Estimating Methods for Production Test Labor

1984

Garry K. Stonebraker
University of Central Florida

Find similar works at: https://stars.library.ucf.edu/rtd

University of Central Florida Libraries http://library.ucf.edu

Part of the Engineering Commons

STARS Citation


This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
ESTIMATING METHODS
FOR PRODUCTION TEST LABOR

BY
GARRY KEITH STONEBRAKER
B.S.E., UNIVERSITY OF CENTRAL FLORIDA, 1981

RESEARCH REPORT

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Graduate Studies Program of the College of Engineering University of Central Florida Orlando, Florida

Spring Term
1984
ABSTRACT

This report mentions some of the difficulties faced by labor estimators when predicting the labor necessary to produce complex weapon systems. Specific attention is focused on estimating the durations and frequencies of testing, troubleshooting, and retesting activities. Emphasis is placed on estimating in a logical manner while using factors based on subjective judgment.
# TABLE OF CONTENTS

Chapter

I. INTRODUCTION ......................................................... 1

II. PRODUCTION TESTING .................................................. 5

III. STANDARD TIMES AND REALIZATION FACTORS .................. 12
    Standard Times ..................................................... 12
    Realization Factors .................................................. 14

IV. TEST ACTIVITY DURATIONS .......................................... 18
    Introduction .......................................................... 18
    Standard Times for Test and Retest ............................. 18
    Estimating Troubleshooting Times ............................... 26

V. ACTIVITY FREQUENCY .................................................. 33
    Role of Activity Frequency ........................................ 33
    Tests Modeled as a Bernoulli Process .......................... 34
    Modeling Test Arrangements ...................................... 36
    Multinomial Distribution of Failures .......................... 45
    Tabular Technique of Estimating Activity
    Frequencies .......................................................... 50

VI. COMPLETING THE ESTIMATE ......................................... 61

VII. SUMMARY AND CONCLUSIONS ....................................... 69

LIST OF REFERENCES ...................................................... 76
CHAPTER I
INTRODUCTION

The preparation of a cost proposal for the high volume production of complex weapon systems provides many challenges for the manufacturing labor estimator. The estimator must identify and evaluate the planned manufacturing processes for labor content and possible losses, quantify the findings, and submit the resulting labor hour estimate in sufficient detail to satisfy the cost breakout specified in the Request For Proposal (RFP) and auditors representing the procuring agency.

Estimates, by definition, involve some degree of judgment and uncertainty. The uncertainty associated with a manufacturing estimate is compounded when the item under consideration has a limited manufacturing history. This is often the case with weapon systems which will employ new technology in a continual effort to optimize performance. In addition, the lead time requested for the submittal of production proposals virtually eliminates the possibility of incorporating actual historical data into the first two or three production estimates. The manufacturing estimator must continually project into the future at a rate which exceeds the flow of historical data.
Given the environment from which the labor estimator projects future requirements, it is important to be on guard for some common errors which could adversely affect the accuracy of the estimate. Hajeck (1977) identifies several potential estimating errors including omissions, misinterpretations, misuse of estimating techniques, and the failure to adequately assess and provide for risks. The first three possible errors can generally be avoided by the estimator through diligence, scrutiny, and thoroughness. Risks and uncertainties in manufacturing operations are assessed and provided for through the inclusion of realization factors in the labor estimate. These factors usually increase the estimated labor hours according to their numeric values.

In general, standard labor hours are assigned to individual tasks and realization factors are used to relate these ideal times to actual factory requirements. The primary responsibilities of the manufacturing labor estimator include the application of accurate labor standards and the establishment of realization factors. The manufacturing labor estimator, as defined in this paper, is one who specializes in predicting the labor hours necessary to perform the fabrication, assembly, and test operations associated with the production of a deliverable end product.

Labor standards, representing standard times to accomplish activities, are usually developed through the ap-
application of some form of work measurement technique. Standard times which are fully documented with verifiable accuracy are known as engineered labor standards and are generally beyond reproach in contract negotiations. Realization factors, on the other hand, promise to generate an increasing amount of attention and will therefore require adequate substantiation as to their development, application, and possible verification. Captain J. C. Dougherty, Assistant Chief in the Ballistic Missile Office, describes this trend when he states that "the labor content represented by engineered labor standards will become 'de facto' acceptable in negotiations", and "the focus will then shift to the real 'negotiables', the realization factors" (1984).

Although the application of labor standards is usually accomplished by way of a detailed and structured procedure, realization factors are often established by less formal methods such as overall guesstimates. The purpose of this paper will be to investigate techniques of labor standard and realization factor development applicable to production testing activities.

In the production of complex weapon systems, testing frequently has a significant impact on the total recurring labor requirements. The impact is noticeable in two areas. First, the labor necessary to perform the testing is often a substantial portion of the total manufacturing cost, and secondly, the impact of test failures on manufacturing
costs (in terms of rework) is an important additional factor to be considered. Because of the impact of testing on the remaining manufacturing operations, it is recommended that the analysis of test activities described in this paper be completed prior to the determination of the realization factors that will be applied to fabrication and assembly operations.

The definitions of labor standards, standard times, and realization factors will be presented in subsequent chapters as they appear in the military standard on work measurement (MIL-STD 1567A). Although MIL-STD 1567A is not a requirement on every military contract of applicable scope, it is expected to become increasingly important in the future, as noted by Wade (1984).
CHAPTER II
PRODUCTION TESTING

Production testing includes the activities necessary to perform initial tests on deliverable hardware, troubleshoot units which have failed tests, and retest units as required. Typically, tests are performed after some assembly operations have taken place. Figure 1 illustrates the general production flow of a circuit card assembly. The addition of test failure routings in Figure 2 illustrates the potential impact production testing may have on the manufacturing effort of fabrication, assembly, and rework. More will be mentioned about this impact later in the paper. Meanwhile, the focus will be restricted to test, retest, and troubleshooting activities.

The term production testing is used to distinguish these tests from tests that are performed on a sampling basis such as lot acceptance tests, first article tests, or

Figure 1. Production Flow of a Typical Circuit Card
annual reliability tests. Production testing is intended to denote quality assurance tests which are part of the normal production flow and are performed on all of the units manufactured. Since this report is only concerned with production testing, the term production may be omitted in subsequent text.

Production tests may be designed or required for a number of specific purposes, but they all have the same objective of providing a defect-free product. For example, a test may be performed in order to provide manufacturing
personnel with information regarding the build-up of tolerances (mechanical or electrical) in the product which may be necessary to proceed with the manufacturing processes. Assemblies may be functionally tested to verify that performance parameters conform to standards or to screen for defective or substandard parts before they are incorporated further into the final product. Tests of this nature play an important role in the 100% in-process inspection of hardware. Similar tests are also performed on the final product prior to delivery to minimize the probability of failures in the field. Another common type of testing is known as pre-conditioning. During pre-conditioning, units are subjected to environmental stresses (e.g., vibrations and temperature extremes) and/or prolonged periods of operation so that weaknesses in the system can be detected and repaired (or discarded and replaced). Pre-conditioning tests may be performed on any level of hardware from piece parts to final assembly. The survivors of pre-conditioning tests are then referred to as "tried and true".

Whenever a test activity ends with an unsatisfactory result, the rejected unit will most likely be submitted to a troubleshooting activity in order to isolate and identify the cause(s) of failure. The troubleshooting activity associated with a particular hardware level test is defined as the activity necessary to identify the cause(s) of
failure to the next lower hardware level or to the inte-
gration and assembly effort. For example, Figure 3 illus-
trates the production flow associated with three circuit
card assemblies and the integration with a motherboard and
placement into a chassis to form an electronics assembly.
The failure of the test by the electronics assembly may be
traced to any one or more of the three circuit card assem-
blies (CCA), the motherboard, or the effort associated
with the integration and assembly (I&A) of the parts. If

Figure 3. Production Flow of an Electronics Assembly
one component on one circuit card assembly had caused the failure, the troubleshooting activity associated with the electronics assembly test would identify the particular CCA at fault (which is at the next lower level of hardware). The troubleshooting activity which is necessary to identify the specific component on the faulty CCA would usually be a separate activity performed on the particular CCA after removal from the electronics assembly. This activity would be considered a circuit card assembly troubleshooting activity.

The resubmittals of a unit to a test which was previously performed is termed a retest. The unit being resubmitted may not be completely unchanged (i.e., failed component parts may have been replaced), but if the unit retains the same manufacturing serial number it will be considered essentially unchanged. The consideration of the new replacement parts having an effect on the probability of the system (now consisting of many parts with different ages) failing subsequent retests is beyond the scope of this paper and will not be considered. Readers interested in this effect are referred to the textbooks authored by Shook and Highland (1969), Jensen and Petersen (1982), and Dhillon (1983) for information and additional references on the topic of renewal theory in reliability.

A retest may be performed after a unit fails the initial submittal to a test and undergoes troubleshooting and
repair. The electronics assembly example illustrates this situation. Notice also that a retest may be required even though the initial test was completed satisfactorily. This situation can also be illustrated by the electronics assembly example. The CCA could have passed the original functional test performed on it, but, because it suffered a component failure during the subsequent testing at the electronics assembly level, the CCA must be retested at the functional CCA test after repair. This does not imply that all lower level tests will be repeated. The level of retesting which will be performed may be inhibited by processes which alter the physical characteristics of the unit after initial testing. For example, conformal coating on circuit card assemblies greatly inhibits the ability to re-submit a CCA to in-circuit testing. It may be determined that the effort associated with a particular retest is too great of a labor expense for the information which might be gained. A formal quality engineering document will usually provide information regarding the level of retesting required to assure the cause of a failure was diagnosed and corrected satisfactorily before incorporating (or reincorporating) the unit further into the product.

It is probably apparent that an estimate of labor (or equipment) requirements necessary to test, troubleshoot, and retest must consider at a minimum, both the duration of each activity (a time standard) and the frequency of occur-
rence for each activity (a realization factor). The following chapter will discuss standard times and realization factors in general. Subsequent chapters will discuss specific applications to production testing.
CHAPTER III
STANDARD TIMES AND REALIZATION FACTORS

Standard Times

The fundamental unit of measurement which forms the basis of most manufacturing labor estimates is standard time. A particularly descriptive definition of standard time is provided by Engwall (1984):

Standard time is defined as the time determined by accepted Industrial Engineering techniques to be required by a qualified operator, displaying normal skill and effort, encountering normal delays and fatigue, under capable supervision and following the prescribed method for completing a defined operation.

The definition above could easily apply to normal time if the text regarding delays and fatigue were disregarded. Standard time is often defined as normal time factored to include allowances for the workers personal needs, mental and physical fatigue, and unavoidable delay (PF&D). When standard times are applied to production activities directly identifiable to a specific task, the results are known as labor standards.

The Industrial Engineering techniques mentioned in the definition of standard time are further specified in MIL-STD 1567A. The military standard divides labor standards into two classes. Type I engineered labor standards refer to standards established through the use of a recog-
nized work measurement technique such as time study, standard data, a recognized predetermined time system, or a combination thereof to derive at least 90% of the associated normal time. Readers not familiar with work measurement are referred to textbooks authored by Barnes (1968), Mundel (1978), Tucker and Lennon (1982), Karger and Bayha (1977), and Karger and Hancock (1982). Type I standards also require accuracy of ±10% with a 90% or greater confidence at the operation level. Documentation requirements include an operations analysis, the standard (prescribed) method of performing the task, any performance rating applied, and a record of all time values used in determining the final standard time. Type II labor standards are identified simply as those standards which do not meet the requirements of Type I standards.

The Type I standard is obviously the preferred unit of measurement. Accordingly, a further requirement of MIL-STD 1567A is 80% coverage of all touch labor hours with Type I labor standards (or as a minimum, a plan to upgrade Type II standards to Type I with 80% coverage).

The development of labor standards related to production testing will be discussed in greater detail in the following chapter. The discussion is not intended to be a tutorial for work measurement, but rather a guide for the application of techniques to the particular tasks associated with production testing.
Realization Factors

A realization factor is defined in the military standard on work measurement as:

A calculated factor (exclusive of personal, fatigue, and delay (PF&D) allowances) by which labor standards are modified when developing actual man-hour requirements.

This definition was updated in the latest revision of MIL-STD 1567, MIL-STD 1567A released March 11, 1983, to the following two part definition:

(a) A ratio of total actual labor hours to the standard earned hours.
(b) A factor by which labor standards are multiplied when developing actual/projected man-hour requirements.

The definitions imply that the term realization factor applies to one all-encompassing multiplier. The final product could be identified in that manner, but the application of multiple realization factors is recommended because it will provide greater visibility into the specific allowances being considered by the estimator to relate labor standards to real world requirements. Commonly employed realization factors may be classified into one of three general categories:

1) Contingency allowances
2) Performance factors
3) Allowances for losses and non-standard activities

Contingency allowances are ordinarily based on judg-
ment and are usually a function of the project maturity. Examples of factors which fall into this category include terms such as standards growth factors, engineering change factors, confidence factors, or in a more disparaging sense, fear factors. These factors will generally reflect the labor estimators faith in the labor standards, manufacturing processes, and a stable product design at the time the estimate was made. With regard to production testing, the estimator may choose to employ a contingency factor if a planned test is in the preliminary stages of development and there is little or no historical data from tests of a similar nature available for review.

Performance factors are used in a general sense to account for actual time expenditures (when producing good parts) that are different than the engineered labor standards. The additional, or perhaps lower, expense will then be attributed to start-up costs (starting performances) and the complementary manufacturing progress function (learning curves). Selection of performance factors for estimating total labor hours over a fixed quantity of units is often one of the labor estimators final tasks. The selection and use of performance factors for estimating production labor requirements will be discussed in more detail in the chapter titled Completing the Estimate.

The third category of realization factors, allowances for losses and non-standard activities, are included in an
estimate to provide for the additional effort which will be required to recoup certain losses associated with many production processes. Specifically, the experienced labor estimator should be aware that some manufacturing activities will require total duplication when in-process work is scrapped; and that other, non-standard, activities must be performed to rework items which do not satisfy the acceptance criteria as submitted. Recognition of these additional expenditures of human (and equipment) resources is particularly important in the defense industry since most weapon systems are characterized by very tight tolerances and strict criteria used to assess conformance to engineering specifications.

With respect to production testing, the relationship of product testing to scrap losses and rework requirements was illustrated in Figure 2 in the second chapter. The similarity between the non-standard activities of troubleshooting and rework can now be noted. Neither activity is planned as part of the normal production flow. However, it is almost certain that some items will be submitted for troubleshooting and rework. The detailed work elements can rarely be determined prior to submittal to either activity, thus requiring judgment as to the average duration of all such activities for estimating purposes. In both cases, the necessity of performing the task is dependent upon some potential defect in the unit. Parts which cannot
be reworked and are scrapped have an effect upon the fabrication and/or assembly operations similar to the effect test failures have upon tests in the form of retests. In most cases, when a retest is performed exactly like the initial test, the duplicate effort required to replace the scrap loss can be calculated with percentage multipliers for estimating purposes. A technique for deriving the retest percentages and troubleshooting percentages will be discussed in the chapter titled Activity Frequency. The potential for incorporating the results into the manufacturing (fabrication/assembly/rework) estimate of labor will be discussed briefly in the concluding chapter.
CHAPTER IV
TEST ACTIVITY DURATIONS

Introduction

The labor required to perform the activities of testing or retesting can be represented by compilations of standard times applied to the individual operations which comprise the tests. However, troubleshooting activities are not usually quantified as easily, due to the inherent uncertainty of the work content in this type of activity. The purpose of this chapter is to mention some considerations which will aid in the development of standard times to represent these activities.

Standard Times for Test and Retest

The standard times for test/retest activities are usually developed by identifying the work elements of each activity and quantifying the elements with time standards. The level of detail used to describe the elements depends in part on the method chosen to quantify them. A standard data system may be available which provides normal times in a somewhat macroscopic form for application to common procedures such as the handling of circuit cards for testing or the continuity testing of wiring harnesses. The tests which are peculiar to a specific product will usual-
ly require a lower level description of tasks for standard
time application purposes.

Other common methods of determining and applying
standard times include the use of predetermined time sys­
tems and time studies. Predetermined time systems general­ly offer a much more microscopic view than do the standard
data systems. If an acceptable standard data system (which
is based on groupings of predetermined time system elemen­
tal values) is available, the direct use of a predeter­
mined time system is usually avoided.

Most tests of complex systems involve the use of com­
puters to control the test inputs, collect and compare da­
ta, and provide the results. Some tests are fully automa­
tic, while others may require varying amounts of manual in­
teraction (semi-automatic). The use of time study tech­
niques for the computer controlled portions of a test, if
possible, is recommended. The test time, even for the ful­
ly automatic portions of a test, may tend to vary from unit
to unit. It may be preferrable to observe several test cy­
cles, at a minimum, prior to setting the time standard.
The labor estimator must also be aware of any potential
changes to the software used in tests in order to determine
the possible effect on standard times.

The accuracy requirement for Type I labor standards
may be assured through the careful application of an accep­
table standard data system (or predetermined time system)
of known accuracy. The question of additivity of elemental times has been approached by several authors including Buffa (1956) and Smith (1978). The major concern is that the variances of the sum of elemental data becomes increasingly large as the number of elements increases. Both Buffa and Smith note the actual results of adding elemental data are generally acceptable in practice, however the purist may still hold justifiable reservations. This question is not mentioned specifically in the military standard on work measurement, but the accuracy requirements are recommended to be met at the "super operation" level consisting of times of approximately one-half hour. The accuracy of time studied operations may be verified through the use of formulas which assist the estimator in determining the number of samples necessary based on the variance of the observed times. The accuracy obtained for any time standards applied through the use of a standard data or a predetermined time system can easily be verified through the use of time study observations once the system is in place. The derivation and explanation of the formulas are included in most textbooks on work measurement. Interested readers are referred to these sources for further information and the assumptions and conditions associated with the formula use.

In the early stages of a project, the estimator may be faced with the necessity of approximating times for
tests which are not in place and possibly lack clear definition of labor content. The estimator must turn to persons who are most well-informed about the tests for support in estimating the test times and labor requirements. The representatives of test engineering or quality engineering may provide the estimator with the necessary support. In these cases is important for the estimator to break down the tasks associated with each test into categories such as loading, test time, and unloading. These distinct tasks can then be quantified by approximation if necessary. This approach forces the estimator to simulate the physical activities and reduces the chance of omissions due to superficial examination. It is common to refer to time values estimated in this manner as standards based on engineering judgment.

Table 1 lists some broad elements which are common to many tests. A typical frequency required for each element is included to distinguish set-up operations from those performed per unit under test (UUT). The use of automatic test sets will often require a certain amount of time to be spent performing self-tests on computer equipment before running any tests. These make ready activities consume time which is then prorated over the number of units expected to be processed until the next self-test is required (usually daily). Additionally, some test sets (e.g., circuit card test sets) may make use of one computer console
<table>
<thead>
<tr>
<th>Element Description</th>
<th>Freq.</th>
<th>Normal Time</th>
<th>Standard Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read work instruction daily</td>
<td>daily</td>
<td>5.0</td>
<td>5.88</td>
</tr>
<tr>
<td>Check equip. calib.</td>
<td>daily</td>
<td>5.0</td>
<td>5.88</td>
</tr>
<tr>
<td>Obtain disc</td>
<td>daily</td>
<td>5.0</td>
<td>5.88</td>
</tr>
<tr>
<td>Power-up computer and run self-test</td>
<td>daily</td>
<td>25.0</td>
<td>29.41</td>
</tr>
<tr>
<td>Obtain adaptor, hook-up, self-test</td>
<td>lot</td>
<td>10.0</td>
<td>11.76</td>
</tr>
<tr>
<td>Obtain UUT (or lot)</td>
<td>lot</td>
<td>3.0</td>
<td>3.53</td>
</tr>
<tr>
<td>Place in test set</td>
<td>UUT</td>
<td>2.0</td>
<td>2.35</td>
</tr>
<tr>
<td>Hook-up &amp; begin test</td>
<td>UUT</td>
<td>14.0</td>
<td>16.47</td>
</tr>
<tr>
<td>Test time</td>
<td>UUT</td>
<td>35.0</td>
<td>41.18</td>
</tr>
<tr>
<td>Disconnect &amp; remove</td>
<td>UUT</td>
<td>5.0</td>
<td>5.88</td>
</tr>
<tr>
<td>Review test results, complete paper</td>
<td>UUT</td>
<td>9.0</td>
<td>10.59</td>
</tr>
<tr>
<td>Review test results, complete paper, disposition UUT</td>
<td>UUT</td>
<td>9.0</td>
<td>10.59</td>
</tr>
<tr>
<td>Remove adaptor</td>
<td>lot</td>
<td>7.0</td>
<td>8.24</td>
</tr>
<tr>
<td>Shut-down computer</td>
<td>daily</td>
<td>9.0</td>
<td>10.59</td>
</tr>
<tr>
<td>Remove disc &amp; place aside</td>
<td>daily</td>
<td>2.0</td>
<td>2.35</td>
</tr>
</tbody>
</table>

Daily rate = 40 units    Lot size = 20 units

Extended standard = \((59.99/40) + (23.53/20) + (76.47)\) = 79.15 minutes
with various adaptors to test individual types of parts, distinguished by different part numbers and configurations, but of the same generic class. The self-tests of adaptors may be required at a rate different than the computer. For example, Table 1 illustrates the case of 40 units per day planned for the computer, made up of 2 part number lots of 20 units each. The prorating of set-up activity times over the number of units tested per set-up and adding the result to the sum of the UUT run times provides the estimator with one value, the extended standard. The extended standard is partially based on the planned average rate of production and may vary with different production contracts.

Personal, Fatigue, and Delay Allowances

The example in Table 1 also distinguishes between normal and standard times for each task. The allowances for Personal, Fatigue, and Delay (PF&D) are factored into the normal times for tasks, resulting in standard times. The allowances for PF&D are often set at 5% each for a total allowance of 15%. The Department of Defense publication titled STANDARDIZATION OF WORK MEASUREMENT (DoD 5010.15.1-M Basic Volume, September, 1973) offers guidelines for applying PF&D allowances in any particular situation. Possible reasons for increasing PF&D allowances include strict environmental conditions such as a clean room or an area containing explosives. In these situations the
workers must take special precautions before entering the area in the morning, after breaks, after lunch, as well as when they are working. This situation can necessitate additional allowances for personal needs and fatigue. When the layout of a production area demands close coordination between work stations, additional allowances for unavoidable delay may also be appropriate. The application of PF&D allowances is generally understood to mean that of the available hours for work in a day, a certain percentage will be reserved for PF&D. For example, in an eight hour day (480 minutes), 72 minutes (@15% PF&D) will be allowed for Personal, Fatigue, and Delay. The conversion of normal time to standard time is therefore:

\[
\text{Standard Time} = (\text{Normal Time}) \left(\frac{1}{1-\text{PF&D}}\right)
\]

where PF&D is expressed in decimal form

Additional Considerations

Some additional points the labor estimator must consider include the difference between machine and labor time. Some tests may be of a lengthy duration but do not require an operators constant presence. In these cases (e.g., pre-conditioning tests) the estimator should calculate the minimum time required for an operator to load, monitor, and unload units, as well as the machine cycle time. It is important to note that unless the test can be left completely unattended, the operators time cannot be
reduced below the machine cycle time until it is assured that adequate additional tasks will be available to occupy the operators idle time. Management should be notified of the possible cost savings obtained by sharing the operator. The labor estimator will then use either the machine time to reflect the labor standard or estimate the amount of time the operator will probably spend (not necessarily the absolute minimum possible) performing the test.

Another consideration of possible consequence concerns retests which are of shorter duration (planned) than the initial test of a unit. It often occurs that some lengthy tests, consisting of many cycles, may not require total replication when retesting. In cases where retests are planned to be considerably shorter than the initial test, the times for retests should be maintained separately to be multiplied by the expected frequency of retests as described in the following chapters.

Some tests are identified as a single test in documentation when they are actually comprised of two or more mini-tests which could stop the test from proceeding upon failure. It may be easier to treat these tests as separate by dividing the standard time among them. This is particularly true in cases, such as hot/cold tests, where the yields expected from each mini-test are different and the unit may be removed upon failure.
There are other situations which may occur and require special consideration, but are beyond the authors' experience. One extremely complicated case (in terms of detailed standards application) is the consideration of the new generation of automatic test equipment. These test sets are designed to provide not only test results, but also diagnostics of failures through search routines based on the type of problem encountered. In other words, a test which encounters data out of range in the first few checkpoints will not continue to test but will branch into troubleshooting routines in order to isolate the area of failure. These test sets have great potential to reduce the troubleshooting labor requirements, but the labor estimator should recognize the impractability of assuming manual troubleshooting would be completely eliminated. The next section describes approaches to estimating troubleshooting requirements when conventional methods involving considerable human involvement are employed.

Estimating Troubleshooting Times

Troubleshooting activities can rarely be quantified in the same manner as test activities. The troubleshooting procedure for a given failure is similar to a sequential decision problem where the result of the current task will determine the next task, if any, to be performed. In large complicated systems, the allowance for troubleshooting may be estimated as a percentage of the total test/retest labor
hours. In other systems, it may be possible to identify the most likely steps that would taken to isolate the failure at any given hardware level.

The modular nature of many weapon systems aids the troubleshooting effort. In the case where two major assemblies (A and B) form a final assembly (C), the troubleshooting effort of the final assembly will be undertaken to isolate the cause of failure between A, B, or the integration and assembly effort (I&A). The troubleshooting activity would probably begin with checking the final assembly for loose connections. If no loose connections were found, the system may be separated with a known "good" assembly replacing one of the two major assemblies, say A. The "new" assembly would then be tested. If the "new" assembly passed the test, it would be assumed that the assembly A which was replaced is bad. The troubleshooting of the final assembly would then be complete and assembly A would be dispositioned for further troubleshooting. In this case it would be fairly easy to assign standard times to each task. The expected troubleshooting time could then be computed based on projected causes of failure. A more complicated example based on three major assemblies (A, B, C) will be discussed. In this case, there exists the possibility of performing three or four tasks before isolating the cause of failure.
1. Check connections
2. Replace assembly A and retest
3. Replace assembly B and retest
4. Replace assembly C and retest

Notice that it might safely be assumed that assembly C was the cause of failure if steps 1 thru 3 all resulted in test failures. On the other hand, the technician might be unwilling to assume anything at that point. For the sake of illustration, assume that a problem with the connections (I&A) is expected to be the cause of failure 20% of the time. Furthermore, assume the assemblies A, B, and C are expected to be the cause of failure 40%, 30%, and 10% of the time respectively. If we estimate the time required to perform each of the tasks 1, 2, 3, and 4 to be 10, 30, 30, and 30 minutes respectively, we can determine the average troubleshooting time in the following manner:

\[
\text{Average Troubleshooting Time} = 10 + 0.8(30) + 0.4(30) + 0.1(30)
\]

\[
= 49 \text{ minutes}
\]

Notice the tasks were assumed to be performed in the order which corresponded with the highest probability of isolation (except task 1), and task 4 was assumed to be performed if reached.

