Advanced EHF Mission Control Segment (MCS) Software Maturity

GSAW 2008 Presentation

31 March – 3 April 2008

LOCKHEED MARTIN

THE AEROSPACE CORPORATION

Agenda

- System Overview
- Introduction
- Reliability Growth Model Analysis
- Process Improvements

Authors:

Asya Campbell
Systems Director
EHF Mission Control Segment, MILSATCOM
The Aerospace Corporation

Scott Carey
Deputy Program Director, AEHF MCS Program
Lockheed Martin

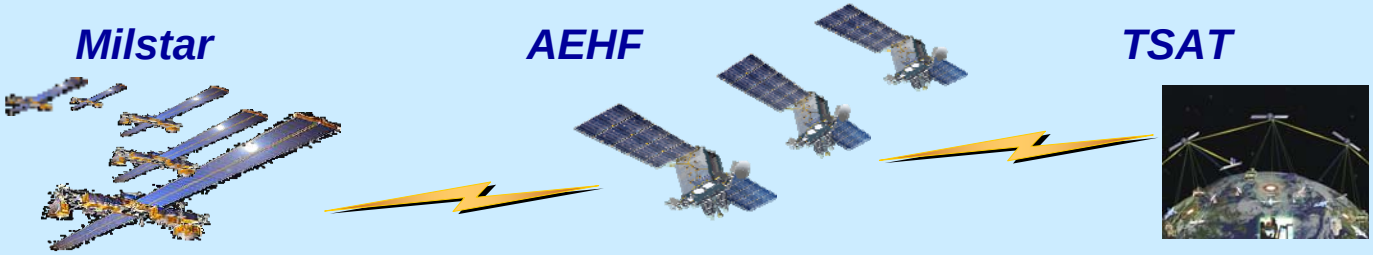
Carlos Rexach
Senior Engineer
EHF Mission Control Segment, MILSATCOM
The Aerospace Corporation

David Thorpe
Engineering Program Manager, AEHF MCS Program
Lockheed Martin

Contact Author:
David Thorpe
610-354-5179
david.p.thorpe@lmco.com
Lockheed Martin IS&GS
Building 33-300
P. O. Box 8048
Philadelphia, PA 19101

This presentation describes work performed under IWTA-C CJ25V0801N to Lockheed Martin Space Systems Company, Space and Strategic Missiles, Sunnyvale, CA under prime contract F04701-02-C-0002 to U. S. Air Force, SMC/MCA, Los Angeles AFB, El Segundo, CA.

System Overview



AEHF Space Segment (3 Space Vehicles)

- Space Vehicle
  - System Timing
  - Autonomous Fault Detection and Correction
  - Jamming/Nuclear Protection
- Payload
  - Communication and Routing
  - Antenna Coverage/ Comm Capacity

Baseband

- Direct I/F to User
- User Interoperability
- End-to-End Services

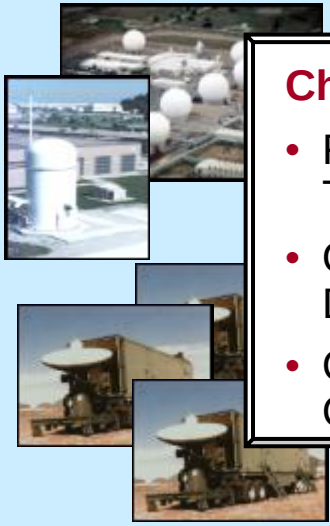
Terminals

- AEHF and Milstar
- International Partners
- User Comm Services
- User Resource Control

Mission Control Segment

- Apportionment Planning
- Sustainment
- Terminal Control
- Command and Control
- Training and Simulation
- Over-the-Air Rekey
- Ground Mobiles

AEHF MCS Mission Operations (MOPS)



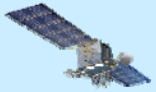
Characteristics

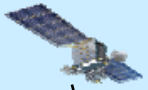
- Function: Command & Control, Telemetry, etc.)
- Object Oriented / Database Driven Design
- OO, Unix, C++, FORTRAN, SCL, Oracle, SQL

Test Program Description

- Modified waterfall
- Incremental Development
- Test Approach
  - Element test – req'ts based
  - MCS test – scenario based

Milstar





SDR/ SRR | Sys and Seg PDR | CY03 | CY04 | CY05 | CY06 | CY07 | CY08 | CY09 | CY10 | O&M

1 | 2 | 3 | 4 | 5 | 7

EPDR | MCS CDR

Inc 1 –Infrastructure Envt Completed Early

Increment 2 – Milstar Reuse

Increment 3 – LDR/MDR Apportionment

Increment 4 – LDR/MDR Command and Control

Increment 5 – XDR Command and Control

Increment 7 – Deferred Functionality

Factory Delivery

Field Deployed

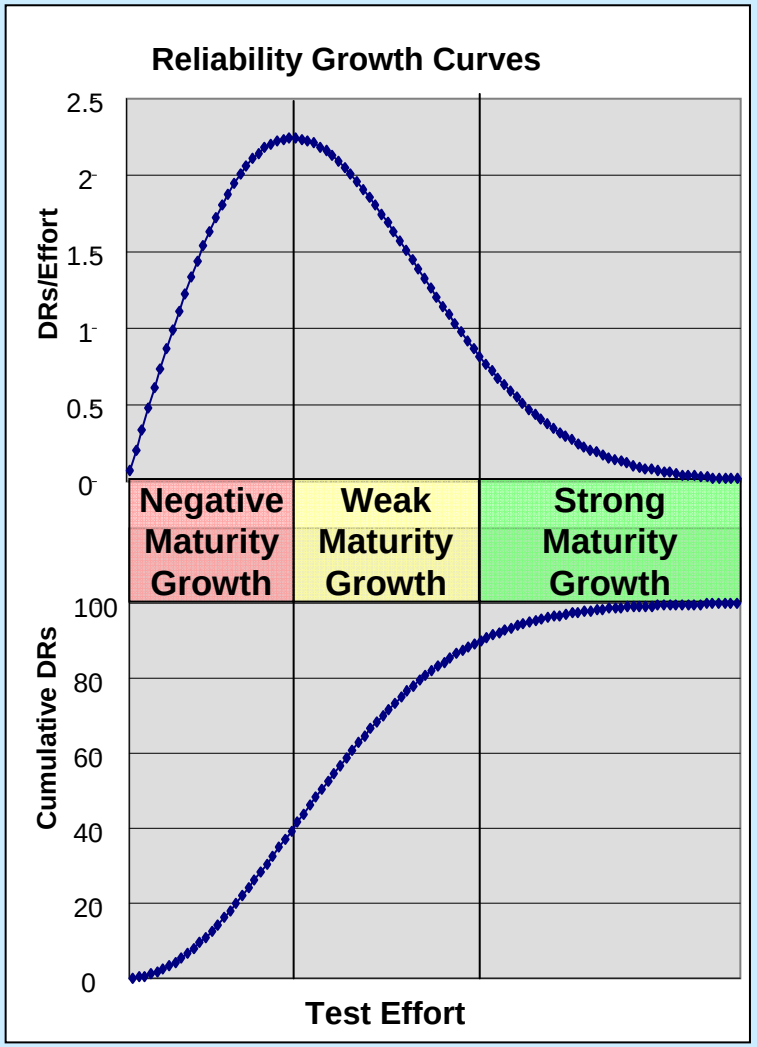
"Photos reprinted courtesy of the United States Department of Defense."

