Design & Evaluation Methodology For Built-In-Test

Donald H. Lord
Grumman Aerospace Corporation, Bethpage
Daniel Gleason, Member IEEE & ASQC
Rome Air Development Center, Griffiss AFB

Key Words—Built-In-Test Equipment, Testability, Maintainability, Availability.

Readers Aids—
Purpose: Widen state-of-the-art
Special math needed: None
Results useful to: Maintainability engineers

Abstract—This paper provides guidelines and procedures to optimize the design of built-in-test (BIT) during the conceptual phase of system design. Optimization of the BIT design is achieved by properly specifying three key design parameters: BIT effectiveness, mean corrective maintenance time, and BIT reliability. These parameters together with the BIT design-to-cost target form the baseline criteria for designing the BIT equipment during subsequent phases. The paper provides straightforward mathematical tools, sensitivity analyses, and tradeoff procedures.

INTRODUCTION

Quantitative built-in-test (BIT) performance is so extensively woven into the prime-system cost-effectiveness equation as to require its determination to be undertaken as an integral part of the prime-system design approach. The present tendency to design BIT as an afterthought must be replaced by systematic design methodology initiated during the conceptual phase. There must then be a rigorous and trackable BIT engineering design procedure during each subsequent phase of system acquisition.

The development and acquisition of cost-effective BIT is a prerequisite to the control of prime-system operating and support costs. BIT optimization can only come about by synchronizing its design with that of the prime system. Thus, during the conceptual phase, BIT design must be exposed to the same planning processes, specification requirements, design reviews, and tradeoff studies that are performed for the prime system.

The most critical step in BIT optimization takes place during the conceptual design phase, when BIT concepts and performance requirements are determined. During each succeeding design phase (the system, subsystem and detailed design phases) these initial requirements serve as a design-to-baseline. Thus, they are the key determinant in achieving an optimum BIT design. The objective of this paper is to provide analytic procedures and mathematical tools which permit the system designer to specify these BIT requirements in precise, calculable, and measureable engineering terms. In the sections that follow, the complex interactions between BIT performance parameters and system reliability, availability, maintainability and life cycle cost are analyzed. Sensitivity analysis and historical data are provided to assist the system designer in selecting and evaluating the impact of BIT performance. Ref. [1] is the basis for this paper.

Notation

\[ A_o \] operational availability
\[ C \] number of repair-in-place maintenance actions
\[ E \] BIT effectiveness
\[ F \] ratio of BIT uncertainty to BIT reliability
\[ L_m \] cost of a direct-maintenance man-hour
\[ \lambda_{system} \] prime system failure rate
\[ \lambda_{BIT} \] BIT failure rate
\[ M \] maintenance man-hours per flight hour
\[ M_a \] mean administrative delay time
\[ M_c \] mean corrective maintenance time
\[ MDT \] mean operational downtime per maintenance action
\[ MDT' \] mean operational downtime per malfunction
\[ M_l \] mean logistics delay time
\[ M_p \] mean preventive maintenance time
\[ MTBF \] mean operational time between failure
\[ R_{BIT} \] BIT reliability figure of merit
\[ R_{nd} \] number of no-defect removals at organizational level
\[ R_r \] number of removals due to failures
\[ U \] BIT uncertainty

BIT PERFORMANCE PARAMETERS

A typical current specification of BIT performance is:

"The BIT features shall be such that at least 98 percent of the equipment failures shall be detected; and, at least 99 percent of the failure indications shall result from equipment failures; also, BIT nonambiguity (the ratio of results isolated to a single unit, to the total number of probable malfunctions) shall be not less than 0.97."

These typical current BIT requirements are subject to interpretation, difficult (if not impossible) to measure, and seldom (if ever) achieved, in an operational environment.

The primary objective of BIT is to correctly detect system malfunctions and accurately fault-isolate to a single replaceable unit. The selected BIT performance parameters should 1) reflect this objective, 2) be measurable during system design, acceptance, and deployment, and 3) be relatable to prime system support performance parameters of reliability, maintainability, availability, and cost. This would permit BIT design tradeoffs to be performed meaningfully. By the above three criteria, the following BIT performance parameter was chosen:

BIT Effectiveness,
The values of the parameters comprising \( E \) can be empirically evaluated while the system is in the field. Maintenance-data systems typically provide the numbers of: 1) repair-in-place or adjustment actions, 2) remove-and-replace actions resulting from identified malfunctions, 3) no-defect removals subsequently identified. These three terms correspond to \( C, R_n, R_{nd} \) respectively, thus satisfying the second criterion for selection of effectiveness \( (E) \). Unfortunately, there are many factors inherent in maintenance-data systems which distort the compilation of such data. Data extracted from such maintenance-data systems are, therefore, somewhat suspect when used to evaluate \( E \) or \( U \). For this reason a special closed-loop data system dedicated to BIT evaluation should be implemented during the operational test and evaluation phase or be implemented on the limited basis after the system is fielded.

