Chapter 5
Availability, Operability, and Testability

Abstract Effective sustainment of systems and components during the operation and maintenance stages of the system life cycle require specific purposeful actions during the design stages of the system life cycle. The availability and testability of the system and its constituent components are essential to ensure that the systems continues to provide the required functions to its stakeholders. Availability and testability are non-functional requirements that exists at both the component- and system-level and are intertwined and interrelated. Inattention to testability concerns may lead to decreased availability having far reaching affects that include the systems own viability. The ability to understand how availability and testability are implemented in the design process and formal metrics and measurement processes for each non-functional requirements ensure that they are adequately addressed during all system design endeavors.

5.1 Introduction to Availability and Testability

This chapter will address two major topics. The first topic is availability and the second is testability. The first topic will review availability and the basic theory, equations and concepts that underlie its utilization. A section will address how availability is applied in engineering design and conclude with a metric and measureable characteristic for reliability.

The second major topic of this chapter will define testability and discuss how it is used in engineering design. The relationship to availability is reviewed and applied to the availability equation. The section concludes with a metric and measureable characteristic for testability.

The chapter has a specific learning goal and associated objectives. The learning goal of this chapter is to be able to identify how the attributes of availability and testability affect sustainment in systems endeavors. This chapter's goal is supported by the following objectives:

K.M. Adams, *Non-functional Requirements in Systems Analysis and Design*,
Topics in Safety, Risk, Reliability and Quality 28,
DOI 10.1007/978-3-319-18344-2_5

- Define availability.
- Describe the terminology used to calculate availability.
- Describe the relationship between maintainability and availability.
- Construct a structural map that relates availability to a specific metric and measurable characteristic.
- Define testability.
- Describe the relationship between testability and operational availability.
- Construct a structural map that relates testability to a specific metric and measurable characteristic.
- Explain the relationship between testability and availability.

The ability to achieve these objectives may be fulfilled by reviewing the materials in the sections that follow.

5.2 Availability and Operability

This section will review the basics of availability and how it is applied during systems endeavors. Availability is an important measure utilized in assessing a systems' ability to provide the required functions and services to its stakeholders.

5.2.1 Availability and Operability Definitions

Operability, from a systems engineering perspective, is defined as

> The state of being able to perform the intended function (IEEE and ISO/IEC 2010, p. 240).

Availability, from a systems engineering perspective, is defined as:

> 1. The degree to which a system or component is operational and accessible when required for use. 2. Ability of a component or service to perform its required function at a stated instant or over a stated period of time (IEEE and ISO/IEC 2010, p. 29).

From these definitions it should be clear that the non-functional requirement for operability is easily satisfied within the definition for availability. As a result, the rest of this chapter will treat operability as part of the non-functional requirement for availability.

Availability is usually simply stated as the ratio of the system uptime over the sum of system uptime and system downtime. Availability as a general concept is "the period of time for which an asset is capable of performing its specified function, expressed as a percentage" (Campbell 1995, p. 174).

However, availability has a number of unique definitions, characterized by either (1) the time interval being considered, or (2) the type of downtime (i.e., either corrective repairs or scheduled maintenance). There are other definitions presented in Table 5.1 that may contribute to improved understanding.

Table 5.1 Key elements that differentiate the definitions for availability

Definition	Source
Inherent Availability ($\mathbf{A_i}$): "Includes only the corrective maintenance of the system (the time to repair or replace the failed component), and excludes ready time, preventive maintenance downtime, logistics (supply) time, and waiting administrative time"	Elsayed (2012, p. 202)
Achieved Availability ($\mathbf{A_a}$): "Includes corrective and preventive maintenance downtime. It is expressed as a function of the frequency of maintenance, and the mean maintenance time"	Elsayed (2012, p. 202)
Operational Availability ($\mathbf{A_o}$): "The repair time includes many elements: the direct time of maintenance and repair and the indirect time which includes ready time, logistics time, and waiting or administrative downtime"	Elsayed (2012, p. 203)

Operational availability (A_o) is the most appropriate measure of system or component availability since it includes systems and their components in their real-world operational environments. The next section will address operational availability (A_o) mathematically.

