Assessment and Integration of Software Risk with Overall System Risk

Presented by
Michael Yau, Ph.D.
ASCA, Inc.
Redondo Beach, California, U.S.A.

at:
2005 Asia Pacific Conference on Risk Management and Safety
December 2, 2005
Hong Kong

Introduction

- **Software (SW) is a key component of modern space systems**

- **Software risk modeling framework and technique development has not yet resulted in generally accepted solutions to assessment problem**

- **Most Probabilistic Risk Assessment (PRA) or system reliability assessments consider SW contribution to risk negligible in comparison to, and/or included in, hardware component contributions to system failure rate.**

SW-Induced Space System Failures

- Ariane 5 Launch Vehicle Failure, June 4, 1996
- Delta III 259/Galaxy X, August 26, 1998
- Centaur Upper Stage Failure in Titan-IV Launch Vehicle Mission, April 30, 1999
- Mars Climate Orbiter Mission Failure, September 1999
- Mars Polar Lander Mission Failure, December 1999

Conditional Software Risk Model

- To better estimate the risk contribution of the software to the overall system, a conditional software risk approach was suggested by “Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners,” Version 1.1, August 2002.

- “Conditional risk” formulations (also referred to as “structural,” “white box,” or “context-dependent” formulations
  - recognize that software behavior is context dependent
  - partition the input space and estimate the condition risk in each partition

Conditional Software Risk Model (cont.)

- **Quantification of the software failure risk can be represented by the following Risk Index formulation:**

where:

[illegible]

  - – $P(C_k)$ = Probability of occurrence of a software error-forcing context (or input condition)
  - – $(P_{S/Ck})$ =Conditional probability of successful software execution, given the software error-forcing context

SW Conditional Risk Approach under Development

- **As part of a project funded by NASA Headquarters through Johnson Space Center (JSC), a conditional risk approach combining Dynamic Flowgraph Methodology (DFM) and SW testing analysis is being developed**

- **This approach consists of 3 major steps:**

  1. ***Identify mission-critical function supported by software.***

  2. ***Use DFM to identify conditions of mission-critical function execution that may include, or trigger, software errors. These conditions are then quantified to estimate the P(C_k) terms.***

  3. ***Quantify or "bound" the probability of software errors using the conditional failure model in preceding chart.***

     - *Identify the nature, testability level and actual type testing executed for each identifiable condition C_k of interest, so that an empirical adjustment factor can be applied to the probability of failure for the associated software module, by application of one of the available, test-based, "black-box" reliability models*

DFM Background

- **Developed by ASCA, Inc. in the 1990s as a software tool to support PRA**

- **Software was used in the safety analysis of several software controlled systems. The results validated DFM's ability to handle software & hardware interactions and to perform dynamic analysis**

  - Digital feedwater control system in an advanced Pressurized Water Reactor (NRC SBIR)

  - Control system for the Combustion Module-1 System (NASA Glenn Research Center project)

DFM Features

- **Graphic modeling environment and automated analysis engine that can handle**
  - cause-effect relationships
  - time-dependent relationships
  - feedback loops
  - multiple (>2) states
- **Discretized state-vectors represent key process parameters**
- **Mapping between the discretized state-vectors governed by multi-valued logic rules**
  - decision tables
  - transfer-boxes
  - transition-boxes

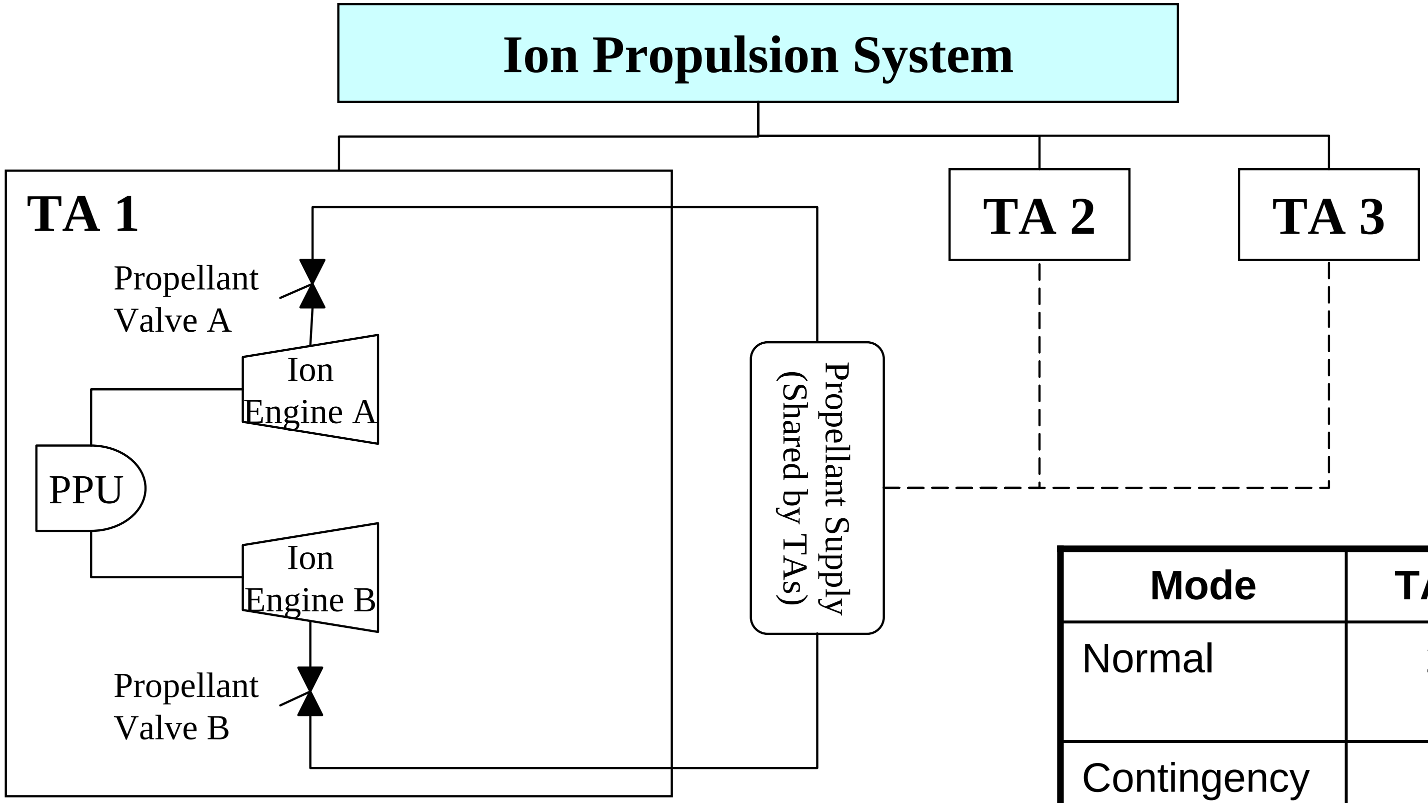
SW Risk Estimation Table

- **A SW risk table like the one shown below can be assembled to take into account the actual applicability to a specific function of a POF estimate obtained via a SW reliability-growth model**
  - *i.e., the SW reliability model may have been applied to a SW module containing the function without actually exerting the latter, or exerting it under other than the system mission configuration*

Input Condtn. Ck	*SW Function*	*Type of Testing Executed*	*Adjustmnt. To Condtnl. Prob. PF_{Ck}*
Normal	Routine	Formal in Actual HW/SW System Configuration	None: Use Value from SW Rel. Growth Model
		Formal in Simulated System Configuration	Adjust SW Rel. Growth Model Value w/ 2-5 Factor
Exception	Defined - Simple	Formal in Actual HW/SW System Configuration	None: Use Value from SW Rel. Growth Model
		Formal in Simulated System Configuration	Adjust SW Rel. Growth Model Value w/ 5-10 PF Factor
		Not Formally Tested	Assume PF to be 0.01 – 0.1
	Defined - Complex	Formal in Actual HW/SW System Configuration	None: Use Value from SW Rel. Growth Model
		Formal in Simulated System Configuration	Adjust SW Rel. Growth Model Value w/ 10-50 PF Factor
		Not Formally Tested	Assume PF to be 0.2-0.5
	Undefined	N/A	Assume PF to be > 0.5