Introduction

- This brief provides results of our work to develop accurate and practical software defect density estimates using Weibull and Yamada-S Reliability Growth Models (RGM)
- Significant findings include:
  - Analysis of RGM curves developed with DR data in the “calendar-domain” can lead to inaccurate (and optimistic) estimates of total DRs in the code. The contributions of test effort on DR detection rates must be understood.
  - Aggregation of data (components, priority levels, etc.) from populations with dissimilar maturity characteristics degrades accuracy of estimates.
  - RGM results can provide useful information to software testing and release decisions

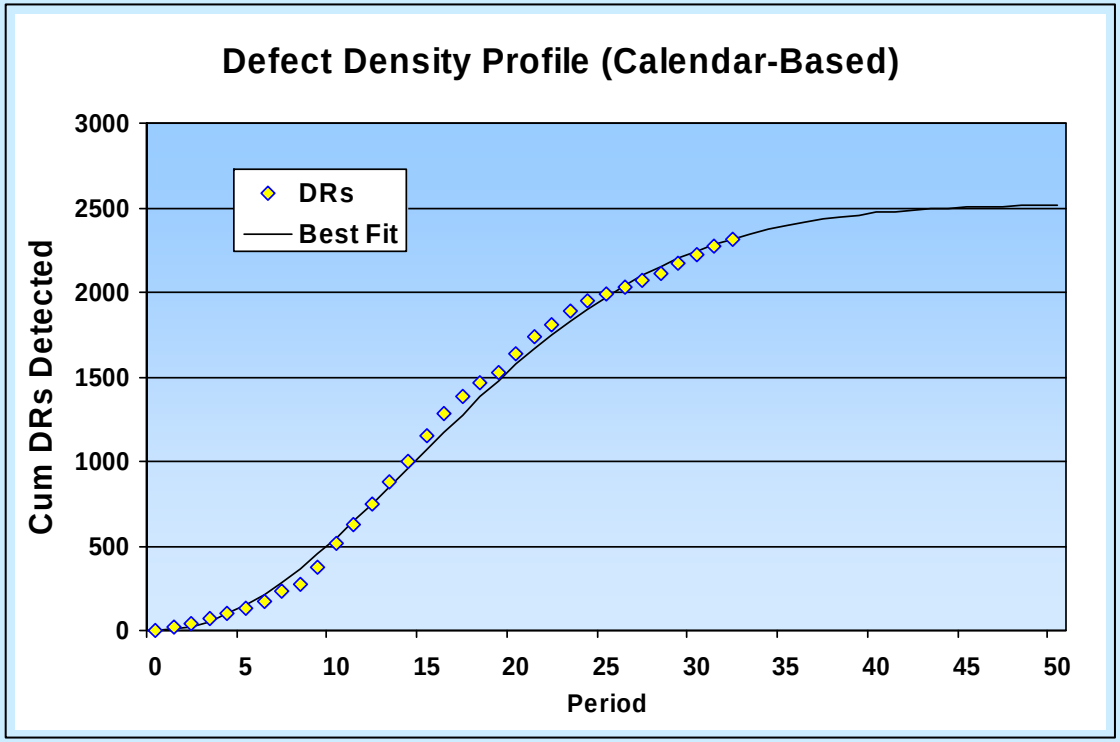
What are Reliability Growth Models (RGM)

- As we progress through the software test program, defect detection becomes increasingly more difficult
- RGMs are families of mathematical functions (Weibull, NHPP, etc.) that model the detection rate of software DRs during development (and operations)
- RGM analysis can be used to predict future detection rates and DR density once the code begins to demonstrate reliability growth
- RGMs are robust, although the DR data must still satisfy a number of assumptions