5.2.2 Equations for Operational Availability (A_o)

Operational availability may be simply expressed as shown in Eq. 5.1.

Basic Availability Equation

$$A_o = \frac{System\, uptime}{System\, total\, time\, (uptime + downtime)} \tag{5.1}$$

Equation 5.1 may be expanded by including the maintainability terms *mean time between failure* (MTBF) and *mean time to repair* (MTTR) that were discussed in Chap. 4 and by introducing a new term *mean logistics delay time* (MLDT). The equation for operational availability is expanded and shown in Eq. 5.2.

Expanded Availability Equation

$$A_0 = \frac{MTBF}{MTBF + MTTR + MLDT} \tag{5.2}$$

MLDT includes both *mean supply delay time* (MSDT), *mean outside assistance delay time* (MOADT) and *mean administrative delay time* (MADT). The terms are defined as follows:

- MSDT includes delays during the acquisition of spare parts, test equipment, and special tooling required to accomplish the maintenance.
- MOADT includes delays due to the arrival of specialized maintenance personnel during travel to the systems' operational location to perform maintenance.

- MADT includes delays due to the development of procedures, permission to restrict system operation during maintenance, system isolation, and preparation for and the setting conditions (i.e., draining fluids, de-energizing circuits, etc.) within the system that permit maintenance actions to be accomplished.

Using these terms the MLDT is expanded and shown in Eq. 5.3.

Mean Logistics Delay Time Equation

$$MLDT = MSDT + MOADT + MADT \tag{5.3}$$

The operational availability (A_o) equation may now be re-written to include the additional terms from MLDT and is shown in Eq. 5.4.

Fully Expanded Availability Equation

$$A_0 = \frac{MTBF}{MTBF + MTTR + MSDT + MOADT + MADT} \tag{5.4}$$

5.2.3 Availability in Systems Design Efforts

Availability, like reliability and maintainability, is a major factor in determining system effectiveness. Availability is addressed in *IEEE Standard 1220—Systems engineering—Application and management of the systems engineering process* (IEEE 2005) and is specifically addressed in three areas.

- As an element of the requirements analysis task in Sect. 6.1.1
  Stakeholder expectations are defined and balanced in terms that address specific non-functional requirements such as availability.
- As an element of the requirements analysis task in Sect. 6.1.9
  The design team must define life cycle process concepts, which includes the maintenance support concept which will have a direct effect on availability.
- As an element of the design synthesis task in Sect. 6.5.2
  Design solution alternatives are evaluated and the non-functional requirement attribute availability is a distinct measure used in the discriminating of the alternatives.

A typical stakeholder requirement may state that *the system must be available at least 99.5 percent of the time*. By stating the required uptime the stakeholders are constraining the amount of downtime and the variables that contribute to downtime (i.e., MTBF, MTTR, and MLDT (i.e., MSDT, MOADT, and MADT)).

The next section will discuss how to measure availability.

Table 5.2 Four-level structural map for measuring availability

Level	Role
Concern	Systems Sustainment
Attribute	Availability
Metric	Component availability
Measurable characteristic	Mean time between failure (MTBF), Mean time to repair (MTTR), and mean logistic delay time (MLDT)

5.2.4 Measuring Operational Availability (A_o)

At the end of Chap. 3 we stressed the importance of being able to measure each non-functional attribute. A structural mapping that relates availability to a specific metric and measurement entity are required. The four-level construct for availability is presented in Table 5.2.

The section that follows will discuss an additional sustainment concern by addressing the non-functional attribute for testability.

