



Software Safety In TLS



2011 ADF Aviation Software Symposium



AUSTRALIAN
AEROSPACE
EUROCOPTER, AN EADS COMPANY

Slide 0000: Overview

— Introduction

- ARH Tiger background

— Current Situation

- What we already do
- Room for improvement

— Software Safety

- Software Safety and how it works
- Case Study for ARH

— Fault Trees

- Fault Tree example
- Software Fault Tree example

Slide 0001: Context

The Target Audience

- People dealing with software-intensive Safety Critical systems

The Catalyst

- Software Safety Regulation

The Goal

- Safer Systems

The Byproduct

- Compliance with the regulations
- Documentation

The Disclaimer

- Not claiming to be the expert

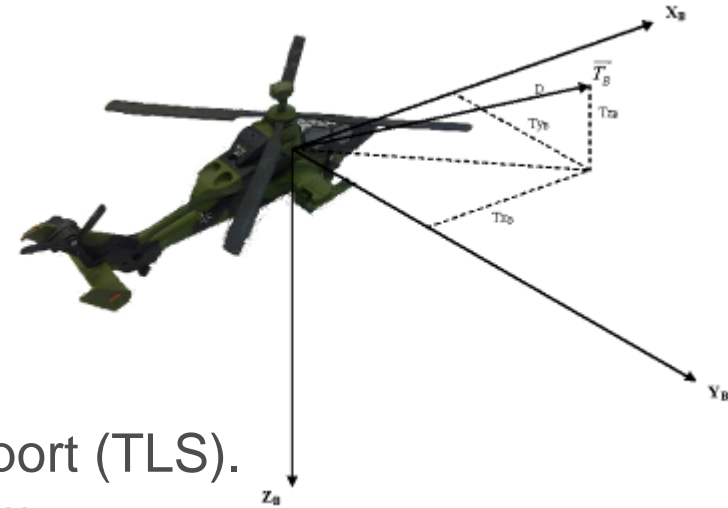
Slide 0010: Background

AA Platforms

- ARH Tiger
- MRH90

ARH Tiger

- In early stages of Through Life Support (TLS).
- Established Systems Safety Program.
- Established Software Assurance processes.
- Basic system (Nav, Coms, PFD, etc...) level 2A Sw.
- Basic system SRS 3000+ pages.
- Weapons system (Gun, Rockets, Hellfire, etc...) level 2A Sw.
- Weapons system SRS 1500+ pages.
- Trial run of Software Safety for current Sw development cycle.



Slide 0011 : The need for Safer Software

— Why do we need Safer Software?

- Increasing software control of safety critical systems.
- Constant growth of software intensive systems.
- The hardware it controls can cause harm.

Slide 0100: What currently happens

System Safety

- Looks at a system and defines the assurance levels required.

Software Assurance

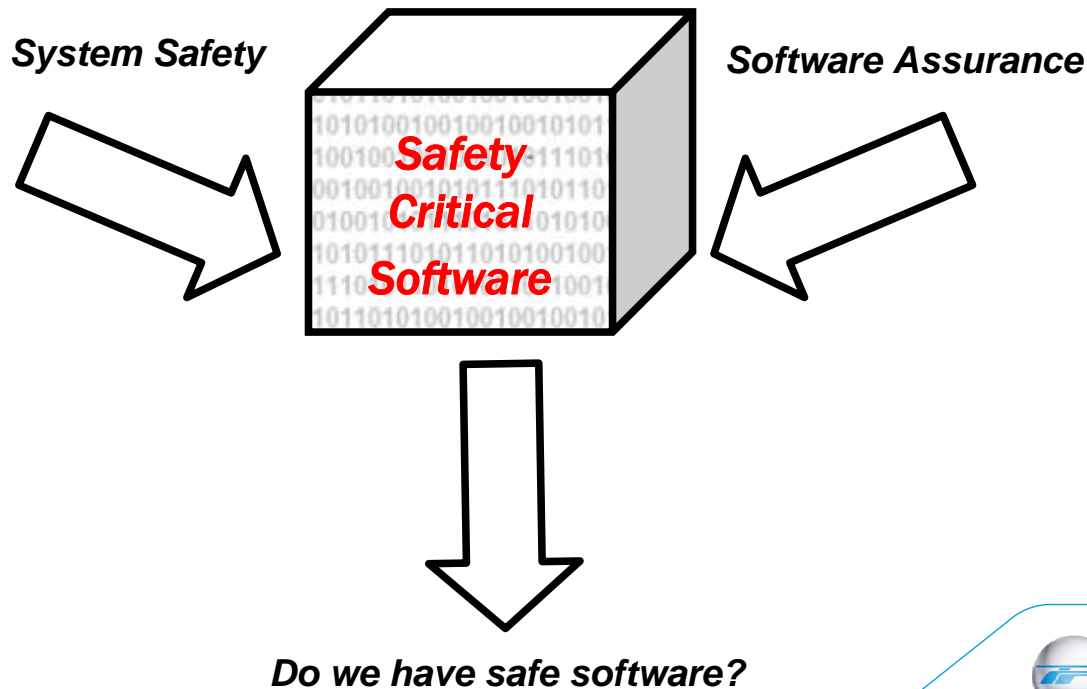
- Defined the processes required to develop software that meets the assurance level.



Slide 0101: What are we left with

The Gap

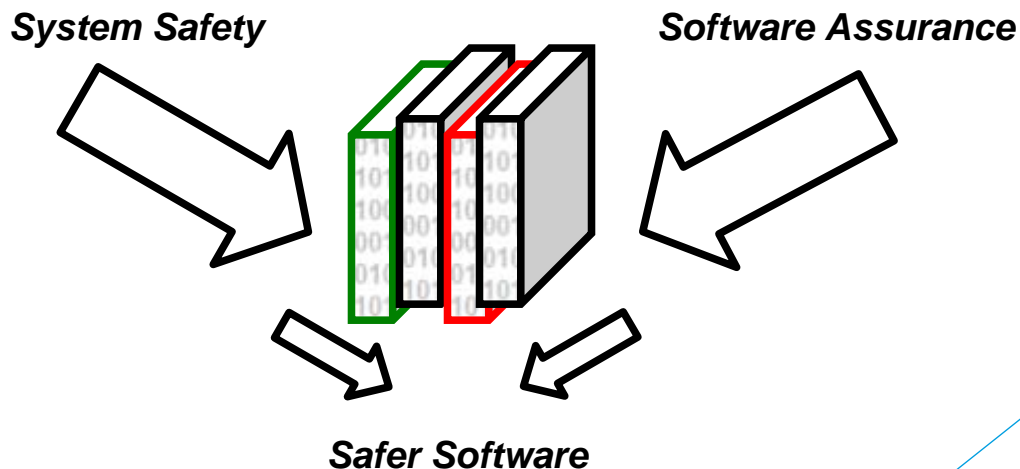
- What is the critical functionality that caused this required level of Software Assurance?
- Just because we have sufficient confidence that the software meets requirements does that indicate the system is safe?
- How can we bridge the gap from System Safety and Software Assurance?



Slide 0110: What is Software Safety

What is Software Safety?

- Software Safety (SwS) is an analytical approach to identifying critical functionality of software and mitigating risk of failure to produce robust, fault tolerant software systems.
- Software Safety should be an integrated Systems/Software safety program.
- Software Assurance is a key component.
e.g. Coding standards, best practices



Slide 0111: More about SwS

— Fault Tree Analysis Technique

- Software Fault Tree Analysis (SFTA)
- Not a fault tree at code level.
- Use system Top Event as a starting point.
- Identify single points of failure and mitigate with requirements.

— Other things we looked at

- Software Failure Mode and Effects Analysis (SFMEA).
- JSSSEH: Has some good info in it.
- AMCOM 385-17: Good checklists.

— Hard to find relevant training in Australia

- Most training about Software Safety in initial design stages.
- Found good specific Software Safety training in USA. www.hcrq.com
- Needed to adapt Sws to a project in TLS phase.

Slide 1000: SwS in the ideal situation

— Software Safety in initial design stage

- SwS Combined with System safety during initial requirements definition/design stage of a projects.
- The benefits this has on system/software architecture.
- How Top Events should be defined with software considerations.
- Correctly define the critical parts of the software.
- Fault Tree Analysis is used to defined Safety Requirements.
- Safety Requirements mitigate the risk of a Top Event.

Slide 1001: Introduce SwS for ARH TLS

Information we need to build up

- Processes for Software Safety Program.
- Take the hardware Top Events and expand to a software level.
- Identify software Safety Requirements and trace to Top Event it is mitigating.
- Identify generic Safety Requirements.
- Develop Software Fault Trees.

Introduce SwS to the development cycle

- Analysis of software changes that impact safety requirements.
- Influence safety critical changes to maintain alignment with safety architecture.
- Focus engineering effort for safety related Sw changes.

Slide 1010: Extra benefits

Value-adding in TLS phase .

- Top Events and Safety requirements can be used to develop regression testing and how this can be beneficial for a delta qualification's.
- New Engineers are aware of critical functionality.
- Assists in the categorisation of newly discovered problems.

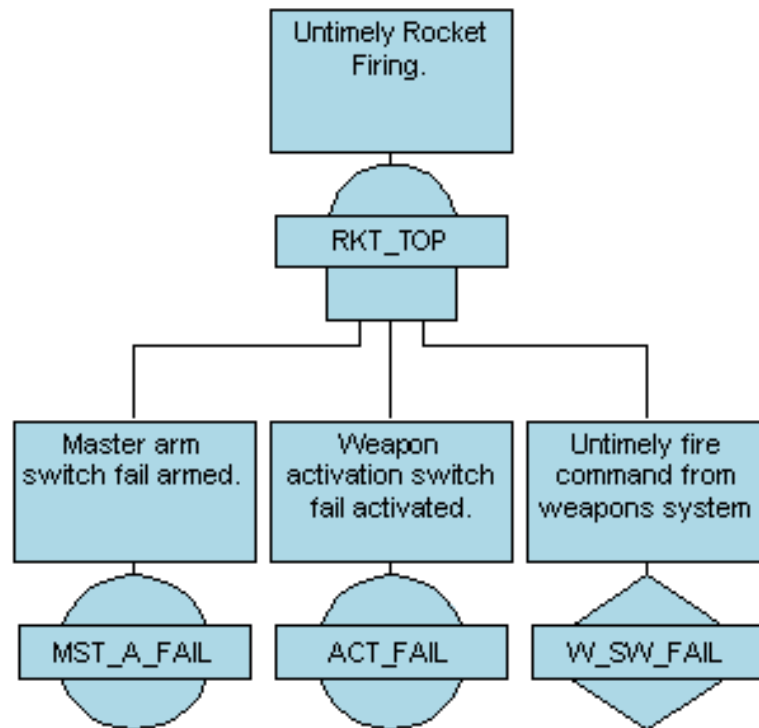
Slide 1011: Differences between Sw Safety and Sw Assurance

- Software Assurance relates to the level of confidence we have that SW meets requirements and is developed using best practices.
- Software Safety techniques reduces a systems sensitivity to faulty requirements / specs and software bugs.
- Software Assurance does not guarantee requirement are safe.