Example of SW Risk Assessment

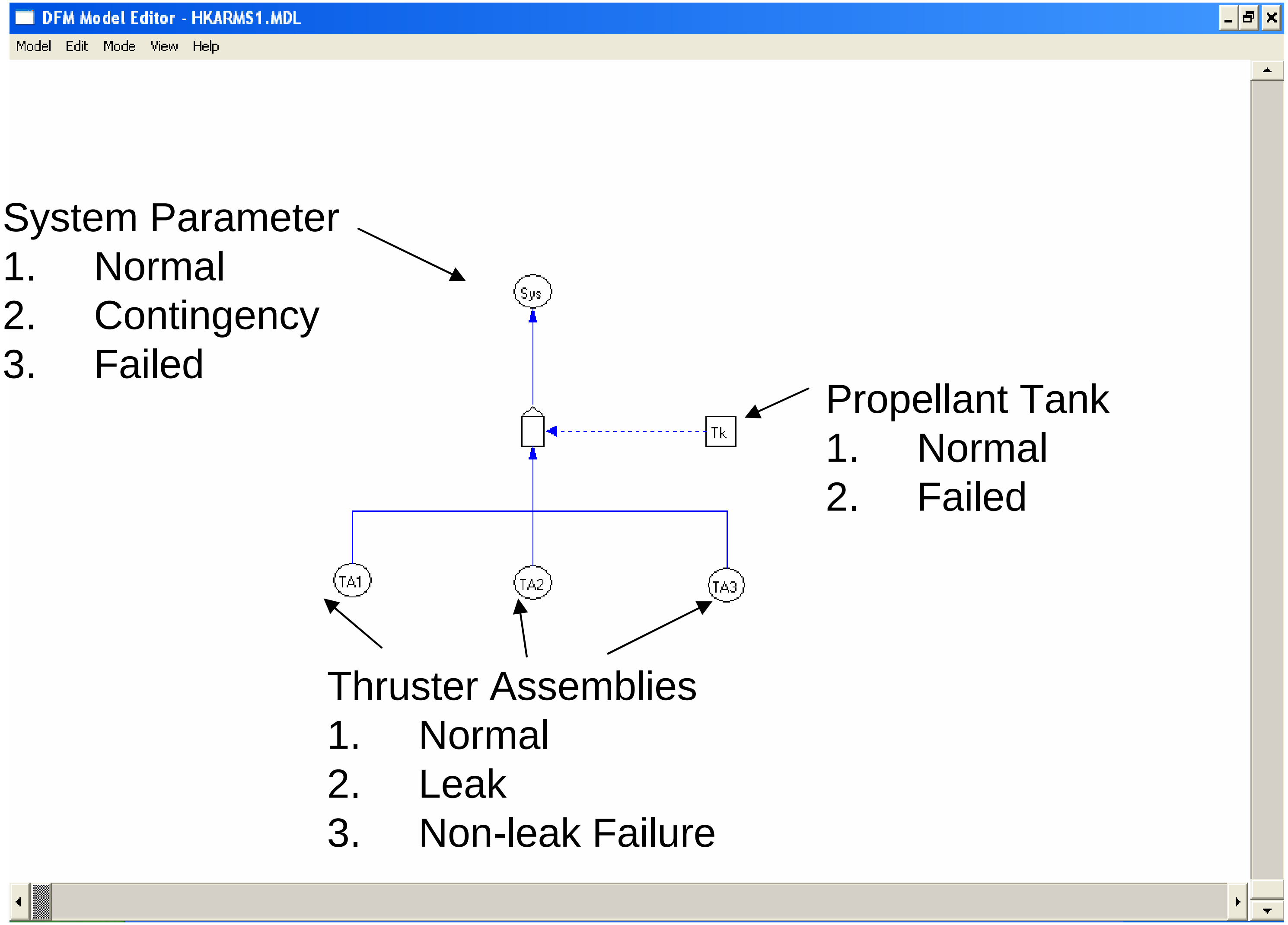
- **Considers the assessment of probability of failure of the Ion Propulsion System with 3 Thruster Assemblies (TAs).**