Other cases may be much complicated with the addition of intermittent failures. For example, an electronics assembly may fail while undergoing a test at a temperature
extreme. The most likely first step would be to again check for possible connection problems. If none are found, the unit may be tested at ambient temperature. If the unit fails the test, the troubleshooting procedure may take the form of the example above. If the unit passes the test at ambient temperature, it may be an indication of a temperature related failure of an intermittent nature. In this case the unit would probably be placed back into a temperature chamber for retest. The temperature would probably be increased gradually while monitoring the unit for signs of failure. Upon failure at the temperature extreme again, it would be confirmed that the unit had an intermittent, temperature related defect. The nature of temperature tests may add a new dimension to the method of determining an average troubleshooting time. The temperature gradations may provide knowledge of an unsuccessful assembly replacement in less time than it would take to be sure of the successful isolation. Assume that the electronics assembly consists of three circuit card assemblies as in the earlier example. The situation may be such that we can expect to see the system fail, if it is going to, in 30 minutes, while the test may continue for a full 60 minutes to assure no failure. The computation leading to the average troubleshooting time would necessarily take this into account. Table 2 on the following page illustrates how the assumptions of different times for verifying failure or success
<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Time Expended if Problem Solved</th>
<th>% of Problems Solved</th>
<th>Expected Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check connections</td>
<td>10</td>
<td>10%</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Test at ambient temp.</td>
<td>30</td>
<td>30%</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>Test at temp. extreme</td>
<td>30</td>
<td>10%</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>Replace assembly A and test at temp. extreme</td>
<td>60</td>
<td>15%</td>
<td>19.5</td>
</tr>
<tr>
<td>5</td>
<td>Replace assembly B and test at temp. extreme</td>
<td>60</td>
<td>15%</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>Replace assembly C and test at temp. extreme</td>
<td>60</td>
<td>15%</td>
<td>10.5</td>
</tr>
<tr>
<td>7</td>
<td>Request engineering assistance</td>
<td>10</td>
<td>5%</td>
<td>.5</td>
</tr>
</tbody>
</table>

Total Average Troubleshooting Time = 100.5 minutes
are taken into account in determining an average time. The first three tasks are assumed to require the same length of time whether or not the failure is isolated (problem is or is not solved). The next three tasks are assumed to require less time to verify that a problem still exists than to be sure it has been eliminated. The physical meaning is that the failure shows up at approximately the same temperature each time the unit is tested. If it does not occur, this may indicate the correct substitution has been made. It is very likely that the test would be run for an extended period of time and perhaps at greater temperature extremes to be sure the defect is not simply being overlooked, but is truly removed. The example does not include any provisions for multiple circuit card failures in establishing a standard time. The example is also simplified in the description of the troubleshooting procedure and assumes the use of engineering support (task 7) to help isolate defects that are not readily diagnosed.

The methods described in this section are by no means an exact science. In any particular situation the problem of quantifying troubleshooting activity durations may be handled by a variety of methods, including rough estimates of the average time or the addition of a fixed percentage of the test/retest time. These examples are included because it is believed they may provide some guidance to those who desire a similar approach.
It is important to note that the assumptions regarding the probable causes of failure were only assumptions and a potential cause of error. As mentioned earlier, the need to project ahead with limited historical data is often a labor estimators greatest handicap. The use of judgment in these cases is not wholly without merit. Knowledge of the physical make-up of assemblies and their complexities plays a big role in the assignment of numeric values to these probabilities. The distribution of failures (to lower level subassemblies) will be discussed further in the next chapter but also in the context of assuming values based on judgment. In the defense industry, designs are often changing which raises questions regarding the direct use of past experience in many instances. This report is directed towards estimating in a logical fashion rather than estimating using historical data.
CHAPTER V

ACTIVITY FREQUENCY

Role of Activity Frequency

The number of times a test (or troubleshooting) activity must be performed is an important consideration when preparing a labor estimate. The "frequency of occurrence" depends on both the characteristics of the test in terms of the probability of a unit passing, and the position of the test in the overall test flow which effects the number of retests sent to it from higher level failures.

Most tests are expected to fail some of the units which are submitted (tests which are performed solely to gather information are possible exceptions) or they would probably be eliminated as requirements in production. Given the anticipated frequency that each activity will be required in order to deliver an acceptable end product and the standard time consumed per activity, multiplication will provide the estimator with the standard time for each activity per end product delivered. The purpose of this chapter is to describe one method of predicting the number of times each troubleshooting and test/retest activity will be performed.
Tests Modeled as a Bernoulli Process

A test may be described as a trial which can have one or the other of just two possible outcomes, success or failure. The term success will be understood to indicate the unit passed the test and was accepted, while failure will indicate rejection.

As in many applications of probability theory to practical decision problems, assumptions are necessary to conduct analysis. The primary assumption of this analysis is that testing can be modeled as a Bernoulli process. In other words, the same probability of a particular test ending in a success is assigned to each future test regardless of the outcome of any future test. This assumption implies that the outcomes of tests are independent of each other and failures will not tend to occur in streaks. Since retests will not be distinguished from initial tests, a unit will be assigned the same probability of success regardless of whether or not it had been tested previously.

The estimator may be willing to approximate the operations of production testing with a Bernoulli model in many instances. Possible areas of concern may center around the assumption of retests and initial tests having the same probability of success or "test yield". Another concern may involve the independance of trials. Component parts which are produced in lots may tend to vary in their quality and failures may tend to follow failures more fre-
quently when certain lots are incorporated into the product. For now the validity of the model will be assumed.

Selection of Yields

Most of the controversy associated with production test estimates is generated over the selection of specific numeric values to represent the test yields. As mentioned previously, the labor estimator is often required to project these values for a relatively large number of future tests with little or no historical data. In these cases, where there is considerable uncertainty as to the long run value of the test yields, the estimator must rely almost wholly on subjective judgment. If there is a small amount of data concerning the results of tests of the first production units, the estimator may be justified in fixing these values as the lower limits of test yields.

It is important, however, to verify that the early yield data is not distorted because of faulty test equipment which is failing good units or passing bad units. Another possible cause for distortion could be the use of re-tests to collect data. These potential difficulties with using early production data highlight the need for consultation with persons knowledgable of the test activities (i.e., test engineers, reliability engineers, quality engineers, and manufacturing personnel).
In other cases, the estimator may have a considerable amount of "stabilized" yield data relative to the production quantity of units currently being estimated. In this case, the estimator may be concerned that the actual test yields experienced over the future fixed quantity of units may be significantly different from the established long-run values due to the inherent variability of the testing process. The risk of this "Bernoulli uncertainty" can be assessed by the use of tables for the Pascal cumulative distribution function (CDF). It is likely however, that the changing designs of weapon systems may inhibit the use of such risk assessment techniques by rendering the past data inapplicable. For additional information regarding the use of probability models of production systems and the assignment of probabilities based on experience (relative frequency) readers are urged to refer to the excellent text by Schlaifer (1959).

**Modeling Test Arrangements**

As mentioned in the previous section, risks can be assessed through the use of the Pascal CDF if the test yields, p, are known. For the remainder of this chapter the test yields will be assumed to be known. The Pascal distribution governs the number of Bernoulli trials, N, required to obtain a fixed number of successes, r. When assessing the risk in an estimate, the estimator will be in-
interested in determining the probability that the actual number of test/retests will exceed his projection.

