

# FRACAS Improves Production Bottom Line

Eduardo V. Cota , Raytheon Missile Systems

Louis Gullo, Raytheon Missile Systems

Key Words: FRACAS, Cost Savings, Long-term Corrective Action, Short-Term Directives, Short-Term Corrective Actions

## SUMMARY & CONCLUSIONS

Failure Reporting Analysis and Corrective Action System (FRACAS) process and procedures provide long-term corrective actions and short-term directives that reduce the cost of rework, increase production throughput and reduce lost material. Each failure mode adds a Failure Period (FP) to the normal build operation process cost, involving cost for troubleshooting and rework activities, which increases the Unit Production Costs (UPC). Long-term corrective actions eliminate failure modes, which reduce their inter-arrival rate, thereby reducing the number of tested units that deviate from the normal production process flow and enter an FP. Short-term directives characterize failure modes in such a manner as to provide a deterministic set of troubleshooting and rework procedures to restore the Unit Under Test (UUT) to performance acceptability and therefore reduce the FP cost and the number of tested units that enter a troubleshooting backlog.

## 1 INTRODUCTION

This paper identifies a set of Failure Reporting Analysis and Corrective Action System (FRACAS) processes and procedures and describes how they improve the bottom line success of a high rate production line. The bottom line is improved by reducing rework costs, reducing lost material, and increasing production throughput. Unit Production Cost (UPC) is the cost associated with production divided by the number of unit's produced<sup>1</sup>. Reducing the UPC of the produced item increases production bottom line, profit. Any increase to production cost to produce a fixed number of units will reduce producer profit.

### 1.1 Problem Description

During high rate production, test failures that are not quickly troubleshot to determine root cause are disassembled and are removed from the production line for further troubleshooting at a later time. If the problem is intermittent, then the unit may be moved forward in the production flow with a number of successful retests without realizing the underlying failure mode and risks. These intermittent failure modes may take excessive amounts of test technician time and resources to troubleshoot, diverting time and resources for testing other production units. Because of the high rate of production, units that are pulled off for troubleshooting build

up quickly, and eventually create a large troubleshooting backlog. This troubleshooting backlog is referred throughout this paper as a 'bone pile'. Bone piles usually contain failed units as well as good units. The bone pile accumulates material inventory that cannot be sold and shipped without material disposition, which results in increased cost due to material losses. Units in the bone pile may be scrapped if troubleshooting is unsuccessful at determining a root cause and the limitation of troubleshooting activities has been reached. A method to quickly disposition units in the bone pile must be developed before severe bottom line impacts are felt across the organization.

### 1.2 Problem Example

Consider a widget production line that has two major assembly process operations A and B (Op A and Op B). During normal operations of the widget production line, the process flow should follow Figure 1. The baseline UPC of each widget is based on this build flow, which is the cost of performing Op A and Op B. The baseline UPC is the expected cost planned for the widget production line. This cost plan itemizes the cost of performing Op A and Op B.



Figure 1 Normal Build Operation Process Flow

When test failures occur during normal operations, the production process flow changes as shown in Figure 2.

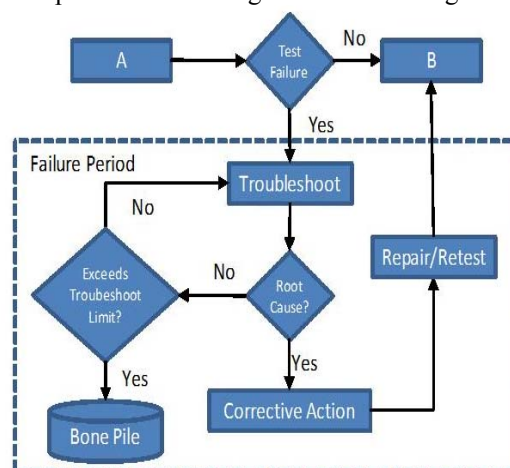


Figure 2 FP Operation Process Flow

The process identified in the dotted line in Figure 2 is the FP as referenced throughout this paper. The FP deviates from normal operation flow, adding additional tasks, which adds cost and time to production of a widget. The more time spent in the FP or the increase number of cycles through the FP process loop, the larger increase in widget UPC, the smaller the profit from producing widgets. During failure troubleshooting, a single failure mode can be troubleshot repeatedly for multiple widgets until the failure mode is root caused and corrected. When a single failure mode affects multiple widgets, the increased UPC is multiplied by the number of widgets affected. If corrective action involves resolution of a systemic failure mechanism, all units in inventory and Work In Process (WIP) may be purged or reworked regardless whether they were previously tested and passed. UPC is increased as the number of widgets produced declines due to material lost during FP troubleshooting, rework, repair and retesting, and the time spent during troubleshooting, rework, repair and retesting.