Mode	TAs Available
Normal	2/3 thruster assemblies
Contingency	1/3 thruster assemblies

System Level DFM Model

- **Use DFM to determine and quantify the error forcing contexts of the software**



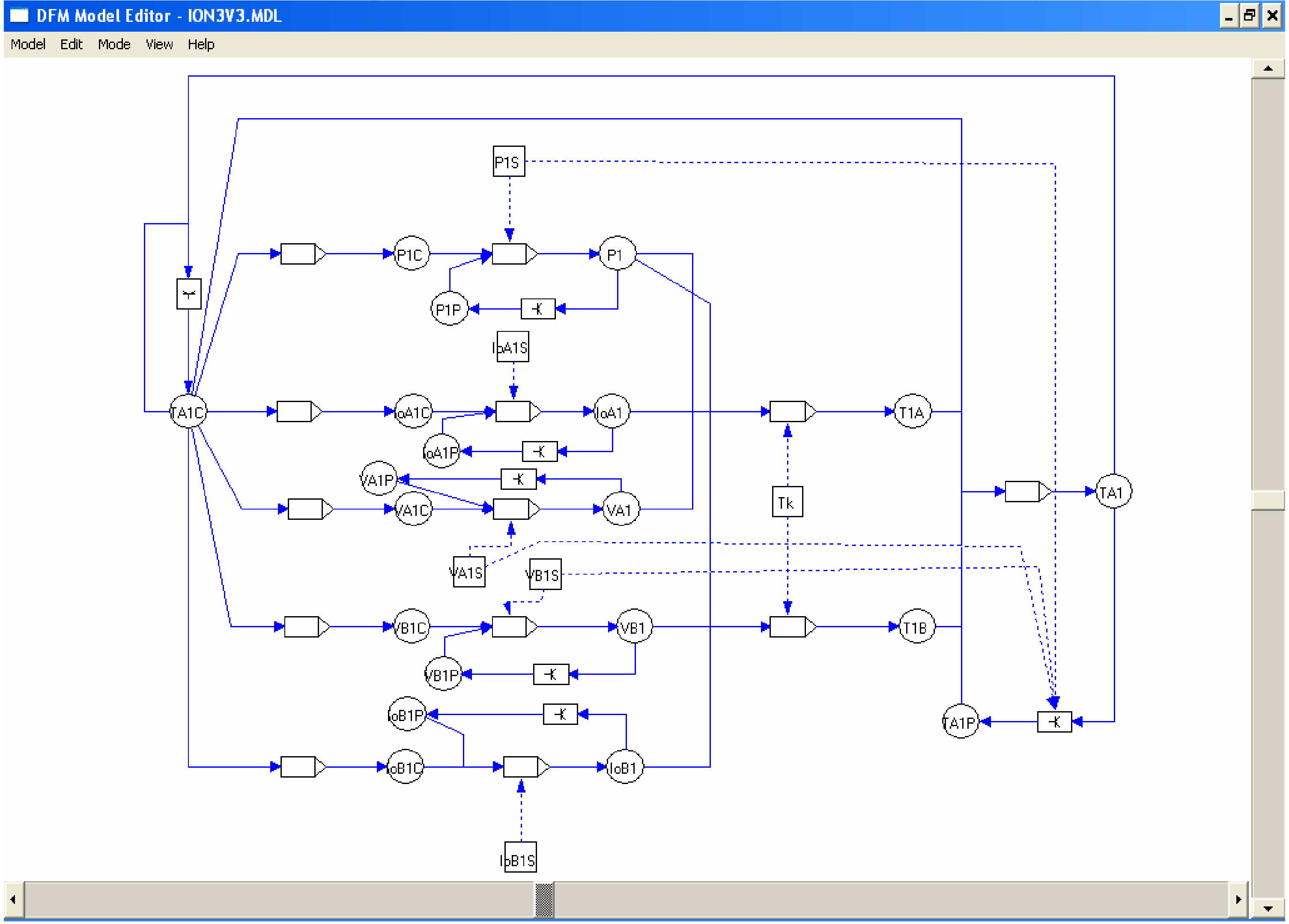
System Parameter
1. Normal
2. Contingency
3. Failed

