

The Integrator

INCOSE North Star Chapter



Volume 6, Issue 2

April 2010

North Star Newsletter

INCOSE North Star Newsletter Communication

Areas of interest you'd like to see? Let us know! Drop me a line at Eileen.Arnold@INCOSE.org!

Eileen Arnold, Editor BAE Systems

FEAR: "Never let the fear of striking out get in your way" – Babe Ruth, professional baseball player

Systems Engineering at its Best!

Enabling Measurement-Driven System Development by Analyzing Testing Strategy Tradeoffs - Richard W. Selby

The processes of software testing and defect detection continue to challenge the systems engineering and software communities. Even though the software testing and defect detection activities are inexact and inadequately understood, they are crucial to the success of a software project. Measurement-driven system development focuses on using quantitative data to evaluate capabilities, benefits, progress, and tradeoffs as well as identify improvement opportunities. This article summarizes a controlled study that addressed software testing effectiveness and focuses on the combination of individual testing techniques into team-based testing strategies. This analysis is intended to enable measurement-driven process improvement by characterizing how testing effectiveness relates to several factors, including testing strategy, software type, and developer expertise. In this study, a representative group of software development professionals applied common testing techniques to different types of software. This study compares the six possible team combinations of three testing techniques:

- 1) Code reading by stepwise abstraction,
- 2) Functional testing using equivalence partitioning and boundary value analysis, and

Continued on page 2

Chapter President's Corner

Mark Elpers, Medtronic
INCOSE North Star Chapter 2010 President

Welcome again to the North Star chapter of INCOSE! Spring is here and all are excited about the upcoming summer. In addition to INCOSE North Star Chapter excitement, expectations of a successful baseball season at our new ballpark are at all time highs. Baseball has been an INCOSE topic in the past, a paper by Jim Armstrong, which only goes to show the wide-spread usefulness of applying Systems Engineering!

Our 2010 plan for spring builds on the early Chapter schedule success with a chapter meeting and tutorial from industry expert Kevin Forsberg as well as some community outreach to our student community. We conclude the quarter with a review of the papers submitted for the International Symposia and a lecture from another industry leader, Eric Honour, on the return on investment you can expect for various levels of systems engineering effort.

Most importantly, I want to thank everyone for the overwhelming response to our member survey. We are already extracting useful information from the results and we will be using this information to make our chapter meetings, tutorials, communication mechanisms and other activities as useful to you as they can possibly be. I would also encourage you to email me directly with suggestions for improvements or suggestions for things we should stop doing. We are always looking for volunteers to help keep this an outstanding Chapter! You are our customers and we want to do everything we can to make our chapter a valuable item in your professional lives. Stay tuned for more details and thanks again for the great survey responses! <mark.elpers@medtronic.com>

North Star Chapter Website
<http://www.incose.org/northstar>

3) Structural testing using 100% statement coverage criteria.

Thirty-two professional developers applied the techniques to three unit-sized programs in a fractional factorial experimental design. The 28 faults in the programs comprise a reasonable distribution of faults that commonly occur in software. All the faults in the database maintainer and the numeric abstract data type were made during the actual development of the programs. The text formatter contains a mix of faults made by the original programmer and faults seeded in the code. Note that this investigation involves only those types of faults occurring in the source code, not other types such as those in the requirements or specifications.

A fault categorization scheme used, partitioned software faults into the six classes with fault totals indicated in parentheses:

- 1) Initialization (Omission (0), Commission (2)),
- 2) Computation (Omission (2), Commission (2)),
- 3) Control (Omission (2), Commission (4)),
- 4) Interface (Omission (2), Commission (11)),
- 5) Data (Omission (2), Commission (0)), and
- 6) Cosmetic (Omission (0), Commission (1)).

The controlled study included five phases: training, three testing sessions, and a follow-up session. All groups of subjects were exposed to a similar amount of training on the testing techniques before the study began. In the testing sessions, the individuals were requested to use the testing techniques to the best of their ability. The subjects' desire for the study's outcome to improve their software testing environment ensured reasonable effort on their part. Note that when the subjects were applying either functional or structural testing, they generated and executed their own test data; no test data sets were provided. There were no limits on the amount of time for testing the programs. When a subject was structurally testing, an automatic program coverage monitor tool identified unexecuted statements. The study concluded with a debriefing session for discussing the results and the subjects' observations.

The major results of this study

- ◆ The six combined testing strategies detected 17% more of the programs' faults on the average than did the three single techniques, which was a 35% improvement in fault detection.
- ◆ The highest percentages of the programs' faults were detected when there was a combination of either two code readers or a code reader and a functional tester. However, a pairing of two code readers detected more faults per hour than did a pairing of a code reader and a functional tester.

- ◆ The pairing of two individuals of advanced expertise resulted in the highest percentage of faults being detected.
- ◆ The most cost-effective (number of faults detected per hour) testing strategy overall was when code reading was applied by an individual. The most cost-effective combined testing strategy was when a code reader was paired with either another code reader or a structural tester.
- ◆ Both the percentage of faults detected and the fault detection cost-effectiveness depended on the type of software being tested.

WELCOME, NORTH STAR NEW MEMBERS!

Name	Company	Title
Yilmaz Bayazit	Eaton Corp	SE
Jeff Bradley	Mayo Clinic	Lead Eng.
Peter Chong	Goodrich Corp	Eng. Mgr
Kenneth Enstrom	GD-AIS	SW-SE
Sharon Gabrielson	Mayo Clinic	Section Head
Weston Kirch	Aerotek	SW Eng Acct Rep
Chris Petersen	Medtronic	Sr. Prin. SE
Chris Senanayake	NA	NA
Michael Stoner	Eaton Corp	Eng.

2010 Meeting Calendar

13 May	SE in Student Projects	Wayzata High School
10 June	IS Paper Dry Run plus Tour	U. of St Thomas
11-15 Jul	20th INCOSE International Symposium	Chicago, IL
12 Aug	SE in Risk and Project Management	LMCO
9 Sep	SE in Operating Systems	City Water Works
14 Oct	SE in Production	Ford Plant
11 Nov	SE in Architecture and MBSE	Medtronic