## 2 FRACAS DEFINITION AND PROCEDURES

### 2.1 FRACAS

FRACAS is a closed loop failure reporting, analysis and corrective action system. The FP is composed of FRACAS processes, which characterize and isolate failure modes to a particular root cause. The troubleshooting operation includes efforts to determine the root cause of the failure. This operation can be repeated multiple times until the failure is corrected, allowing the failed unit can continue in the production flow or until it is moved to a bone pile for scrap or more detailed troubleshooting. FRACAS processes and procedures ensure failure modes are understood and resolved with corrective actions to remove the failure mode or put in place fixes that will reduce the failure mode consequence and/or likelihood. Applying corrective actions reduces FP cost and provides deterministic troubleshooting directives that will reduce the number of failure units that could potentially enter the bone pile.

An effective FRACAS will also emphasize the ability to identify and track failure trends for appropriate corrective action<sup>2</sup>. FRACAS reduces the UPC by improving the effectiveness and efficiency of troubleshooting activities with the data for value-added decisions. During FRACAS efforts, failure modes are characterized and assigned corrective actions were applicable. Per MIL-HDBK-338, corrective action is a *documented design, process, procedure, or materials change implemented and validated to correct the cause of failure or design deficiency*<sup>3</sup>. Corrective action is assigned as either short-term corrective action or long-term corrective action. Short-term corrective action is triggered to fix the failed unit(s). When only short-term corrective action is taken, root cause is not addressed and removed therefore the cost of producing widgets increases by the cost to fix failed widgets. Therefore, long-term corrective action is triggered to remove the failure root cause and ensure additional efforts are not added to the normal production flow.

### 2.2 Long-term Corrective Actions

Long-term corrective action is action taken when the failure mode is characterized and root cause determined. This type of corrective action is implemented as either design changes or design process updates and are developed to resolve systemic failure conditions. These conditions follow patterns of occurrences and have long-term effects. Systemic failures may exhibit intermittent symptoms and occur very infrequently, surfacing only when exercised by certain types of stimuli or stresses. Long-term corrective actions may take a long time to verify effective implementation of the corrective action to ensure no further recurrences of the failure mode are possible.

### 2.3 Short-term Corrective Actions

Short-term corrective action is action taken to remedy the failed widget and improve its performance to an acceptable condition or ensure that the failure mode is not realized prior to implementing long-term corrective action. Short-term corrective action may include the creation of a short-term directive, which identifies the appropriate amount of troubleshooting and Repair and Rework (R&R) needed to get the failed widget into an acceptable condition. The short-term directive is applicable until a long-term corrective action is implemented.

### 2.4 Short-Term Directives

Short-term directives are created for failure modes whose root causes are known, but further effort is required to determine or implement long-term corrective action that will be a positive impact to the widget reliability and a long lasting improvement in the widget functionality. As parametric data is collected, analyzed and a trend is identified to characterize the failure mode, the short-term directive is documented to detail the failure mode characteristics and to provide a list of the tasks to enable the appropriate level of R&R to verify the failed widgets acceptability. The life of a short-term directive depends on the time to implement long-term corrective actions or yield monitors that would trigger a long-term corrective action investigation to eliminate the failure mode. Yield monitors are part of a Statistical Process Control (SPC) system to track attribute data at a particular inspection or test operation in the production flow and determine when the unit pass/fail data exceeds the predefined control or specification limits. As stated earlier, short-term directives do not remove the failure root cause. But it ensures that failures not root caused to the design, for example a test setup failure, does not cause the widget to enter the bone pile or go through extensive troubleshooting.

### 2.5 CA Impact

Each category of corrective action provides a cost savings. Long-term corrective action removes the failure mode, which reduces the failure inter-arrival rate. Because Long-term corrective action reduces the inter-arrival time of failures, it reduces the number of failures that go through the

FP. The reduced number of widgets that experience a FP reduces the widget UPC, because each FP requires time and resources which results in additional cost to the widget operations bottom line.

