Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 114 (2017) 28–36

Procedia Computer Science

www.elsevier.com/locate/procedia

Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA

Minimizing Safety Risks in Complex Systems

Chandru Mirchandani, PhD; INCOSE Fellow; AIAA (Assoc. Fellow)*

Adjunct Professor, George Washington University, Washington DC, USA

Abstract

There is a need to develop an adaptive model by which the changes in the testing and upgrade processes for complex software can be observed.

The metrics obtained from a software system and a sufficient data base for software failures and operating hours, help to develop a model for predicting the reliability-relevant failures based on actual data and fitting a distribution. However, the collection, identification and classification of the failure data is not timely enough to identify the cause of the failure.

From the system engineering perspective testing to an acceptable level is needed to validate the system capability and performance. Testing in a perceived operational environment reduces the risk of the system not meeting the required capabilities. Historically this risk was evaluated in terms of cost, schedule and data. This paper will examine the safety hazards to system design by examining the consequences based on the interaction of data and function and the ensuing behavior complicated by an uncertain input from the human. The paper will identify and develop metrics and interactions thereof which when varied can change the test profile to minimize the uncertainty or risk of the system performance in realistic scenarios for enterprise service systems minimizing safety risk.

Peer-review under responsibility of the scientific committee of the Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems.

Keywords: Safety Hazards; CHI-squared Distribution;

* Corresponding author. Tel.: +1-301-609-0129;
E-mail address: chandru.mirchandani@gmail.com

1877-0509
Peer-review under responsibility of the scientific committee of the Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems.
10.1016/j.procs.2017.09.060

1. Introduction

Safety Hazard, what is it….? It can be broadly defined as a negative consequence that poses danger to the system or its operators based on the interaction of data and function and the ensuing behavior complicated by an uncertain input from the human. To understand the hazard and safely mitigate it, we have to measure it, and this paper proposes the following steps that can be used to estimate it quantitatively and qualitatively:

- Observing the occurrence from data if available OR
- Estimating the frequency of occurrence from analysis
- Observing the consequence of the pattern of observed data if available OR
- Predicting the consequence based on analysis
- Evaluating the consequence based on the cost (or harm) to property and life.

2. Metrics for Predicting Software Failures

Many factors influence software reliability, namely the software development processes, the complexity of the system and software requirements, experience of the software developers, the development environment, and the amount and thoroughness of testing. The literature states that software failures are due to code deficiencies, i.e. faults, which when triggered by data or a computer state that causes execution of these faults, result in failures. Software development incorporates these factors into a software reliability modeling approach based on the analyses of corrective action field data collection and failure recording of our software development and testing experience. This experience shows that the fault intensity rate (faults discovered per month) follows a Weibull distribution over calendar-time.

Unlike hardware, software reliability does not alter with time unless conditions change or are modified during testing and debugging. However, this paper asserts that it is possible to equate the periods for software failure exposure to follow the bathtub curve for a Poisson distribution, where the initial periods are analogous to the infant mortality, and which for software, are the initial integration problems. With increasing periods, the integration problems are eclipsed by actual software problems which include design and interfacing issues between software subsystems within the whole system. These failure exposure rates reach local maxima before they start to decrease whereby they effectively follow the Weibull distribution which shows a reliability growth with more and more of the inherent software defects being exposed as failure problems.

3. Software Reliability Model

This model predicts the failure rates of various operational software components based on failure Find or Failure Discovery Plan. The Find Discovery Plan tracks all defects beginning with software hand-off to test, integration and test phases through customer operational use. It is based on data collected from previous programs, where we assume that the validity ratio remains relatively constant throughout testing and Customer Acceptance of the system. The initial data point on the Find Plan curve includes all predicted defects prior to the Software Hand-off AND are transferred into the problem tracking database as well as new problems found during the first month of testing. It should be noted that the failure rate or failure exposure rate is determined in part by the amount of executable code and expected fault content.

The actual collected data is fitted to a Weibull distribution, taking into account only those failures that can cause the system to fail, i.e. fail to provide the operational service. The criticality of the service is determined by the classification given by the stake-holder and can be a routine function with minimal impacts to a critical function with catastrophic consequences. Using the method of least squares the shape and scale parameters for the distribution.

Using the Cumulative Distribution Function given by:

$$F(x;\alpha,\beta)=1-e^{-(x/\beta)^{\alpha}} \quad (1)$$

Where the Probability Density Function (PDF) is given by: