysis, creating a set of LYF tests on very high-fidelity simulators, determining the maximum extent to which integrated testing can be done on the integrated vehicle, and performing an extensive critical fault risk analysis for those aspects that cannot be tested prior to launch.

5.2.5 Multiple Item Acquisitions

Much of the TLYF process was developed with low-volume, one-of-a-kind/first-of-a-kind articles in mind. Most of the historical TLYF escapes are associated with those first-time events. However, customers need to consider how much of this process should be done for subsequent builds of space and launch vehicles. Many space system acquisitions in the 21st century are oriented to small buys, typically 3 – 6 articles. Few space systems plan for larger volume production, but such is sometimes the case. Workhorse launch vehicles, such as the Atlas line, have already faced this problem.

There is a case for continuing the full TLYF process for small-number acquisitions, as the quantity of design changes due to testing and on-orbit results require continued examination of mission activity and flaw paths. For larger acquisitions, the TLYF process should be invoked to address planned changes in design and operations. Lessons learned from the Atlas program indicate that customers should also have methods in place to address unplanned changes, especially in vendor items.

5.2.6 User Equipment and Systems

User equipment is likely to be acquired completely independently of the space and/or ground systems. When new user equipment is being developed concurrently with new space and/or ground systems development, it is vital for the acquisition organizations (space, ground, and/or system) to make provisions to acquire representative units to include in space-to-user and/or ground-to-user LYF tests. In addition to new equipment, the space and/or ground system may need to accommodate legacy user equipment. LYF tests should include both existing and new user equipment in the cases where the mission will be carried out by such a mix.

There may be timing challenges to having new user equipment in place for use in end-to-end LYF tests. If new user equipment cannot be made available for these tests, the program should carry an exception and evaluate the risks. At least one space acquisition initially planned to only use legacy user equipment in its tests, but new user equipment was added at the recommendation of an independent evaluation team. The test with the new user equipment revealed a mismatch that was ultimately corrected in the space vehicle.

5.2.7 System of Systems

In any situation where the acquisition customer provides end items to an enterprise or system of systems, the acquisition customer must negotiate aspects of the TLYF process with other acquisition customers and operational organizations. These negotiations should address a number of issues, including, but not limited to:

- Concepts of operations
- Roles and responsibilities for system-level TLYF processes and products
- The need, fidelity, and timing for simulators and models

5.3 Summary

The TLYF process fosters smarter test design by using the mission as the basis and asking what could prevent mission execution success. It has implications for acquisition strategy, requirements definition, interactive ground and space product development, systems engineering, fault analysis and risk management. Recently, the Space and Missile Systems Center (SMC) formally directed programs to follow aspects of this process to mitigate risk to space systems and mission operations.

Following the process described in this guide promotes two sets of products for reducing mission risk:

1. Operationally realistic tests that are feasible, practical, perceptive, and value-added by addressing mission-critical events and potential flaw paths and contributors

2. A list of critical events and associated fault paths and contributors not tested or exonerated

The first set provides opportunity to uncover flaws in the system prior to flight/fielding unlike other testing approaches. The operational timeline and MCFA help to efficiently and effectively identify the specific test parameters and conditions. These tests are identified, assessed, and allocated to the test program. LYF tests are architected, designed, executed, and evaluated. All identified mission-critical failure paths are either exonerated or revealed and mitigated.

The second set offers decisionmakers an opportunity to assess the associated risks and determine how the risk will be managed. Critical paths that cannot be exonerated or revealed as part of normal program resources are either elevated as a program risk and handled within risk management resources or accepted by the full acquisition community.

6. References

1. J. D. White and L. G. Tilney. *Introduction to The Test-Like-You-Fly Process, Parts 1 & 2*. Materials presented at the 28th Aerospace Testing Seminar. The Aerospace Corporation. March 2014.
2. SMC Standard SMC-S-012. Software Development for Space Systems. 13 June 2008.
3. J. D. White, L. G. Tilney, T. L. Bergen, and C. P. Wright. Chapter 15*, Test Like You Fly: Assessment and Implementation Process for Prelaunch Mission Testing,* Aerospace Report Number TOR-2011 (8591)-2 Vol. 1 Space Vehicle Test and Evaluation Handbook, 2nd Edition March 2012.
4. T. S. Metodi. *Space Vehicle Testbeds and Simulators Taxonomy and Development Guide*. Aerospace Report Number TOR-2010(8591)-16, June 2010. Distribution limited.
5. J. D. White. *Proposed Requirements for the Test Like You Fly Process.* Aerospace Report Number TOR-2012(1315)-3, 20 July 2012.
6. MIL-STD 1833 *Test requirements for Ground Equipment and Associated Computer Software Supporting Space Vehicles*, 13 Nov 1989.
7. CJCSI 3170.01E. (Chairman of the Joint Chiefs of Staff Instruction).
8. Aerospace Technical Instructions and Procedures – TI-MA 3.0 Test Like You Fly Process.
9. ECSS E-10-03A, Space Engineering Testing, February 2002.
10. J. D. White and C. P. Wright. *End-to-End Testing in a Test Like You Fly Context*. Presented at the 23rd Aerospace Testing Seminar. USAF/The Aerospace Corporation, October 2006.
11. B. L. Arnheim and C. P. Wright. *Insight into the Effectiveness of System Level Thermal Vacuum Testing*. Presented at the 21st Aerospace Testing Seminar. USAF/The Aerospace Corporation, El Segundo, California. March 2003.
12. G. Beutelschies. *That One's Gotta Work. Mars Odyssey's Use of a Fault Tree Driven Risk Assessment Process.* IEEE. Jet Propulsion Laboratory, Pasadena California, 2001.
13. S. L. Hogan. *Effective Fault Management Guidelines*. Aerospace Report Number TOR-2009(8591)-14. The Aerospace Corporation, El Segundo, California, 2009. Distribution limited.
14. J. D. White, L. G. Tilney, and F. L. Knight. "Test Like You Fly: Assessment and Implementation Process." The Aerospace Report Number, TOR-2010(8591)-6, January 2010.
15. J. A. Kasper and D. W. Hanifen. Chapter 27, *Inter-Segment Testing,* Aerospace Report Number TOR-2006(8546)-4591Space Vehicle Test and Evaluation Handbook, Nov 2006.
16. Air Force Manual 63-119, Certification of System Readiness for Dedicated Operational Testing, (Section 1.2.5, pg. 6), 20 Jun 2008 [Certified current 14 July 2010].
17. J. D. White, L. G. Tilney, and F. L. Knight *Update on Implementing the "Test Like You Fly" Process into Government Space Programs.* Briefing presented at the 27th Aerospace Testing Seminar. USAF/The Aerospace Corporation, El Segundo, California. October 2012.
18. J. A. Kasper, D. W. Hanifen, G. A. Larsen, and A. V. Weatherfod. Chapter 11*, Inter-Segment Testing,* Aerospace Report Number TOR-2011 (8591)-2 Vol. 1 Space Vehicle Test and Evaluation Handbook, 2nd Edition March 2012.
19. J. D. White, G. A. Larsen, and D. W. Hanifen. Chapter 1*, Test and Evaluation Overview,* Aerospace Report Number TOR-2011 (8591)-2 Vol. 1 Space Vehicle Test and Evaluation Handbook, 2nd Edition March 2012.
20. DOD Instruction 5000.02, "Operation of the Defense Acquisition System," December 8, 2008.

21. J. D. White and L. G. Tilney. *Introduction to The Test-Like-You-Fly Process, Parts 1 & 2*. Materials presented at the 29th Aerospace Testing Seminar. The Aerospace Corporation. October 2015.

22. J. D. White and C. P. Wright. Chapter 33*, Test Like You Fly Guidelines,* Aerospace Report Number TOR-2006 (8546)-4591 Space Vehicle Test and Evaluation Handbook, 2nd Edition November 2006

23. Knight, F. L., "Space Vehicle Checklist for Assuring Adherence to 'Test-Like-You-Fly' Principles", Aerospace Corporation Report TOR-2009(8591)-15, 30 June 2009

24. N. Bhaskar, J. White, *Test Like You Fly/Days-in-the-Life Testing in STSS Demonstration Vehicles,* 08-MDA-3592. July 2008

25. Ray, Justin. *Sea Launch malfunction blamed on software glitch*, Spaceflight Now, March 30, 2000, available online at http://spaceflightnow.com/sealaunch/ico1/000330software.html

AEROSPACE REPORT NO.
TOR-2014-02537-Rev A

The Test Like You Fly Process Guide for Space, Launch, and Ground Systems

Approved Electronically by:

Jacqueline M. Wyrwitzke, PRINC DIRECTOR MISSION ASSURANCE SUBDIVISION SYSTEMS ENGINEERING DIVISION OFFICE OF EVP	Todd M. Nygren, GENERAL MANAGER SYSTEMS ENGINEERING DIVISION ENGINEERING & TECHNOLOGY GROUP

Aerospace Corporate Officer Approval:

Charles L. Gustafson, SR VP ENG & TECH
ENGINEERING & TECHNOLOGY GROUP

Content Concurrence Provided Electronically by:

Lindsay G. Tilney, PROJECT LEADER SR
SYSTEMS INTEGRATION & TEST OFFICE
MISSION ASSURANCE SUBDIVISION
OFFICE OF EVP

Technical Peer Review Performed by:

Jacqueline M. Wyrwitzke, PRINC DIRECTOR
MISSION ASSURANCE SUBDIVISION
SYSTEMS ENGINEERING DIVISION
OFFICE OF EVP

AEROSPACE REPORT NO.
TOR-2014-02537-Rev A

The Test Like You Fly Process Guide for Space, Launch, and Ground Systems

Special Programs Security Approval Granted Electronically by:

Alvania W. Thompson, SECURITY STAFF IV
CHANTILLY SPECIAL PROGRAMS SECURITY
OFFICE OF EVP

SY0150