5.3 Testability

In this section the basics of testability and how it is applied during systems endeavors will be reviewed. Testability is an emerging measure that could be utilized as a means for improving the ability to properly assess a system's conformance with the functions and services required by its stakeholders.

5.3.1 Testability Definitions

Testability, from a systems engineering perspective, is defined as

> 1. The extent to which an objective and feasible test can be designed to determine whether a requirement is met. 2. The degree to which a requirement is stated in terms that permit establishment of test criteria and performance of tests to determine whether those criteria have been met (IEEE and ISO/IEC 2010, p. 371).

Testability, as a non-functional requirement, has additional definitions shown in Table 5.3 that may provide improved understanding of the term and its use in systems design endeavors.

Using the oldest definition for testability, Valstar (1965) posits that testability is an element of the inherent availability (A_i) and can be related by the formula in Eq. 5.5.

Table 5.3 Additional definitions for testability

Definition	Source
"Testability, symbolized by τ, the mean-time-to-test or to find a failure provided that test equipment, facilities and manpower are available"	Valstar (1965, p. 54)
"The capability of a system that allows effective fault detection and diagnosis"	Kelley et al. (1990, p. 22)
"The tendency for failures to be observed during testing when faults are present. Software has high testability if it tends to expose faults during testing, producing failures for many inputs that execute a fault. Software had low testability if it tends to hide faults during testing, producing almost no failures even through at least one fault is present"	Voas and Miller (1993, p. 207)
"Testability can suggest places where faults can hide from testing, which testing cannot do. Testability complements formal verification by providing empirical evidence of behavior, which formal verification cannot do"	Voas and Miller (1995, p. 18)

Inherent Availability Equation

$$A_i = \frac{\omega}{\omega + \tau + \rho} \tag{5.5}$$

where

ω Mean time to failure or MTTF.

τ Mean time to find a failure provided that test equipment, facilities, and manpower are available.

ρ Repairability or mean time to repair or MTTR.

Re-written using the more familiar terms from the previous section, Eq. 5.5 becomes Eq. 5.6.

Inherent Availability Equation with MTTF and MTTR

$$A_i = \frac{MTTF}{MTTF + \tau + MTTR} \tag{5.6}$$

Because inherent availability considers only corrective maintenance, that is, faults that are caused by failure, testability (τ) also has a role in the prediction of the operational availability of a system. The operational availability Eq. 5.5 may be re-written as shown in Eq. 5.7.

Expanded Availability Equation with Testability

$$A_0 = \frac{MTBF}{MTBF + \tau + MTTR + MSDT + MOADT + MADT} \tag{5.7}$$

In summary, testability directly affects system availability.

> Availability is dependent on how well the operator can assess the condition of the system, how quickly he/she can detect and locate the cause of degraded or failed units, and how efficiently he/she can rectify the malfunction (Kelley et al. 1990, p. 22).

5.3.2 Testability in Systems Design

Testability may be categorized in two ways:

1. *Architectural testability*—"an evaluation of the characteristics of a system based on its design features and the specified intent of the design" (Kelley et al. 1990, p. 22).
2. *Modal testability*—"an evaluation of the testability of a system's design while configured to perform one or more specific functions. This type of analysis is based on the system's design features, its functional flow characteristics, and its performance specifications" (Kelley et al. 1990, p. 22).

Formal measures for testability attributes have been proposed for a number of design tasks and are presented in Table 5.4.

Design for testability is a technique whereby designers proactively ensure that their design decisions support the development of a robust test program throughout the systems life cycle (Buschmann 2011). For both cases in Table 5.4 the tasks and measures describe correlated testing criterion that brings a basis for evaluating design as part of a broader design for testability approach. By having a defined testability metric and associated characteristic that are measureable, design alternatives may be afforded another measure with which they may be discriminated.

Testability, like availability, is a major factor in determining system effectiveness. Availability is addressed in *IEEE Standard 1220—Systems engineering—Application and management of the systems engineering process* (IEEE 2005) specifically addresses testability in three areas.