Analysis of Calendar-Based DR Detection Rates

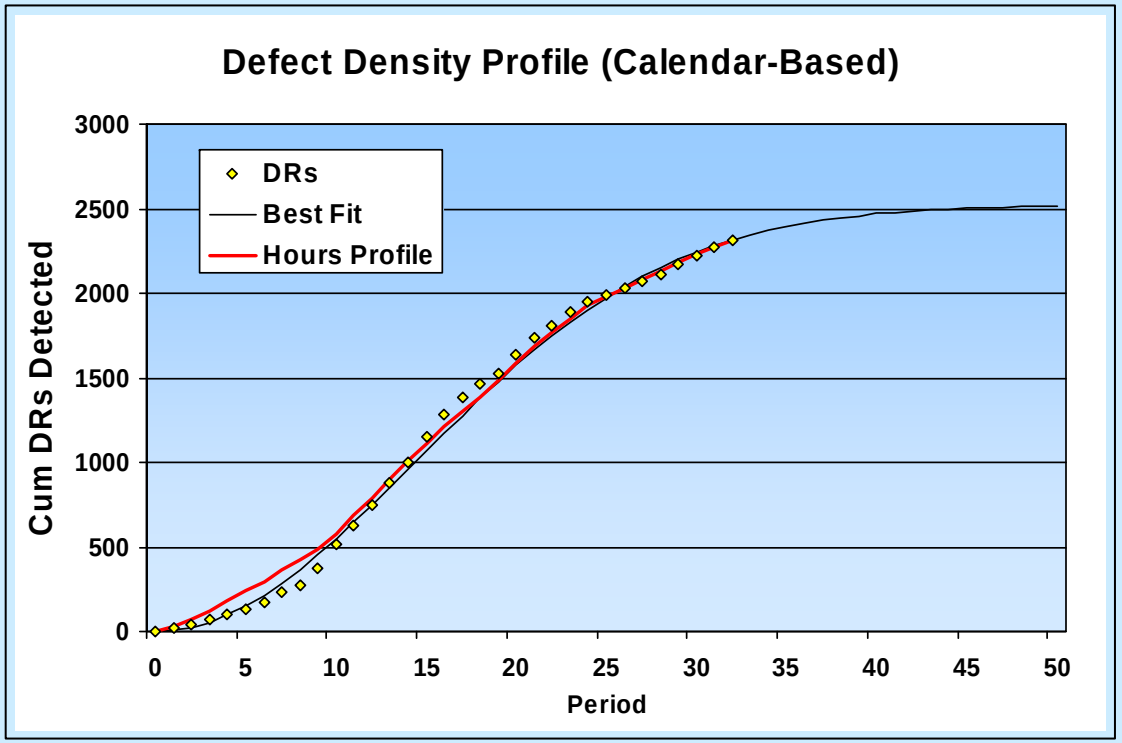
- Calendar-based models assume a constant test effort across all periods (hours, days, weeks, etc.)
- RGM analysis of MOPS DRs show strong reliability growth with 93% of total estimated DRs found to date



What is driving the drop in DR detection rates?

Analysis of Calendar-Based DR Detection Rates

- Calendar-based models assume a constant test effort across all periods (hours, days, weeks, etc.)
- RGM analysis of MOPS DRs show strong reliability growth with 93% of total estimated DRs found to date