Short-term directives identify an acceptable work around that reduce the amount of time and resources needed to characterize the failure mode and R&R the failed widget to an acceptable condition. Reducing the amount of time and resources needed during the FP reduces the rework costs of producing widgets as well as reducing the number of widgets that enter the bone pile.

Both long-term corrective action and short-term directives increase the throughput of widget production. Long-term corrective action reduces the inter-arrival time of failures, which yield less test failures, which means increase successes. The increase success rate results in a greater number of widget produced. Short-term directives are known issues that have a known set of rework and retest to verify the failed widget acceptability, which provides a known finishing point to the FP. Having a known end to the FP allows widget operations to plan R&R efforts, which will keep the failed widget from entering a failure investigation bone pile, thus increasing widget throughput. Widgets that enter a bone pile increase the time and materials needed to produce the amount of widgets needed to meet the customer's purchase order, which increases UPC and decreases profit.

### 3 DEMONSTRATING FRACAS SAVINGS

#### 3.1 Cost of FRACAS

As shown in Figure 1, a widget must go through widget operations A and B. Each widget operation has its own cost to complete the operation, and therefore, in a normal operation flow the cost it takes to produce a single widget is the sum of Op A and Op B costs. Normal operation costs are deterministic and operational cost variations are ideally minimized.

$$\text{Normal Op cost} = \text{Op A cost} + \text{Op B cost} \quad (1)$$

Where:

*Op A cost* is a real number representing the cost of performing operation A.

*Op B cost* is a real number representing the cost of performing operation B.

Throughout this paper, as an example, the widget factory is expected to produce 100 widgets per it current contract. Recently the widget factory experiences a Test Equipment (TE) failure. The TE failure occurs a rate of one (1) failure per every hundred (100) widgets produced. When the TE failure occurs, the cost to produce a widget increases by the cost it takes to complete the processes identified in the FP, identified by the dotted line area in Figure 2. At the beginning of FRACAS implementation, the process costs of the FP are not deterministic and may include large variances, because the initial widget failures have not been root caused to the TE.

To show how FRACAS activities identified in this paper reduce the FP cost, FP cost is shown in the following equation.

$$\text{FP cost} = \lambda(\text{TS cost} + \mu(\text{CA cost}) + \text{R\&R cost}) \quad (2)$$

where:

$\lambda$  is a real number between 0 and 1, representing the inter-arrival rate of test failures for all units in production

*TS cost* is a real number representing the cost of troubleshooting the failure

$\mu$  is a real number between 0 and 1, representing the probability of finding a root cause of a failure

*CA cost* is a real number representing the cost associated with identifying and assigning corrective action

*R&R cost* is a real number representing the cost to Repair and Retest (R&R) the widget and verify its acceptability

Limits are applied to the equation to show how FRACAS activities reduce the FP cost. FP cost includes the cost to troubleshoot the failure, implement corrective action and R&R the failure source. FP costs are not deterministic, because failures are random and not all failures can be root caused to a single failure source. In some instances, failures cannot be root caused and the amount of troubleshooting performed will send the failed unit to a troubleshooting bone pile. The probability of identifying the failure mode root cause is denoted by  $\mu$  for this paper. FP is the sum of troubleshoot cost, corrective action cost multiplied by  $\mu$  and the cost to repair and/or replace the failure source, multiplied by a expected failure inter-arrival time which is denoted by  $\lambda$ .

#### 3.2 Troubleshooting Savings

FRACAS reduces the FP increase to UPC by improving the effectiveness and efficiency of troubleshooting activities with the data for value-added decisions. This increase is achieved by generating a controlled set of process and procedures for troubleshooting. Controlling the set of process and procedures provide a consistent execution of troubleshooting failures, reducing the troubleshooting cost. This is done by removing cost associated with generating failure unique process and procedures.

$$\lim_{\text{TS cost} \rightarrow 0} \text{FP cost} = (\mu(\text{CA cost}) + \text{R\&R cost}) \quad (3)$$

In regards to the TE failure example, initial TS determined that the widget failures are rooted caused by a defect in the TE Software (SW). Initial TS resulted in 100 hours of engineering effort. It was determined that a TE SW change will remove the SW defect, but cannot be implemented until after the all current widget orders have been built. As a result of engineering efforts, a short-term directive can be created, which describes parametric data and additional testing needed to isolate the failure to a TE SW failures and verify the widgets acceptability. Short-term directive efforts require 20 hours of non-engineering TS cost. As shown in equation 3, applying consistent troubleshooting efforts (i.e. short-term directives) results in a reduced TS cost.