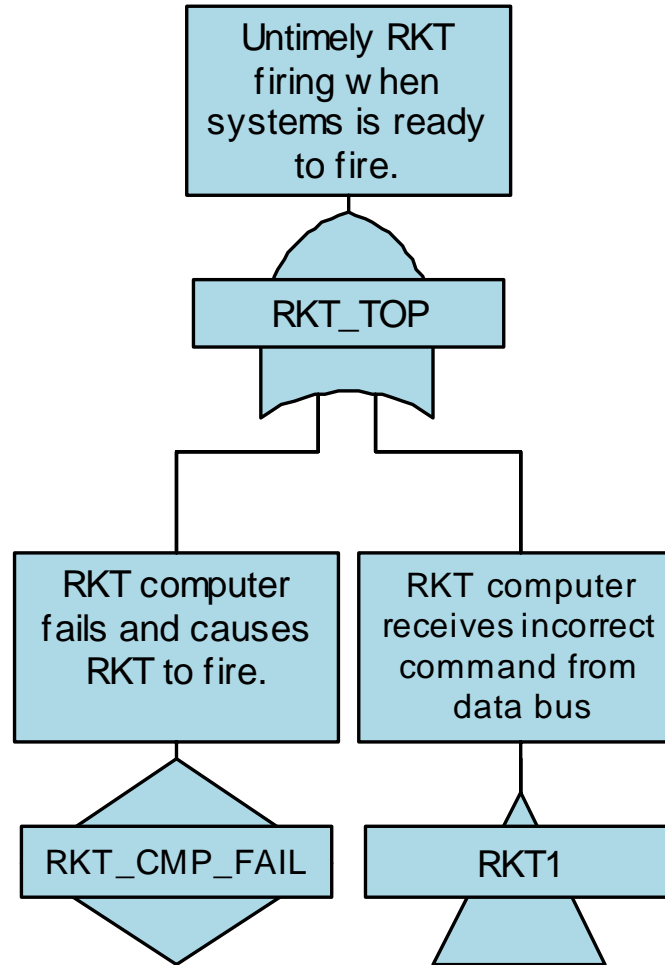
Slide 1100: Fault Tree Example

Rocket System.