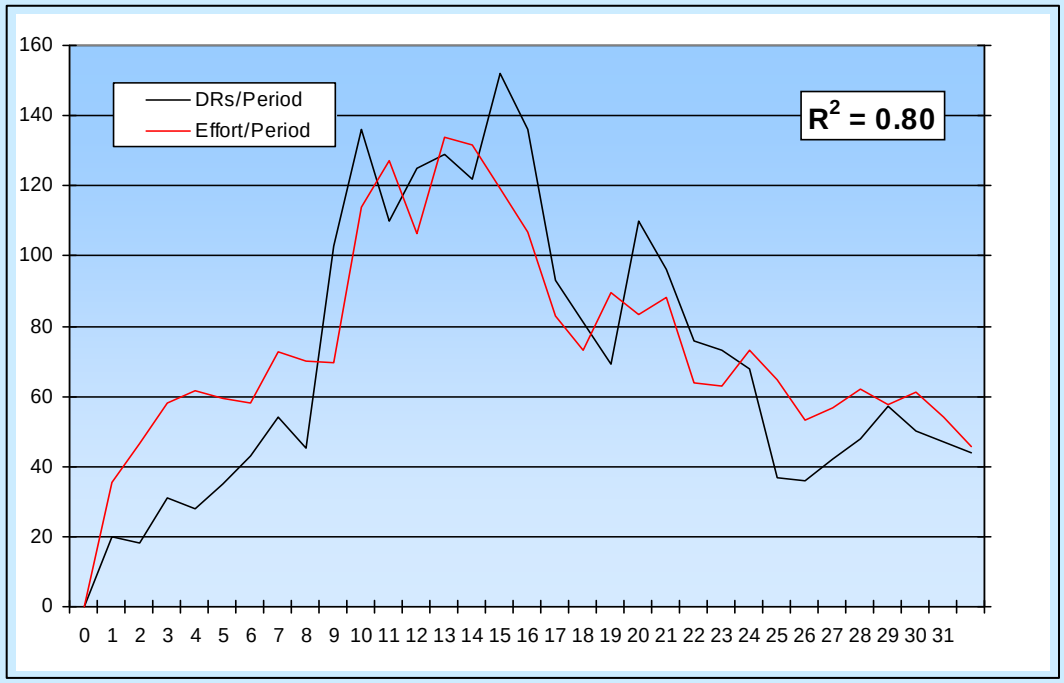


- Hours profile shows strong correlation between effort and DRs

Test effort, not reliability growth, is driving DR detection rates

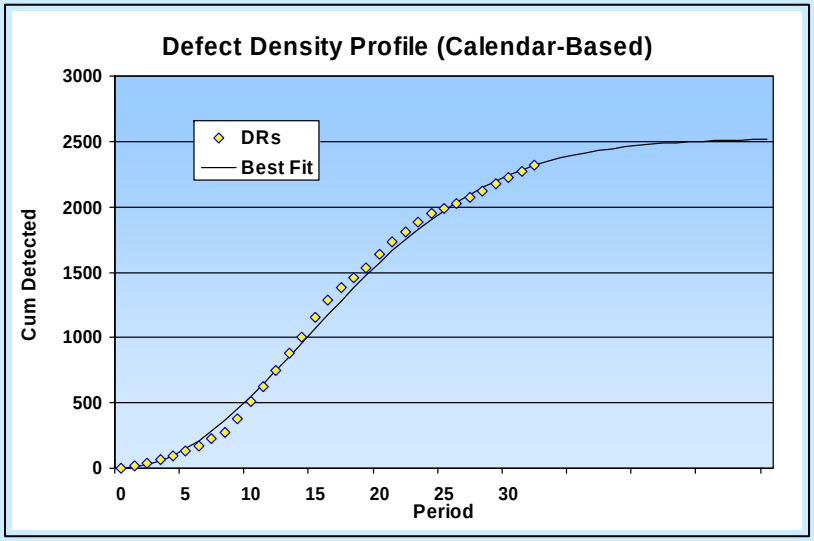
Analysis of Calendar-Based DR Detection Rates

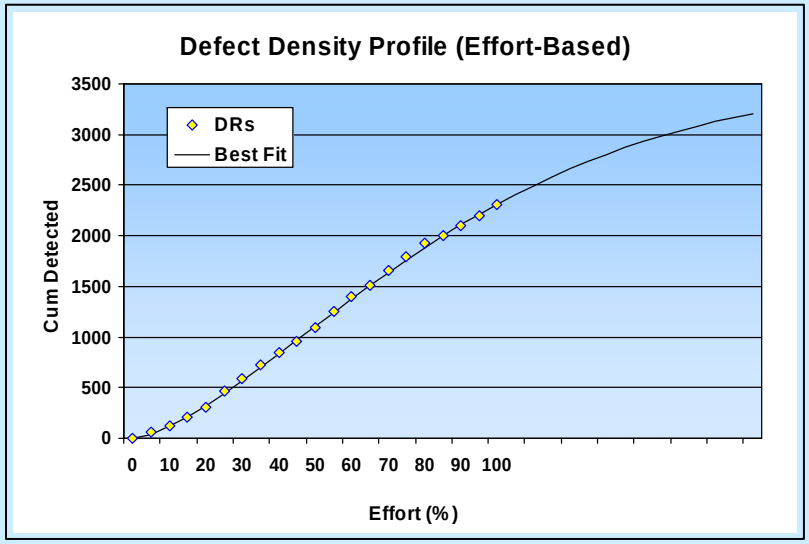
- Test hours alone explains 80% of the change in DR detection rates
- Data demonstrates a strong correlation between test effort and DR detections
- Reliability growth is a minor contributor to changes in MOPS DR detection rates