#### 3.3 Long-term Corrective Action Savings

When long-term corrective action is applied,  $\lambda$  is reduced and the number of the widgets that experiences a FP is reduced.

$$\lim_{\lambda \rightarrow 0} \text{FP cost} = (\text{TS cost} + \mu(\text{CA cost}) + \text{R\&R cost}) = 0 \quad (4)$$

In regards to the TE SW example, failures arrive at a rate of one (1) failure per hundred (100) widgets produced. When

producing widgets, three (3) failure modes were identified, including the TE SW failure mode. The three (3) failure modes arrive at a rate of five (5) failures per hundred (100) widgets produced. When the TS SW change is implemented, the rate of widget failures is reduced to a rate of four (4) failures per hundred (100) widgets produced. As shown in equation 4, implementing long-term corrective action reduces the FP cost because the inter-arrival rate of widget failures is reduced to four (4) out of every hundred (100) widgets produced.

### 3.4 Short Term Directive Savings

When FRACAS includes short-term directives, the FP is reduced by circumventing repetitive troubleshooting and the CA period. The short-term directive FP process flow changes as shown in Figure 3.

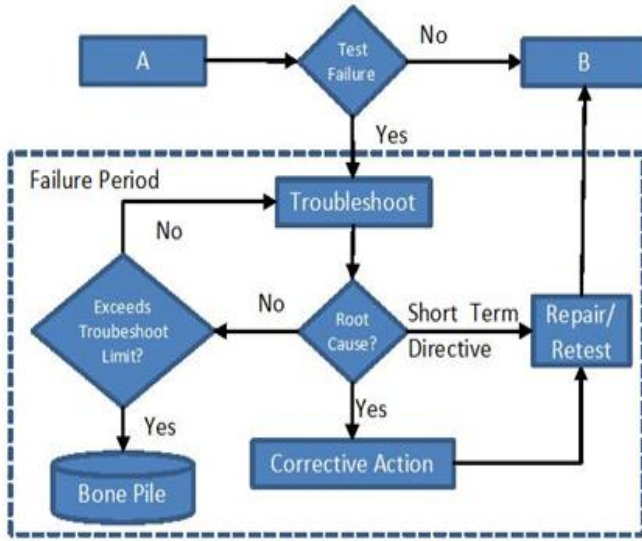


Figure 3 Short-Term Directive Operational Process Flow

Given short-term directives, failure modes are known and characterized in such a manner as to provide the widget operators troubleshooting procedures. Applying the short-term directive provides a set of R&R that is needed to set the failed widget to an acceptable condition. Given that a failure has a short-term directive, the FP cost then becomes the sum of the troubleshoot cost and R&R cost, multiplied by the inter-arrival time  $\lambda$ .

$$FP \text{ cost} = \lambda(TS \text{ cost} + R\&R \text{ cost}) \quad (5)$$

Therefore adding short-term directives to the FRACAS plan will reduce the FP cost by removing  $\mu$  (corrective action cost), thus reducing the cost and schedule of a widget failure.

As described in section 3.2, the TS cost is reduced by short-term directives, because they provide a detailed description of failure characteristics to isolate a failure to TE SW. Because the short-term directive also includes a listing of R&R and pending long-term corrective actions (i.e. TE SW change), efforts to identify and implement corrective action is removed from the FP, as proven in equation 5.

### 3.5 Using Short-Term Directives to Reduce Bone Pile

As noted earlier short-term directives can also reduce the number of widgets that go into a bone pile. To show this, we first determine how to estimate how many widgets go into a bone pile without short-term directives. The expected number of widgets in a bone pile is equal to the expected number of failures multiplied by the expected number of failure without a root cause raised by troubleshooting threshold limit and multiplied by the total number of widgets that are produced.

$$N = \lambda * (1-\mu-p)^r * M \quad (6)$$

Where:

**N** is a positive integer from 0 to M representing the number of units that enter the bone pile

**M** is a positive integer representing the total number of widgets produced

**r** is a positive integer representing the number of times a failed unit is troubleshooted before entering the bone pile

**(1-μ-p)<sup>r</sup>** is the percentage of failed units that will enter the bone pile

**p** is a real number between 0 and 1, representing the percentage of failures that have a short-term directive

As short-term directives are generated,  $(1-\mu-p)$  becomes smaller because  $p$  increases. As  $p$  increases, there are less failures requiring effort to identify root cause. As a result, less widget failures enter a bone pile.