During the conceptual design effort, tradeoffs are performed between reliability, availability, and maintainability. The reliability, availability, and maintainability requirements are determined by prime-system operational analysis. In this process, the question arises, "What is the relationship between BIT Effectiveness \( (E) \) and the prime-system reliability, availability, maintainability and cost requirements?" - i.e., the third criterion for selection of \( E \) as the measure of BIT performance. These matters are discussed in the paragraphs which follow.

**BIT INFLUENCE ON SYSTEM RELIABILITY**

Adding BIT will increase the system complexity and hence the failure rate. A figure of merit used to express this impact is

\[
R_{\text{BIT}} = \frac{\lambda_{\text{system}}}{\lambda_{\text{system}} + \lambda_{\text{BIT}}}
\]

where \( \lambda_{\text{BIT}} \) is the failure rate of the circuitry dedicated solely to BIT functions. The literature generally concludes that the failure rate of BIT circuitry and/or software not exceed 10% of the system's failure rate. Then,

\[
R_{\text{BIT}} \geq \frac{\lambda_{\text{system}}}{\lambda_{\text{system}} + 0.1 \lambda_{\text{system}}} = 0.91.
\]

The above criterion for setting BIT failure rate to a maximum of 10% of the system failure rate is, in general, sound. However, good judgment should be used in applying this criterion to a specific system since it is, at best, a rule of thumb. Regardless of the magnitude of \( R_{\text{BIT}} \), it must be included in the prime-system's reliability calculation. \( R_{\text{BIT}} \) will also impact the prime-system availability as discussed in the following section.

**BIT INFLUENCE ON MAINTAINABILITY**

The parameters which have the greatest impact on operational readiness are mean operational downtime (MDT,) and mean operational time between failure (MTBF). Mean downtime includes unscheduled and scheduled maintenance time (Not Operationally Ready
EFFECTIVENESS, time
characteristics
For both avionic and ground electronics systems, $\overline{M}_{pr}$ is negligibly small and will be neglected in the discussion that follows.

BIT uncertainty ($U$) drives $\overline{M}_{i}$, $\overline{M}_{a}$, and $\overline{M}_s$ to a degree determined by the support, maintenance, and logistic concepts. For the purpose of comparing different system concepts, BIT uncertainty must be included. As a consequence, (7) must be modified to account for this uncertainty. The degree to which $\overline{M}_{i}$, $\overline{M}_{a}$, and $\overline{M}_s$ are affected by BIT uncertainty will vary from system to system. However, for comparison purposes, (8) estimates the impact of additional maintenance actions due to undetected failures, false alarms, and BIT ambiguity.

$$\text{MDT}_o \approx (\overline{M}_{ct} + \overline{M}_1 + \overline{M}_s)U$$  (8)

BIT INFLUENCE ON SYSTEM AVAILABILITY

A theoretical and practical definition of availability which is useful for comparing existing systems is—

$$A_o = \frac{\text{MTBF}_o}{(\text{MTBF}_o + \text{MDT}_o)}$$

$$= \frac{1}{(1 + (\text{MDT}_o/\text{MTBF}_o))}$$  (9)

where MTBF$_o$ is the operational mean time between failure, based on true maintenance actions, and MDT$_o$ is based on the assumption of one maintenance action per malfunction.