The projection, for most estimates, will be the mean or average number of test/retests required to pass a given number of units. The expected value of the number of Bernoulli trials required to pass \( r \) units, \( E(N) \), with the probability of success, \( p \), can be computed with the following equation:

\[
E(N) = \frac{r}{p}
\]  

(1)

Equation (1) is applicable when calculating the number of tests and retests which are expected to be necessary at a single stand-alone test. A stand-alone test refers to the position of the test in the production flow. A stand-alone test receives no returns (or flowback) from failures at higher level tests. The final test of a product prior to delivery may be represented in this manner. Figure 4 illustrates the given situation. The expected number of troubleshooting activities required at this position due to failures of this test, \( E(NTS) \), can be computed using either method provided in Equation (2).

\[
E(NTS) = \frac{rq}{p} = \frac{r}{p} - r
\]  

(2)

where: \( q = \) the probability of the test resulting in a failure.

\[ = 1 - p \]
Figure 4. A Single Stand-alone Test Flow Diagram

Table 3 reflects the results which are expected when an initial quantity of 100 units \((r = 100)\) are required to successfully pass testing at a single stand-alone position with an 85% yield. As illustrated, out of each lot submitted 85% pass while 15% fail. The units which fail will re-

TABLE 3. RESULTS OF 100 UNITS THROUGH FIGURE 4, 85% YIELD

<table>
<thead>
<tr>
<th>Number of Units Submitted</th>
<th>Number of Units Passed</th>
<th>Number of Units Failed</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>85</td>
<td>15</td>
</tr>
<tr>
<td>15</td>
<td>12.75</td>
<td>2.25</td>
</tr>
<tr>
<td>2.25</td>
<td>1.91</td>
<td>.34</td>
</tr>
<tr>
<td>.34</td>
<td>.29</td>
<td>.05</td>
</tr>
<tr>
<td>.05</td>
<td>.04</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>117.64</td>
<td>99.99</td>
<td>17.65</td>
</tr>
</tbody>
</table>
quire troubleshooting and retests. The column totals for the number of units failed very closely approximates the results which would be obtained from Equations (1) and (2) with \( r = 100, p = .85, \) and \( q = .15. \)

The single stand-alone test is the basic model of a test flow network. Figure 5 illustrates a model of sequential tests performed on the same piece of hardware. An example of this situation is a unit submitted to several tests (e.g., pretest, vibration, functional) in a series. In many of these arrangements, a failure of any test in the series requires the unit to begin the series again, after troubleshooting and repair. Since all failures of the tests \( (T_1, T_2, \ldots, T_L) \) eventually begin again at the first test, \( T_1, \) and must pass each successive test without fail-

![Figure 5. Sequential Testing](image-url)
ure, $T_1$ must pass both the initial requirement of $r$ units plus any failures from the succeeding tests in the series again. Assuming no units are returned to the series from higher levels of hardware testing, the expected number of tests/retests which will be required by any test in the series except the last test, $T_L$, can be computed using Equation (3). The last test in the series requires the use of Equation (1) for computing the expected number of tests/retests.

\[ E(N_i) = \frac{r}{(p_i)(p_{i+1})(p_{i+2})... (p_L)} \]  

for $i = 1, 2, 3, ..., L-1$

where:

$E(N_i)$ = the expected number of tests/retests required of test $i$ to pass $r$ units

$p_i$ = the yield of the $i$th test

$L$ = the last test in the series

$r$ = the required number of successfully tested units

The expected number of troubleshooting activities required for any test in the series except the last test, $T_L$, can be determined by using Equation (4). The troubleshooting requirements for the last test will be determined from the application of Equation (2).
\[ E(NTS_i) = \frac{r}{\prod_{j=i}^{L} p_j} - \frac{r}{\prod_{j=i}^{L} p_j+1} \]  \hspace{1cm} (4)

for \( i = 1, 2, 3, \ldots, L-1 \)

where: \( E(NTS_i) \) = the expected number of troubleshooting activities required due to failures of test \( i \)

\( p_j \) = the yield of test \( j \)

The derivation of Equations (3) and (4) will be explained in the context of the example provided by Figure 6 which illustrates the given situation. In Figure 6, the activity blocks contain expressions representing the number of times each activity is expected to be required. As in previous figures, the dotted routing and activities represent troubleshooting and the solid lines indicate the normal production flow. Notice that the number of units tested and retested at each position includes the initial quantity, \( r \), and those units returned due to failures at the succeeding tests plus the retests generated by the particular test yield. The last test in this series, \( T_3 \), only retests those units which it fails. The second test, \( T_2 \), must perform the number of tests/retests necessary to pass the initial quantity of \( r \) units as well as those returned to it from the failures of the last test, \( T_3 \). These quantities are represented by expected values as seen in Equation (5). Actually, Equation (5) is used to find the expected value of the number of tests/retests and it uses the
Troubleshoot 1

Test 1

Test 2

Test 3

Figure 6. Illustration of a Three Test Series
the expected numbers of failures from previous tests in doing so. Equation (5) illustrates the computations with the number of the last test, \( L \), equal to 3 and the second test \((i=2)\) under consideration. Through substitution the consistency between Equations (5) and (3) is shown.

\[
E(N_2) = \frac{r}{p_2} + \frac{rq_3}{p_3} \frac{1}{p_2} 
\]

By substituting \(1 - p_3\) for \(q_3\), Equation (5) is shown to have the same result as Equation (3), with \(L = 3\) and \(i = 2\).

\[
E(N_2) = \frac{r}{p_2} + \frac{r(1-p_3)}{p_2 p_3} 
= \frac{r}{p_2} + \frac{r}{p_2 p_3} - \frac{r}{p_2} 
= \frac{r}{p_2 p_3}
\]

Test 1 in the example, must be performed enough times to pass the initial quantity of \(r\) units as well as the returns from test 2 and test 3. The data contained in Figure 6 can be used in an equation similar to Equation (5) and can also be shown to have the same result as Equation (3).

In the case of troubleshooting activities, Equation (6) illustrates the computations associated with determining the expected number of troubleshooting activities necessary to support the second test \((i=2)\) in the example. The number of troubleshooting activities depends not only on the number of units failed at the particular test, but
also on the number of failures which are obtained by re-testing returned units.

\[ E(NTS_2) = \frac{rq_2}{p_2} + \frac{rq_3}{p_3} \frac{q_2}{p_2} \]  \hspace{1cm} (6)

Again substituting \( 1 - p_i \) for \( q_i \), Equation (6) can be shown to have the same result as Equation (4), with \( L = 3 \) and \( i = 2 \).

\[ E(NTS_2) = \frac{r(1-p_2)}{p_2} + \frac{r(1-p_3)(1-p_2)}{p_3p_2} = \frac{r}{p_3p_2} - \frac{r}{p_3} \]

Similarly, test 1 in the series would send failures of the initial \( r \) units to troubleshooting plus any failures of units returned to test 1 from tests 2 or 3.

The series testing just described assumes that a failure of any test in the series will send the unit back to the first test. This is not always the case. Some tests that are on the same piece of hardware and are accomplished sequentially are testing totally unrelated characteristics. In these situations an item failing, say the third test, may be repaired and sent back only to the third test for retesting. The formulas provided can be altered to handle these particular situations, but they are often a cause of confusion. So far, the discussion has addressed
tests which do not receive returns from failures at higher level hardware testing. Consequently, the concept of distributing the cost of retesting failures of a final assembly test to the major (and minor) subassemblies has not been addressed. The next section will describe how the multinomial distribution may be employed to distribute failures of assemblies to lower level hardware for further troubleshooting and retests. The discussion will begin with the description of a mechanical method of computing the expected test/retest and troubleshooting requirements which can handle the distribution to lower level hardware and the disrupted series (as described above).

Multinomial Distribution of Failures

The previous examples illustrate the impact that the yield of a test as well as its position in the test flow has on the expected number of test/retest and troubleshooting activities. There was no allowance for further distribution to lower level hardware for troubleshooting and retests. The example presented in Figure 7 represents three subassemblies (A, B, and C) which make up a final assembly, F. As in Figure 6, the final assembly is submitted to three tests. The final assembly troubleshooting is depicted as being the same activity for any of the three test failures. The cause of any final assembly failure is expected to be subassembly A (S/A A) 20% of the time, S/A B
35% of the time, S/A C 25% of the time, and the integration and assembly effort associated with the final assembly 20% of the time. This distribution of failures to causes can be thought of as a multinomial distribution where:

\[
p(i,j) = \begin{cases} 
.20 & \text{for } i=F, j=A \\
.35 & \text{for } i=F, j=B \\
.25 & \text{for } i=F, j=C \\
.20 & \text{for } i=F, j=F 
\end{cases}
\]

and where \( p(i,j) \) indicates the probability that a failure (or a troubleshooting submittal) from test \( i \) is estimated to be caused by the hardware specified by \( j \).

Figure 7. Distribution of Failures to Lower Level Hardware
The expected number of units returned to a specific subassembly from a particular test can then be determined by using Equation (7).

\[ E(N_{i,j}) = p(i,j)E(NTS_i) \]  

\( (7) \)

where: 

\[ E(N_{i,j}) \] = the expected number of units sent to activity \( j \) from activity \( i \)

\[ p(i,j) \] = the probability of a unit being sent to activity \( j \) after failure at activity \( i \)

\[ E(NTS_i) \] = the expected number of units sent to troubleshooting from activity (test) \( i \).