Propellant Tank
1. Normal
2. Failed

Thruster Assemblies
1. Normal
2. Leak
3. Non-leak Failure

Sub-system Level DFM Model

- **This model shows the detailed redundancy management and recovery action sequence of Thruster Assembly 1**



M. Yau

DFM Model Quantification

- **Probabilities of the error-forcing contexts can be determined by analysis of the DFM models**

- **Probabilities are estimated from the bottom up. Quantify the sub-system level model first, and use the results in the system level model**
  - Propellant tank failure probability = 5 x 10^{-4}
  - After analysis of the sub-system level model:
    - P(Thruster assembly leaks) = 1 x 10^{-3}
    - P(Thruster assembly failed in non-leak manner) = 5 x 10^{-3}

DFM Model Quantification (cont.)

- **Analysis of the system level model:**
  - Top Event = System in the failed state, prime implicants (multi-valued logic equivalent of minimal cut sets in binary fault trees) are:
    1. TA1 leaks
    2. TA2 leaks
    3. TA3 leaks
    4. Propellant tank fails
    5. All 3 TAs failed in the non-leak manner
  - P(System = failed) = 3.496 x 10-3
- **Use the same procedure to determine the prime implicants and the probabilities for the contingency state and the normal state.**
  - P(System = contingency) = 7.451 x 10-5
  - P(System = normal) = 9.964 x 10-1

Conditional Risk Model Estimation for Ion Propulsion System

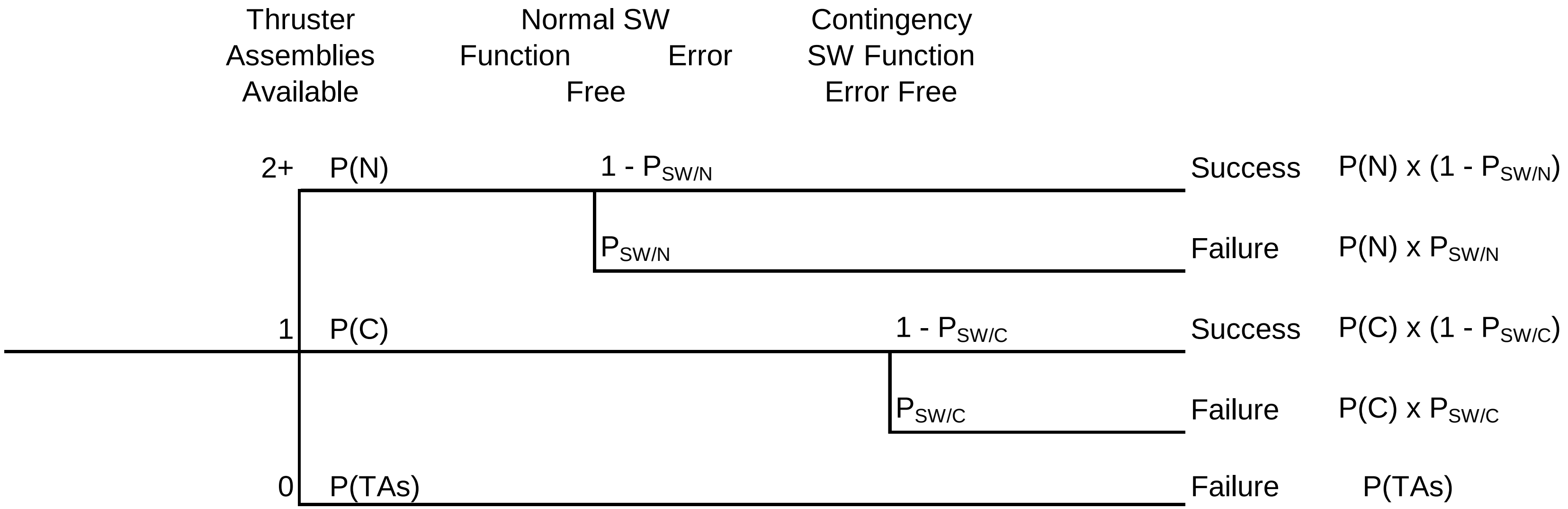
- **In conditional risk model, SW function failure probability is estimated both in terms of SW errors and in terms of SW input conditions, which are usually related to hardware success and/or failures:**