Adding BIT will increase the system complexity and hence degrade the failure rate to the degree that parts or software are added. Therefore, the MTBF$_o$ becomes MTBF$_o \times R_{BIT}$. In a similar manner, the uncertainty ($U$) of the test subsystem as reflected in no-defect removals will drive MDT$_o$ as a function of $U$ depending on the support, maintenance, and logistic concepts. Then, MDT$_o$ can be estimated by ($\text{MDT}_o \times U$). Substitution of these terms in (9) yields (10).

$$A_o = \frac{1}{[1 + (\text{MDT}_o \times U)/(\text{MTBF}_o \times R_{BIT})]}$$

$$= \frac{1}{[1 + (\text{MDT}_o/\text{MTBF}_o) \times F]}$$

$$F = U/R_{BIT}$$  (10)

These definitions apply equally to prime systems and subsystems, but do not apply to systems in standby, alert, or degraded modes of operation. Figure 2, a logarithmic plot of the above relationships, shows the impact of BIT performance and reliability on system availability.
For MDT/MTBF ratios \( \leq 0.01 \), prime-system availability is relatively insensitive to test subsystem performance. In this region, parameters such as cost, weight, size are dominant considerations in specifying BIT performance. For MDT/MTBF, \( > 0.01 \), prime-system availability is increasingly sensitive to the ratio, \( U/R_{\text{BIT}} \). As the test subsystem designer improves BIT effectiveness, it becomes more likely that the BIT reliability penalty \( R_{\text{BIT}} \) will offset this improvement. This suggests that the factor \( F \) might be specified to reflect operational requirements.

**BIT INFLUENCE ON SYSTEM LIFE CYCLE COSTS**

The objective in evaluating BIT candidates is to select the generic type of BIT and specify its performance requirements. It is conceivable that external test equipment can achieve sufficient test effectiveness, (Fig. 1) especially if the prime-system is functionally simple. Such simple systems are generally characterized by non-critical availability requirements, allowable mean times to repair in excess of one hour and mean times between failure in the hundreds of hours. In such cases the tradeoff between external test equipment and BIT should be made using conventional life cycle cost models.

However, for avionics, availability is normally critical, requiring \( M_a \), \( < 30 \) minutes. For such systems, there is no alternative to BIT. The designer's task is not one of life cycle cost tradeoffs so much as an effort to select the most cost-effective BIT parameter.

The question of determining "How much BIT and where?” still remains. What is the relationship between BIT or external test equipment effectiveness and life cycle costs? Obviously, because of diverse system missions, environments, and support concepts, there is no single solution to this question. Some design guidelines and cost-estimating relationships can be given, and these apply equally to both avionic and ground systems. Life cycle costs are subdivided into RDT&E, acquisition, and operation and support costs. The influence of BIT and/or external test equipment upon these elements is summarized as follows:

- **BIT RDT&E Costs** - This term reflects the implementation of new test techniques that require research, development, test and evaluation. Where these costs occur, they must be estimated in the form of a "grass roots” engineering estimate.

- **BIT Acquisition Cost** - This category includes all production and initial support costs. BIT production cost is the major cost driver to be evaluated, and includes both recurring and nonrecurring costs of BIT hardware and software. Maximum test subsystem production cost (both BIT and performance monitoring (PM)) can be estimated using the data of Figure 3.

The relationships in Figure 3 are based on experience and one or two data points for each curve. However, they are conservative since they represent maximum cost values. Figure 3 data apply equally to both BIT and external test equipment. Acquisition of initial support resources is less drastically impacted by BIT. Initial spares and special test equipment are reduced by something in the range of 10 to 20 percent, due to the effectiveness of BIT versus external test equipment. Training, trainers, technical orders, and other such elements seldom cost more than 15 percent of initial support costs, and BIT will generally have too small an impact to permit such related estimates during the conceptual phase.

- **BIT Operating and Supporting Costs** - These costs encompass total personnel and material costs necessary to operate and maintain the prime system over its life cycle, which is usually 10 to 15 years. Of all these costs, only base maintenance cost is significantly impacted by BIT. A reduction in no-defect removals may well be important in this respect. BIT is expected to change the required maintenance manhours for organizational maintenance as expressed by (11).

\[
\text{Change in 0-Level Cost} = M \times T(U_2 - U_1)L_R \quad (11)
\]

\( M \) estimated maintenance manhours per flight hour
\( T \) total flight hours over the life cycle
\( U_1 \) estimated BIT uncertainty
\( U_2 \) estimated uncertainty with external test equipment
\( L_R \) cost of a direct-maintenance manhour

**CONCLUSIONS**

Bit optimization must begin early in the conceptual phase and be synchronized with the prime-system mission...
and operational analyses. BIT effectiveness (E) is proposed as the measure of BIT performance because it is a direct function of fault detection and fault isolation capability, because it is calculable throughout the design process, and because it is measurable during system testing and operational phase.