$$\lim_{p \rightarrow 1} N = \lambda * (1-\mu-p)^r * M = 0 \quad (7)$$

Finally, using the TE SW example, initially failures arrive at a rate of five (5) failures for every hundred (100) widget produced. Initially, all failure modes do not have a short-term directive, which results in a  $(1-\mu-p)$  of at most one (1), if none of the failure modes can be root caused. When the TE SW short-term directive is implemented,  $p$  becomes one (1) of the five (5) failure modes that have a short-term directive. As a result  $(1-\mu-p)$  is equal to at most 0.8  $(1-(1/5))$ . Therefore, as shown by equation 7, creating short-term directives reducing the probability that widget failure and enter the bone pile.

## 4 FRACAS METRICS

When starting FRACAS, initial cost savings are not realized because failure modes have not been collected and analyzed and corrective actions have not been assigned. When beginning a FRACAS program it is important to develop a set of metrics that will track  $\mu$ ,  $p$ , and  $\lambda$ . A decreasing  $\lambda$  or an increasing  $\mu$  or  $p$  demonstrates FRACAS cost savings. A decrease in  $\lambda$  indicates that corrective actions are reducing the inter-arrival rate of failure modes. An increasing  $\mu$  and  $p$  indicate effective failure mode root cause analysis and characterization, respectively. Collected metrics can gauge the appropriate amount of person hours needed to support FRACAS efforts. Person hours are gauged by dividing  $\lambda$  by  $\mu$ . A value greater than 1 suggests that failures are arriving faster than they are closed and additional FRACAS support is needed. Several programs have demonstrated that program costs are reduced when FRACAS metrics are collected and tracked on a regular basis. These FRACAS metrics drive continuous improvements to the

production line, which provide multiple cost reduction opportunities positively affecting the organization's bottom line.

#### *REFERENCES*

1. Frank, Gary, "Cost of Production versus Cost of Production", <http://cdp.wisc.edu/pdf/cstvscst.pdf>, 1998
2. "Reliability Program Standard for Systems Design, Development, and Manufacturing", GEIA-STD-0009,
3. "Electronic Reliability Design Handbook", MIL-HDBK-338, 13 November 2008

#### *BIOGRAPHIES*

Eduardo V Cota  
1151 Herman Rd  
Tucson, AZ, 85734, USA

e-mail: [eduardo\\_v\\_cota@raytheon.com](mailto:eduardo_v_cota@raytheon.com)

In 2006, Eduardo graduated from the University of Arizona with a minor in mathematics and a BS in Systems Engineering. Eduardo works for Raytheon Missile Systems, Engineering Product Support Directorate (EPSD), Reliability Engineering Department located in Tucson, AZ for six years. In 2012, Eduardo received his Masters degree in Systems Engineering from Johns Hopkins University. As a Reliability Engineer, Eduardo has performed countless part de-rating analysis, reliability predictions, Failure Modes Effects and Criticality Analysis for one-shot and continuous systems.

Eduardo has performed FRACAS efforts on development programs and recently on a high rate production line.

Louis Gullo  
3360 E Hemisphere  
Tucson, AZ, 85706, USA

e-mail: [Lou.Gullo@raytheon.com](mailto:Lou.Gullo@raytheon.com)

Lou Gullo works for Raytheon Missile Systems, EPSD, Reliability Engineering Department located in Tucson, AZ. He is a leader of several special projects including a new storage reliability handbook for missiles, software reliability methods and automated electrical stress analysis methods. He has 30 years experience in military, space and commercial programs. He recently co-edited, co-authored and published a book, titled: "Design for Reliability". He previously worked for Raytheon Integrated Defense Systems, Honeywell, Texas Instruments, Flextronics, Tyco/Sensormatic, and the US Army. He is a retired US Army Lieutenant Colonel. Lou has a BS degree in Electrical Engineering from the University of Connecticut in 1980. He is an IEEE Senior Member, an elected member of the Administrative Committee (ADCOM), and appointed as the IEEE Reliability Society Standards Committee Chair. He is a member of the Reliability and Maintainability Symposium (RAMS) Management Committee and Secretary/Treasurer for RAMS 2013. He is a member of the Reliability Maintainability Supportability Partnership (RMSP) Board of Directors. He is a member of the IEC TC-56 and VITA 51 standards committees.