- As an element of the synthesis task in Sects. 6.5.4 and 6.5.13
  Determine the degree to which testability has been included in the solutions.
  Assess the testability of design alternatives to determine built-in test (BIT) and/or fault isolation test (FIT) requirements.

Table 5.4 Examples of testability measures in design tasks

Design task	Measure	Source
Architecture development	Measures of the relative controllability and observability of signals in the system architecture. The testability information is derived from reachability graph analysis of the corresponding Petri Net representation of the system architecture	Jiang et al. (2000)
UML Class Diagram	Measure the number and complexity of the interactions due to polymorphic uses	Baudry et al. (2002)

Table 5.5 Four-level structural map for measuring testability

Level	Role
Concern	Systems Sustainment
Attribute	Testability
Metric	Component testability
Measurable characteristic	τ, the Mean time to find a failure provided that test equipment, facilities, and manpower are available

5.3.3 Measuring Testability

At the end of Chap. 3 the importance of being able to measure each non-functional attribute was stressed. In support of this importance, a structural mapping that relates testability to a specific metric and measurement entity are required. The four-level construct for testability is presented in Table 5.5.

5.4 Summary

In this chapter the non-functional requirements for availability and testability have been reviewed. In each case a formal definition has been provided along with additional explanatory definitions, terms, and equations. The ability to effect the non-functional requirement during the design process has also been addressed. Finally, a formal metric and measurement characteristic have been proposed for evaluating each non-functional requirement attribute.

The next Part of the text will shift the focus to concerns that are directly related to the design itself. The first chapter in the Part on Design Concerns will address the non-functional attributes for conciseness, modularity, simplicity, and traceability. The second chapter in Part III on Design Concerns will address the non-functional attributes for compatibility, consistency, interoperability, and safety.

References

Baudry, B., Le Traon, Y., & Sunye, G. (2002). Testability analysis of a UML class diagram. *Proceedings of the Eighth IEEE Symposium on Software Metrics* (pp. 54–63). Los Alamitos, CA: IEEE Computer Society.

Buschmann, F. (2011). Tests: The Architect’s best friend. *IEEE Software, 28*(3), 7–9.

Campbell, J. D. (1995). *Uptime: Strategies for excellence in maintenance management*. Portland, OR: Productivity Press.

Elsayed, E. A. (2012). *Reliability engineering* (2nd ed.). Hoboken, NJ: Wiley.

IEEE. (2005). *IEEE Standard 1220: Systems engineering—Application and management of the systems engineering process*. New York: Institute of Electrical and Electronics Engineers.

IEEE, and ISO/IEC. (2010). *IEEE and ISO/IEC Standard 24765: Systems and software engineering—Vocabulary*. New York and Geneva: Institute of Electrical and Electronics Engineers and the International Organization for Standardization and the International Electrotechnical Commission.

Jiang, T., Klenke, R. H., Aylor, J. H., & Gang, H. (2000). System level testability analysis using Petri nets. *Proceedings of the IEEE International High-Level Design Validation and Test Workshop* (pp. 112–117). Los Alamitos, CA: IEEE Computer Society.

Kelley, B. A., D'Urso, E., Reyes, R., & Treffner, T. (1990). System testability analyses in the Space Station Freedom program. *Proceedings of the IEEE/AIAA/NASA 9th Digital Avionics Systems Conference* (pp. 21–26). Los Alamitos, CA: IEEE Computer Society.

Valstar, J. E. (1965). The contribution of testability to the cost-effectiveness of a weapon system. *IEEE Transactions on Aerospace, AS-3*(1), 52–59.

Voas, J. M., & Miller, K. W. (1993). Semantic metrics for software testability. *Journal of Systems and Software, 20*(3), 207–216.

Voas, J. M., & Miller, K. W. (1995). Software testability: The new verification. *IEEE Software, 12*(3), 17–28.