Calendar-Based approach is highly susceptible to changes in effort

Analysis of Effort-Based DR Detection Rates





Comparison between Calendar- and Effort-Based DDPs:

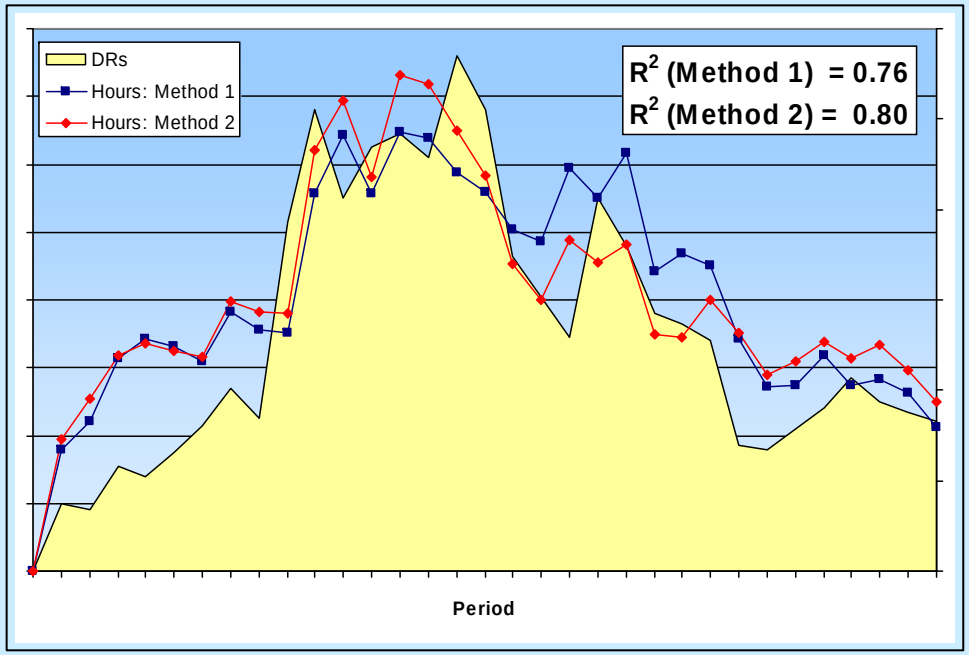
- Calendar-based analysis
  - With 93% of total estimated DRs found the software is in the strong reliability region of its S-curves and ready for fielding
- Effort-based analysis
  - With 65% of total estimated DRs found the software is in the weak reliability region of its S-curves and not ready for fielding

Calendar-Based estimates can yield unreliable results.

Test Effort Profile Estimates

Our goal is to develop a profile that depicts the relative on-console test effort from labor data which includes many other test-related activities.

- Method 1: Total hours across all test accounts and activities
  - Pros: Easiest metric to collect and implement
  - Cons: Assumes all test stages have equal efficiency
- Method 2: Length of test procedures and hours charged to test accounts
  - Pros: More accurate test effort profile: introduces test phase efficiency coefficients
  - Con: More difficult to implement



Both methods yield adequate test profiles

Process Improvements

- Established a set of Software Maturity metrics to track the progress of the software through the development / test lifecycle.
  - Observations from the RGMs along with other key metrics provide great insight into the reliability and readiness of the software.
  - Utilizing RGMs based on test effort provides a more accurate depiction of software reliability
- Developed additional metrics including software change frequency metrics which have identified specific areas requiring additional attention.
- Improved our processes to include a forum for the analysis and discussion of software reliability.
- Developing software maturity goals which can be eventually used to help answer the age-old questions “is the software ready to ship and if not, when will it be ready?”