$$P_{Ion} = \underbrace{P(N)\,P_{SW/N} + P(C)\,P_{SW/C}}_{\text{SW Failure Probability}} + \underbrace{P(TAs)}_{\text{Component Failure Probability (HW only)}}$$

N ≡ Normal condition

C ≡ Contingency condition

TAs ≡ Thruster assembly-set



Software Function Failure Probabilities

- Software-function failure probabilities, $P_{SW/N}$ & $P_{SW/C}$, are derived by use of software reliability model predictions (e.g., Schneidewind's or Musa-Okumoto's formulations), adjusted with factors to account for the testability of the functions of interest and the conditions under which testing and fault-removal was executed

  Example, Part 1: Assume that Table I attributes relative to "Normal Condition" software function are:

  Input Condition: Normal

  Function: Routine

  Type of testing: Formal in simulated system configuration

- Also assume that test results for this software function are such that the applied SW reliability model (e.g., Schneidewind's) yields an estimated non-adjusted POF value:

$$P_{SW/N} = 1.0\ E\text{-}\ 4$$

Software "Normal Function" Failure Probability

- Under stated conditions, Table I suggests an adjustment factor between 2 and 5: because the SW function has operated successfully in one earlier mission, we choose 3 ;

$$P_{SW/N}' = 3\ P_{SW/N} = 0.0003$$

Input Condtn. Ck	*SW Function*	*Type of Testing Executed*	*Adjustmnt. To Condnl. Prob. PF_{Ck}*
Normal	Routine	Formal in Actual HW/SW System Configuration	None: Use Value from SW Rel. Growth Model
		Formal in Simulated System Configuration	Adjust SW Rel. Growth Model Value w/ 2-5 Factor
Exception	Defined - Simple	Formal in Actual HW/SW System Configuration	None: Use Value from SW Rel. Growth Model
		Formal in Simulated System Configuration	Adjust SW Rel. Growth Model Value w/ 5-10 PF Factor
…	…	…	…

Ion Propulsion System
Failure Probability

- **Quantification of the formulae with the probabilities just estimated finally yields:**

HW probabilities:

$P(N) = 0.9964$

$P(C) = 7.451E\text{-}5$

$P(TAs) = 3.496E\text{-}3$

SW probabilities:

$P_{SW/N}' = 3.00E\text{-}4$

$P_{SW/C}' = 1.00E\text{-}2$

For overall Risk Index:

$$RI_{Ion} = P(N)\, P_{SW/N}' + P(C)\, P_{SW/C}' + P(TAs)$$

$$= \underbrace{2.989E\text{-}4 + 7.451E\text{-}7}_{\text{SW-driven Failure Probability}} + \underbrace{3.496E\text{-}3}_{\text{HW-only Failure Probability}}$$

$$= 2.997E\text{-}4 + 3.496E\text{-}3$$

$$= 3.795E\text{-}3$$

Concluding Comments

- **Events in the past showed that software contribution to system risk is NOT negligible.**
- **Conditional risk approach is being developed within the scope of a NASA funded project to estimate system risk attributed to software**
  - Partition the software input space into error-forcing contexts
  - Apply DFM to find the conditions that cause the error-forcing contexts and estimate the probabilities for these error forcing contexts
  - Apply black box reliability growth models (with some adjustments to account for testability) to estimate the conditional software risk
- **Project progress:**
  - Validate this approach with a more complex test case provided by JSC
  - Select black box reliability growth model(s)
  - Refine adjustment factors applied to tests under simulated conditions