The guidelines and mathematical tools relate BIT effectiveness to system reliability, availability, and maintainability, where these parameters are the over-riding criteria for specifying BIT, and are in turn related to life cycle cost. Thus, optimization of benefits versus costs is facilitated. Based on figures 1-3 and the given guidelines, design optima will fall into the following ranges depending on prime system mission, operational environment, and support concepts:

- Effectiveness (E) ranges from 75% to 95%
- Mean Corrective Maintenance Time (MCT) ranges from 30 minutes to four hours
- BIT Failure Rate is less than 10% of the overall prime system failure rate
- The Design to Cost Target for Test Subsystems is 10% to 20% of the prime-system production costs.

The resulting BIT performance will be less than perfect, because hardware and software designers are faced with an inaccurate knowledge of what will fail, and how often. Thus, it is recommended that BIT be tested and demonstrated in the field, prior to full-scale production to assure that design requirements will be met.

REFERENCES


AUTHORS

Mr. Donald H. Lord; Plant 31, B03; ATE Systems Engineering; Grumman Aerospace Corporation; Bethpage, NY 11714 USA.

Donald H. Lord received the BSME degree from Purdue University in 1943. During World War II he was a Navy Sonar Engineering Officer. In 1947 he was employed by Sperry Gyroscope Corporation where his assignments included electromechanical design, production engineering, logistic support and program management for a wide variety of surface search and missile guidance radar systems. In 1966 Mr. Lord joined Grumman Aerospace Corporation's Integrated Logistic Support Department. Over the past 14 years he has been responsible for support system analyses, management plans, automatic test equipment studies and life cycle cost analysis for various spacecraft and aircraft avionic equipments.

Captain Daniel Gleason; RADC/RBET; Griffiss AFB, NY 13441 USA.

Daniel Gleason (M'79) graduated from the University of Colorado at Boulder in 1976 with a BS in Aerospace Engineering. He received his MS in Systems Engineering from the Air Force Institute of Technology in 1976. Capt. Gleason is assigned to the Rome Air Development Center as a project engineer for the Reliability and Maintainability Techniques Section. Captain Gleason is a member of IEEE and ASQC and is an ASQC-Certified Reliability Engineer and Quality Engineer. He also holds a position as an adjunct faculty member at the State University of NY, College of Technology at Utica/Rome.

Manuscript TR80-173 received 1980 December 1; revised 1981 February 6.

Book Review

Ralph A. Evans, Product Assurance Consultant

Structured Systems Analysis: tools & techniques
Chris Gane, Trish Sarson, 1977, approx $30.00 prepaid, 373 pp.
Improved System Technologies, Inc.; 888 Seventh Avenue; New York, NY 10019 USA.

Table of Contents

1. The Need for Better Tools 10 pp
2. What the Tools Are and How They Fit Together 26 pp
3. Drawing Up Data Flow Diagrams 34 pp
4. Building and Using a Data Dictionary 41 pp
5. Analyzing and Presenting Process Logic 54 pp
6. Defining the Contents of Data Stores 34 pp
7. Analyzing Response Requirements 36 pp
8. Using the Tools: A Structured Methodology 34 pp
9. Deriving a Structured Design from the Logical Model 74 pp
10. Introducing Structured Systems Analysis into your Organization 14 pp
Glossary, Index 12 pp

This is one of several good books on this topic. (Several others are also reviewed in this issue.) It has many graphic examples to help understand the method. Structured analysis is not the same as structured programming, although their roots are similar. A major element of structured analysis is the transforming of a dirty problem (relatively undefined) into a sequence of cleaner problems (relatively well defined). The users (on all levels) usually begin with a rather large dirty problem, eg, "solve my accounting mess". Structured analysis is the process by which the analyst and the users find out together what feasible things are meant by "solve my accounting mess" and structure their findings so that those findings are understandable by everyone involved. If the system is to be put onto a computer, the structured analysis will help the computer programmers structure the set of programs.

This technique emphasizes the logical data flow (as opposed to the physical data flow) and the drawing of diagrams (rather than using lots of text) to illustrate that flow. A data dictionary is used to record and define exactly what is meant by each item on the flow diagram. The book deprecates flowcharts because they deal with the physical implementation rather than the logical flow.

The techniques are quite similar to those in Structured Analysis and System Specification, by Tom DeMarco and reviewed in this issue.

I recommend that you read this and several others of this type of book and pick the method of analysis that best fits your needs and environment. Most of the methods can be implemented somewhat gradually – which is a very good thing to do.

* * *