In the example presented in Figure 7, the expected number of units returned to each subassembly from the final assembly can be calculated using the values indicated on the figure and Equations (2), (4), and (7). In this example it has been assumed that the same distribution will be followed for any failure occurring at the final assembly level tests. Therefore, the expected number of units sent to troubleshooting by each of the three final assembly tests may simply be summed up and then distributed. The values will be calculated on a per unit (final assembly) basis (i.e., \( r=1 \)).

First, the expected total number of units sent to troubleshooting by the final assembly tests will be calculated (for \( i \) summed from 1 to 3):

\[ E(NTS_i) = \frac{1}{(.85)(.95)(.98)} \quad r = .2637 \]
And then, the expected number of units returned to each of the subassemblies (S/A A, S/A B, and S/A C) from the final assembly testing can be computed using Equation (7).

\[
E(N_{i,j}) = \begin{cases} 
\cdot20(.2637) & \text{for } i=F, j=A \\
\cdot0527 & \\
\cdot35(.2637) & \text{for } i=F, j=B \\
\cdot0923 & \\
\cdot25(.2637) & \text{for } i=F, j=C \\
\cdot0659 & 
\end{cases}
\]

Failures of the final assembly tests which are attributed to the integration and assembly effort do not disposition units to lower level assemblies. There is no impact from integration and assembly failures on any test activities except the troubleshooting and retests of the final assembly. The impact of all failures of the final assembly testing was addressed in the previous example in terms of the series testing. The impact of returned units on the troubleshooting and retest activities of the sub-assemblies will now be discussed. In the same manner that the return of failures from a series of tests has an impact on the previous tests in the series, so do the returns to lower level hardware impact the number of times units must be troubleshoot and retested. The difference is that unsuccessful series tests result in return of 100% of the failures back to the previous tests in the series; whereas in lower level hardware flowback, the failures are usually attributed to just one of the next lower level pieces of
hardware resulting in less than 100% of the failures being returned to each subassembly. Beyond that distinction, the returns are treated in much the same way when calculating the impact on retest and troubleshooting activities.

For example, assume the test of subassembly A has a yield of 90% (as indicated in Figure 7). The expected number of tests/retests and troubleshooting activities required to provide 100 good units of S/A A can be calculated through the use of Equations (1) and (2). Consider this the "basic" expected requirements.

$$\text{Basic } E(N_A) = \frac{r}{p_A} = \frac{100}{.90} = 111.111$$

But, the subassembly is also expected to receive .0527 units due to failure at final assembly testing per final assembly delivered (or 5.27 units per 100 final assemblies delivered). The total number of test/retests expected to be required of subassembly A is computed by including the expected returns as indicated below:

$$\text{Total } E(N_i) = \frac{r}{p_i} + \frac{E(N_{k,i})}{p_i}$$

for k representing all higher level tests which may distribute failures to i

$$\text{Total } E(N_A) = \frac{r}{p_A} + \frac{E(N_{i,j})}{p_A}$$

$$= \frac{100}{.90} + \frac{5.27}{.90}$$

for i=F, j=A

$$= 116.97$$
The number of troubleshooting activities required at the subassembly A can be found in a similar manner. The difference is that all units sent back to S/A A as a result of a final assembly failure will be troubleshot and any subsequent failure of a retest will also require troubleshooting. The calculations below illustrate this.

Basic $E(NTS_A) = \frac{rq_A}{p_A} = \frac{100(.1)}{.9} = 11.111$

Total $E(NTS_A) = \frac{rq_A}{p_A} + \frac{E(N_{i,j})q_A}{p_A} + E(N_{i,j})$

$\quad = \frac{100(.1)}{.90} + \frac{5.27(.1)}{.90}$ for $i=F, j=A$

$\quad = 16.97$

The activity requirements expected for subassemblies B and C could be computed in the same manner. The effort associated with computing the requirements in a large test flow network will obviously require the dedication of a significant amount of resources. In the following section a technique will be described which may prove useful to the estimator in the bookkeeping aspects of the computations.

**Tabular Technique of Estimating Activity Frequencies**

From the previous example, it is apparent that a systematic method is necessary to calculate the expected numbers of test/retest and troubleshooting activities required for each test. A procedure involving the tabular (arrays)
placement of data will be presented in the following text. First, the general steps of the procedure are listed and then an example problem will be worked to illustrate the method. The steps are:

1. List each possible source of failure (either test or hardware) being considered across the top of the page to serve as column headings. Order the headings so that the higher level sources are to the left of any lower level sources.

2. List each receiver of returns (troubleshooting and test activities) down the left hand side of the page to serve as row headings.

3. List the yields of each source directly underneath or above each heading for reference, but out of the rows headed by receivers of failures.

4. Duplicate the empty matrix.

5. In the first matrix, underneath each source of failure, write the percent of failures which would be attributed to the various receivers in the row headed by each receiver description. This matrix will be known as the failure disposition matrix.

6. Using the second matrix, fill in the computed values representing the number of returns to each receiver from each source on a per-unit basis. The following procedure will be employed to perform the calculations required:
a. Take the inverse of the yield for the first (highest level) test.

b. Subtract one from the result to arrive at the number of failures expected at this test.

c. Multiply this value by the receiver percentages listed in the failure disposition matrix. Place the results in the corresponding locations in the second matrix.

d. For the remaining tests, add the numbers of units dispositioned from higher level failures (i.e., add the numbers in the row corresponding to the particular test under consideration) as listed to the left of the column under consideration in the second matrix.

e. Add one to the row total and divide by the test yield.

f. Subtract one plus the row total from the result to arrive at the number of failures and multiply the number of failures by the receiver percentages listed in the failure disposition matrix, and place the results in the corresponding locations in the second matrix.

g. Proceed with steps d, e, and f until the matrix is complete.

7. The total of each row in the second matrix (referred to as the cumulative return matrix) will provide
the estimator with the frequency of occurrence for each retest and troubleshooting activity listed in the receiver column on a per-unit basis. Adding one to the number of retests results in the total number of tests/retests per unit delivered for each activity.

The following example will illustrate the use of the data arrays. The illustration in Figure 8 represents a three level test flow diagram. The failure disposition matrix provided in Table 4 identifies the percentages of failures from each test that will impact the various test and troubleshooting activities and the test yields. The second matrix, the cumulative return matrix, is provided in Table 5. This matrix is used to accumulate the number of returns sent to each activity.

![Three Level Test Flow Diagram](image-url)

**Figure 8. Three Level Test Flow Diagram**
Table 4. Failure Disposition Matrix

<table>
<thead>
<tr>
<th>Sources Yields</th>
<th>F</th>
<th>A</th>
<th>B1</th>
<th>B2</th>
<th>BA</th>
<th>BB</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F T/S</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>.4</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A T/S</td>
<td>.4</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B1</td>
<td>.4</td>
<td>0</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B2</td>
<td>.4</td>
<td>0</td>
<td>1.0</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B T/S</td>
<td>.4</td>
<td>0</td>
<td>1.0</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BA</td>
<td>.2</td>
<td>0</td>
<td>.5</td>
<td>.3</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>BA T/S</td>
<td>.2</td>
<td>0</td>
<td>.5</td>
<td>.3</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>BB</td>
<td>.2</td>
<td>0</td>
<td>.5</td>
<td>.3</td>
<td>0</td>
<td>1.0</td>
</tr>
<tr>
<td>BB T/S</td>
<td>.2</td>
<td>0</td>
<td>.5</td>
<td>.3</td>
<td>0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Reference the failure disposition matrix in Table 4. The sources of failure are listed along the top with the highest level test at the left with decreasing levels to the right. In descending order along the left-hand side of the table are the activities which will be impacted by the returned units. The body of the matrix consists of the "from-to" probabilities as mentioned in the previous section (p(i,j)). For example, the failures of test F will be sent to F troubleshooting (F T/S) and back to test F.
(eventually) with probability = 1.0 as indicated in Table 4. The failures of test F will also be sent to troubleshooting for assemblies A and B with probabilities of .4 and .4 respectively. The remaining .2 of the failures at test F are assumed to be integration and assembly failures with no further disposition to lower level hardware required. The troubleshooting activities associated with assemblies A and B will also generate the need for retests and so the values are allocated to the tests as well.
Since assembly B is made up of two subassemblies, BA and BB, it may be necessary to further disposition the failures of test F between these lower level assemblies. Since 40% of the test F failures were allocated to assembly B, and it is not expected that a latent assembly and integration defect of assembly B caused the failure at test F, the entire 40% will be dispositioned between BA and BB at .20 each.

The failures of test A have no impact except on the troubleshooting and test activities associated with test A as reflected in Table 4.

