Estimating Software Reliability for Space Launch Vehicles in Probabilistic Risk Assessment (PRA)

Steven Novack[1], Mohammad Al Hassan[2], Adam Harden[1], Steve Kossow[1]
[1] Bastion Technologies
[2] NASA Marshal Space Flight Center

Abstract

It is acutely recognized in the Probabilistic Risk assessment (PRA) field that software plays a defining role in overall system reliability for all modern systems across a wide variety of industries. Regardless if the software is embedded firmware for working components or elements, part of a Human-Machine-Interface, or automated command and control logic, the success of the software to fulfill its function under nominal and off-nominal environments will be a dominant contributor to system reliability. It is also recognized that software reliability prediction and estimation is one of the more challenging and questionable aspects of any PRA or system analyses due to the nature of software and its integration with physics based systems. Irrespective of this dichotomy, any incorporation of software reliability methods requires that the contributions are accountable, quantitative, and tractable.

This paper provides a brief overview of software reliability methods, establishes some minimum requirements that the methods should incorporate for completeness, and provides a logic structure for applying software reliability. Model resolution will be discussed that supports current testing plans and trade studies. We will provide initial recommendations for use in the NASA PRA and present a future dynamic option for software and PRA.

Space Launch Vehicle Software is recognized to be reliable in static conditions, yet relatively vulnerable to a set of failure modes in changing environments/flight phases. Two quantitative methods were chosen to incorporate software reliability into a Space Launch Vehicle PRA accounting for phase adjustments. One method predicts latent software failure using statistical methods, and the second provides estimates of coding errors and software operating system failures based on test and historical data, respectively. Software uncertainty will also be discussed. We determined that recommendations for PRA software reliability should be modeled at the software module level where multiple software components compose a module and combinations of the software architecture can lead to a functional failure.