Assembly B is submitted to two sequential tests. The failures of either test are sent to a common troubleshooting procedure identified in Table 4 as B T/S. In addition, all of the failures of test B1 are sent back for retests at B2 and B1 and the disposition is therefore indicated as 1.0 (100%). The failures of tests B1 and B2 must also be split between the integration and assembly effort and the subassemblies BA and BB. For illustration purposes, assume that test B2 is a thorough functional test and test B1 is a burn-in test of an extended operational duration. In this situation, it is very likely that the first test, B2 would screen most of the integration and assembly defects and that test B1 would result in failure only if a part wore out. The disposition matrix can reflect this by dispositioning only 60% of the B2 failures to the lower assemblies (30% each), while attributing 40% of the failures to
be caused by integration and assembly errors. Test Bl on the other hand, may have its failures dispositioned under the assumption that the integration and assembly defects have been corrected and therefore attribute all Bl failures to either subassembly BA or BB. In this case the disposition is split evenly at .5 each. The remaining tests, BA and BB, will only disposition failures to their own troubleshooting and test/retest activities.

The second matrix, the cumulative return matrix, is used to maintain the numeric "bookkeeping" of the procedure and is illustrated in Table 5. Following the procedure, the inverse of the first test yield (.9) is taken. One is subtracted from the result and the number of failures is then multiplied by the disposition matrix values under the column F. These results are placed in the corresponding locations in the cumulative return matrix. This procedure dispositions the failures of test F to the activities which will be impacted by them. Next, the failures associated with test A will be computed by adding the number of units received from test F (.0444) to 1 and dividing by the test yield (.85). The result is 1.2287. The number 1.0444 is subtracted from 1.2287 to arrive at .1843 failures at test A (per delivered unit) to be dispositioned. Notice that 1.0444, not 1.0, had to be subtracted from the total of 1.2287. The procedure works by computing the number of returns to each activity and storing that value in each lo-
cation of the cumulative return matrix. Since each return has the potential to fail a retest and generate additional failures to be dispositioned, the total number of units each test activity will be required to pass must be considered when computing the total number of failures, test/retest, and troubleshooting activities.

Following the same procedure, the matrix is completed. The numbers of troubleshooting and retests expected to be necessary can be computed by adding the totals in each row. The result is a per-unit estimate of the expected frequency of occurrence for each activity. If the initial tests and the retests are the same for any activity, the total number of test/retests is arrived at by simply adding one to the row total.

This method provides a procedure for handling the computational bookkeeping associated with estimating the activity frequencies for production testing. Among the virtues of this method is the ability to disposition failures which arrive at a single troubleshooting station differently depending upon where the original failure occurred. The ability to maintain this "memory" into the estimating procedure is particularly useful in actual analysis. In general, the estimator can incorporate a great deal of information pertaining to the stresses individual assemblies or parts may be subjected to during any given test into the failure disposition matrix.
In actual applications, the first step in estimating the test requirements would be to obtain or generate a test flow diagram for every deliverable end item. If the failure dispositions are not feasible or desired at the individual test level, an alternative may be to create a model which combines tests and uses hardware as sources of failures. That is, if a series of tests on a certain piece of hardware will all use the same distribution of failures and if the requirements for the individual test and troubleshooting activities are going to be computed outside of the matrix, the size of the data arrays may be greatly reduced. In any case, considerable effort will be necessary to insure that the dispositions are incorporated into the model correctly.

Another way of reducing the size of the arrays is by splitting them into more than one array. For instance, if the final assembly consisted of two major subassemblies which each contained four or five levels of hardware, an array could be created for each of the two major assemblies. The final assembly test(s) would be listed first in each matrix, but each array would then consist only of the tests and troubleshooting activities which would directly impact the particular assembly and its component parts. In this way, the size of one big matrix would be reduced into two smaller matrices by eliminating many of the blank spaces.
When the total expected numbers of tests/retests and troubleshooting activities are computed on a per-unit basis, the total test labor estimate can be completed by combining these values with the extended standard times and other realization factors. The next chapter provides a brief overview of the final estimating steps and considerations.
CHAPTER VI
COMPLETING THE ESTIMATE

The final steps of the estimating procedure involve the application of performance factors and contingency factors to the results obtained in the previous chapters. Performance factors are generally split into two categories, starting performances and learning curves.

Starting performances are estimates of the ratio of the standard time for an activity to the actual time spent performing the activity for the first time, not including any retest time. This factor is expressed as a percent and is used as a divisor into the standard time to arrive at a theoretical first unit cost.

The learning curve (also known as the manufacturing progress curve) is used to reflect the anticipated improvement trend in labor requirements as successive units are produced. The progress curve is identified by the percentage reduction in unit cost as the quantity of units produced doubles. If an 80% curve is selected for use, it would imply that the second unit produced is anticipated to cost 80% of the first unit cost, the fourth unit produced is anticipated to cost 80% of the second unit produced, and so on. The basic learning curve equation is based on the following unit cost equation:
\[ Y = A x^b \quad \text{for } x = 1, 2, 3, \ldots \quad (8) \]

where:
- \( Y \) = the cost (usually in hours) of the \( x \)th unit produced
- \( A \) = the (theoretical) first unit cost
- \( x \) = the unit number
- \( b \) = the exponent of the learning curve slope

\[ \ln (\text{slope}) = \frac{\ln (2)}{b} \]

where: the slope refers to the \% reduction anticipated as the quantity of units produced doubles, expressed as a decimal

When the total cost of a quantity of units is being estimated a very useful learning curve tool is the cumulative factor. A cumulative factor is generated for each unit number by summing the unit curve factors (obtained from Equation (8) when \( A = 1 \)) from unit number 1 through and including the unit number under consideration. The cumulative factors obtained by this method are available in tables or book form. There are also approximation formulas available which are useful for quantities of twenty or more units.

The use of the cumulative learning curve factor is to multiply the "realized" first unit cost (hours) times the factor to calculate the total cost of the number of units produced. The cumulative factor represents the total improvement over the first unit cost which is expected
to be attained for a given quantity of units. For example, the cumulative factor for unit 100 on a 90% curve is 58.14102. Given the first unit cost, the total cost for the first 100 units produced would be estimated, on a 90% curve, to be 58.14102 times that first unit cost. If the estimate is for a follow on quantity of units, the total estimated cost is determined by subtracting the cumulative factor of the previously built quantity from the cumulative factor for the sum of the previous built and the current quantity being estimated. The cumulative factor for 50 units on a 90% curve is 32.141955. If the quantity under consideration was 50 units following the 50 previously built units, the cumulative factor to be used is:

\[ 58.14102 - 32.141955 = 25.99906 \]

Readers unfamiliar with learning curve theory or interested in fitting curves to historical data may refer to Conway and Schultz (1959), Malstrom (1981), and Karger and Hancock (1982). As noted earlier, the most important decisions that the estimator must make involves the selection of the realization or "conditioning" factors.

There are several considerations that an estimator should evaluate when making selections. A few causes for concern will be mentioned.

1. **Manual versus machine time**: In cases where an operator is tending an automatic process (e.g., automatic test equipment) of long durations, the estimator may wish
to consider separating the times by manual or machine controlled elements and then applying separate starting performances and learning curves based on this distinction. It may be reasonable to expect manually controlled operations to have a lower starting performance and a steeper (lower percentage) learning curve than machine controlled operations. This distinction may be a good reason for estimating troubleshooting and test/retest activities on different curves and with different starting performances. The first troubleshooting procedure performed will almost certainly require a great deal of time because of the uncertainty associated with the process. However, as the technician becomes familiar with the hardware, the diagnostic time should decrease rapidly. The troubleshooting activities might therefore be estimated using a lower starting performance than test/retest activities and also a steeper learning curve.

2. Number of units versus number of activities: The previous chapter illustrated the computations of a realization factor which represents the number of times an activity is expected to be performed in order to successfully complete the testing of a unit. For learning curve applications, the estimator must choose to regard either the number of units produced or the number of activities performed as the "counter". For example, if a test activity is expected to be performed 1.35 times, on the average, to
deliver a final product, should the cumulative factor at unit 135 be selected to cost the production of 100 units? If the answer is affirmative, the extended standard per activity would be divided by the starting performance and the result extended to total cost by multiplication by the cumulative factor obtained at unit 135. Another problem situation is illustrated by the troubleshooting activities. If a troubleshooting activity is expected at a rate of .35 times on the average to deliver a final product, should the cumulative learning curve factor be selected at unit 35 or 100 if the estimate was for 100 final products? The selection of either approach in these cases will depend upon the viewpoint of the estimator. If the frequency of occurrence of activities is thought of as a realization factor, the estimator would more than likely use the quantity of final products delivered as the learning curve counter. In general practice, it is customary to select the learning curve factor for the number of good units required. In this respect, the retests and troubleshooting activities frequencies would be considered realization factors and multiplied by the standard times for each activity, conditioned by the starting performance and any contingency factors, and then multiplied by the cumulative factor obtained for the number of final products delivered.

3. Continual learning versus curve flattening: In cases involving automatic equipment, the estimator may wish
to stop the decreasing function of the learning curve at or before the estimated unit cost reaches the standard time. This approach is known as "flattening" the curve or using a "dog-leg" curve. The reasoning to this approach is that the progress function will not continue indefinitely, but there is a real limit which inhibits further cost (hours) reduction on a per-unit basis. The point at which an estimator stops further reduction on the unit curve is generally based on some predetermined maximum performance.

Equation (9) provides the basis for computing the point where the unit learning curve is flattened, the cross over quantity. The cross-over quantity is defined as the last unit number which can be estimated on the curve without exceeding the predetermined maximum performance (MP).

\[
\text{Cross-over Quantity} = \text{Integer } \text{Antilog} \left( \frac{\log(SP/MP)}{b} \right) \tag{9}
\]

where:  
- Integer means to truncate the result of any decimal  
- SP = starting performance (decimal)  
- b = exponent of the learning curve slope  
- MP = maximum performance (decimal)

Given a fixed maximum performance which flattens the unit curve at the cross-over quantity, the total hours may be determined by one of the following methods depending on the position of the lot under consideration on the learning curve.
a. If the cross-over quantity is less than or to the previously built quantity of units, the entire quantity under consideration would be estimated at the maximum performance.

b. If the cross-over quantity is greater than the sum of the previous built and the current quantity, the entire quantity under consideration would be estimated on the learning curve.

c. If the cross-over quantity is within the lot currently being considered, the units up to and including the cross-over quantity would be estimated on the learning curve, and the remaining units would be estimated at the maximum performance.

In summary, the completion of the labor estimate of test touch labor requirements consists of the following steps:

1. For each test and troubleshooting activity, multiply the extended standard time by the factors which represent the expected frequency of occurrence for each activity. The result will be a standard time for each activity on a per end product delivered basis.

2. Apply any contingency factor which may be appropriate to the standard time per end product delivered.
3. Divide the result by the starting performance and multiply the "realized" first unit cost by the appropriate learning curve cumulative factor, divide by the appropriate maximum performance, or perform some combination of calculations as described in the previous paragraphs.

The result of this effort will be an estimate of the labor required to produce a given quantity of end products. The accuracy of the estimate depends not only on the proper application of estimating techniques, but also on the judicious selection of factors to represent test yields, failure dispositions, starting performances, and learning curves. This paper has not covered any of the methods which are used in arriving at the particular values and has instead described the use of the factors once obtained. Although many of the factors are based on subjective judgment in practice, the estimator should make every attempt to quantify historical data in order to provide a realistic and accurate estimate.
CHAPTER VII
SUMMARY AND CONCLUSIONS

This paper deals with the problem of estimating labor requirements for production testing activities. A major concern in such an estimate is the adequate provision for duplicate testing activities (retests) and the non-standard activity necessary to isolate the cause of test failures (troubleshooting). A common approach to this problem is the multiplication of the standard time to perform a test by an all encompassing factor intended to provide the additional labor hours for both retesting and troubleshooting. The basis of the factors used in this manner are often guesstimates which are difficult to assess and justify.

This paper has attempted to provide an alternative approach to estimating production test labor requirements. The approach is basically to identify test/retest and troubleshooting activities as discrete tasks and quantify the activity durations of each task. Next, a major concern in completing the estimate is the prediction of the number of failures expected at each test in order to deliver a fixed quantity of end products. Failures impact the production testing activities in the form of troubleshooting activities required to isolate the defects for rework, and retest activities necessary to assure the products integrity after
repair. Naturally, the more failures experienced, the more test/retest and troubleshooting activities will be necessary. The determination of the expected number of times each activity will be required (activity frequency) to deliver a fixed quantity of end products can be accomplished by modeling the test flow network with test yields and failure dispositions. This paper illustrates how the position of a test in the production flow, as well as the test yield, can be a major factor in determining the number of times each of the activities is expected to be required. The example problems in the section "Multinomial Distribution of Failures" in chapter five and the section "Modeling Test Arrangements" in the same chapter illustrates the effect of a test position on activity frequency.

The failures of tests also have an impact on manufacturing labor in the form of rework requirements. Given the predictions of test failures and troubleshooting requirements, the estimator could use the data to arrive at a rework estimate which is not necessarily a guesstimate percentage multiplier and is tied to the production test estimate. An average rework time could be estimated in a manner similar to the examples of estimating an average troubleshooting time provided in chapter four in the section titled "Estimating Troubleshooting Times". Once an average rework time is estimated, the frequency of rework activities could be estimated in a manner similar to the
predictions of troubleshooting activity frequencies, but also including the impact of quality inspection rejects. The average rework activity time could then be multiplied by the expected activity frequency per end product delivered to arrive at an estimated rework standard per end product delivered. This result could then be converted into a percentage multiplier (of the base manufacturing standard) or estimated separately through the application of unique starting performances and learning curves in a manner similar to the approach described for estimating troubleshooting labor.

The estimating techniques described in this paper can be quite time consuming when applied to real world situations involving many tests. The application of standard times requires a great deal of effort and is without a doubt the single most important task in any labor estimate. The development of a test flow model which identifies the flowback of test failure may also require a great deal of effort in researching and consulting. The benefit of performing these tasks is an estimate which is based on a logical procedure and easily assessed and defended.

The techniques for calculating test labor also apply to the companion activity of estimating test equipment requirements. Test equipment is generally the most expensive type of production tooling which is dedicated to a particular product. For this reason, there is a great deal of
emphasis placed on determining the proper quantities of test sets necessary to meet the production rate requirements of delivered end products per month without compromising either the cost effectiveness of the weapon system or the competitive position of the contractor. The use of the test flow model as described in this paper, with proper validation, could serve as a useful tool when determining the test equipment requirements. There are several additional considerations that must be addressed when determining the quantities of test sets necessary to maintain production at a specified rate. Among the considerations are adequate provisions for equipment calibration, reliability, and downtime. To address these problems, the estimator may quantify equipment availability with estimated derating factors or by methods used by maintainability engineers including the Mean Time Between Failures (MTBF), and the Mean Time To Repair (MTTR). The estimator may also be inclined to include consideration of the equipment utilization based on some form of queueing analysis.

Obviously, some of the additional considerations mentioned for estimating equipment requirements could have an impact on the touch labor requirements. For example, test equipment which fails to function satisfactorily will probably waste some production test labor, as will the idle time caused by the lack of hardware to test and the lack of fill in work. These considerations may or may not have a
significant impact on the labor requirements for any given production facility. The estimator may provide allowance for equipment reliability in the test yields or provide a contingency factor to account for expected labor losses. Allowances for idle time caused by production delays should technically be provided for in the Personal, Fatigue, and Delay (PF&D) allowance, although additional allowances may be justifiable. The calculation of any additional allowances may be difficult to perform and costly to validate, as well as possibly reflecting a temporary situation which may be easily corrected. Often such additional allowances are (consciously or unconsciously) included in the performance factors employed in labor estimates.

Another technique which may be used to model the test flow network is GERT (Graphical Evaluation and Review Technique). The technique provides the possibility of obtaining more information on the test flow network (e.g., variances) while providing a closed form solution to the problem of determining the activity frequencies. For additional information on the applications of GERT, readers are referred to the three part series of articles by Pritsker and Happ (1966), and Pritsker and Whitehouse (1966) and (1969).

Since the test flow model described in this paper is based on the failed units having a "memory" of where the failure occurred for disposition purposes, it is likely that a simulation model would prove useful. In fact, sim-
ulation is recommended as a worthwhile endeavor when the production is likely to continue over several years. Although the initial programming, verification, and validation expenses may appear prohibitive, the ability to try out new arrangements in the test flow and evaluate the effects quickly is a valuable asset. Additionally, a properly designed simulation model may be expanded to incorporate the fabrication, assembly, and rework operations in order to provide a complete view of the production system. The simulation model may be used to evaluate production recovery plans in the event a project falls behind schedule. The model of the system could be preloaded with units to reflect the current state and the exercised with various resource assignments to determine the most feasible of a set of alternative recovery plans. If a simulation model is to be developed, the approach described in this paper of predicting test/retest and troubleshooting requirements would be useful in verifying the simulation model. The assignment of probabilities would likely be altered to reflect the different views of the system (i.e., the simulation model may distribute failures based on a total of 100% at each troubleshooting station, whereas the tabular method distributes a total of 100% only from the originating station). The simulation model could be verified by inputting a fixed quantity of units, r, and running the simulation until the units are all out of the system. The number of
test/retest and troubleshooting activities counted by the simulation model could then be compared to the values obtained by the application of the technique described in this paper for the same quantity of units, r. Allowing for some purely random variations, the two methods should result in similar values.

Probably the most useful future endeavor would be the development of a general micro-computer program which could be used as a substitute for the manual calculations. With such a program, the sensitivity of the estimate to variations in the test yields and failure dispositions could be readily assessed providing a valuable tool during the proposal and contract negotiation stages. The procedure described in this paper in the section titled "Tabular Technique of Estimating Activity Frequencies" appears to conform very well for use in one of the micro-computer spreadsheet programs. The addition of a "front-end" program to interact with the user by prompting and setting up the row and columns based on responses may result in a useful estimating tool which can be easily set-up, exercised, and altered upon demand.
REFERENCES