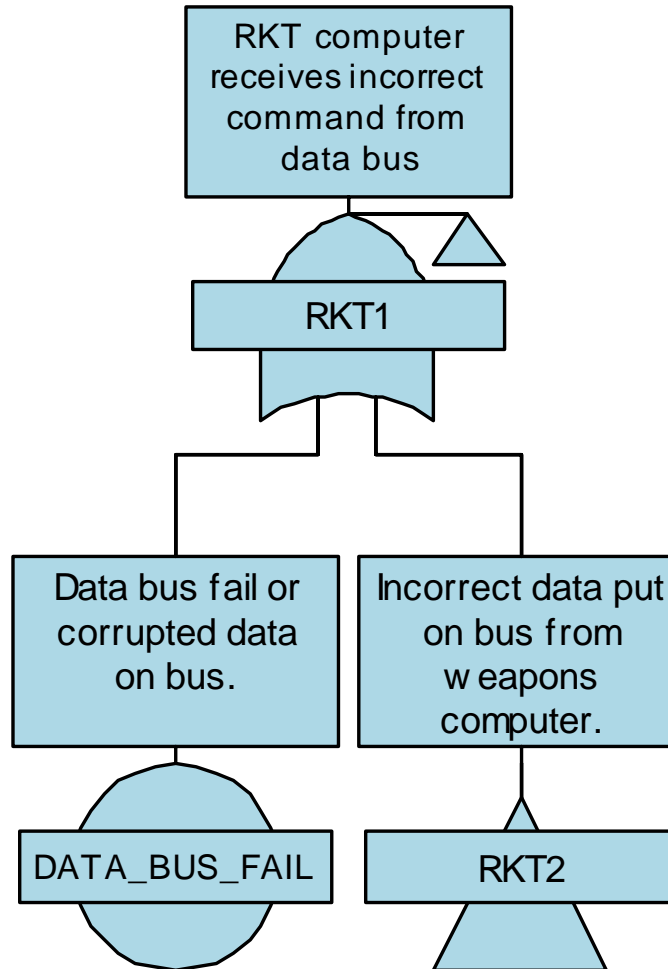
- Top Event: Untimely Firing



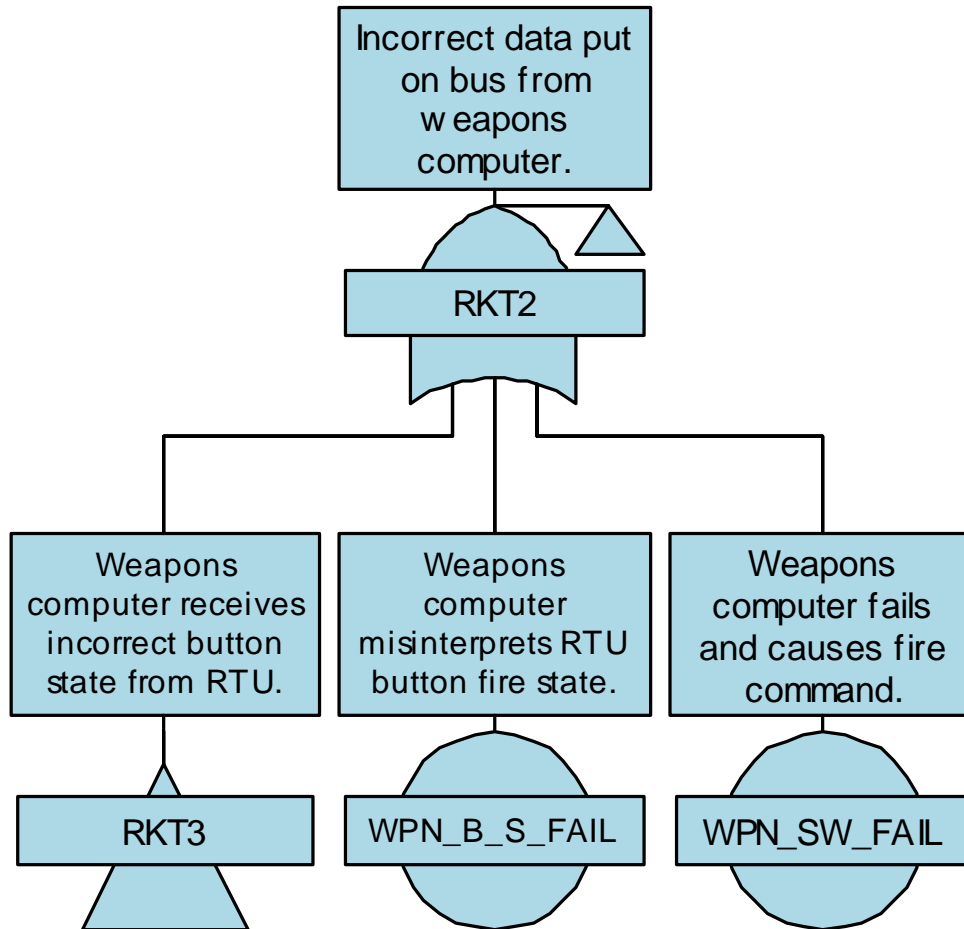
Slide 1101: Sw Fault Tree Example



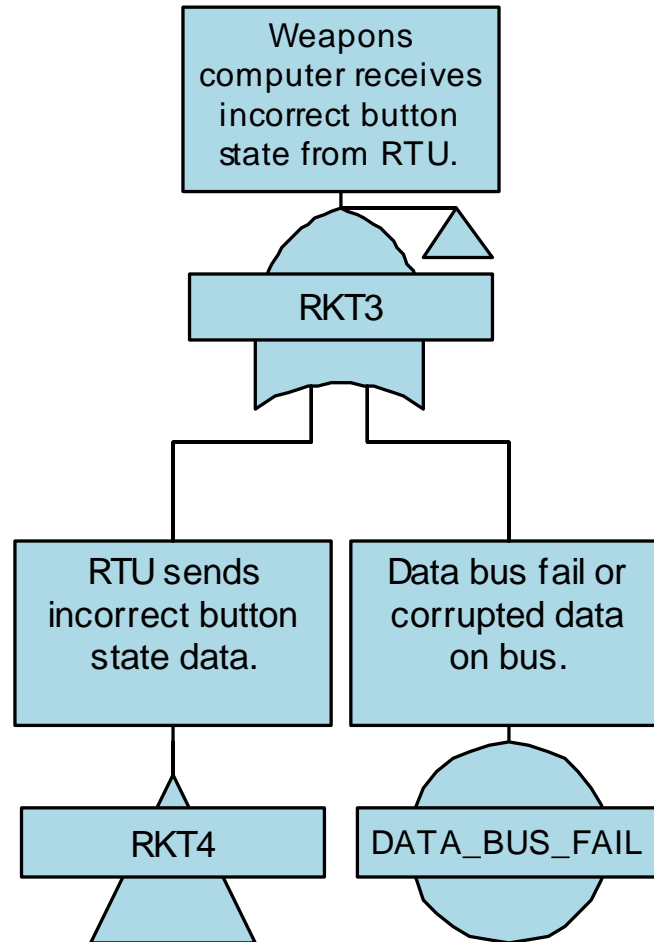
Slide 1110: Sw Fault Tree Example



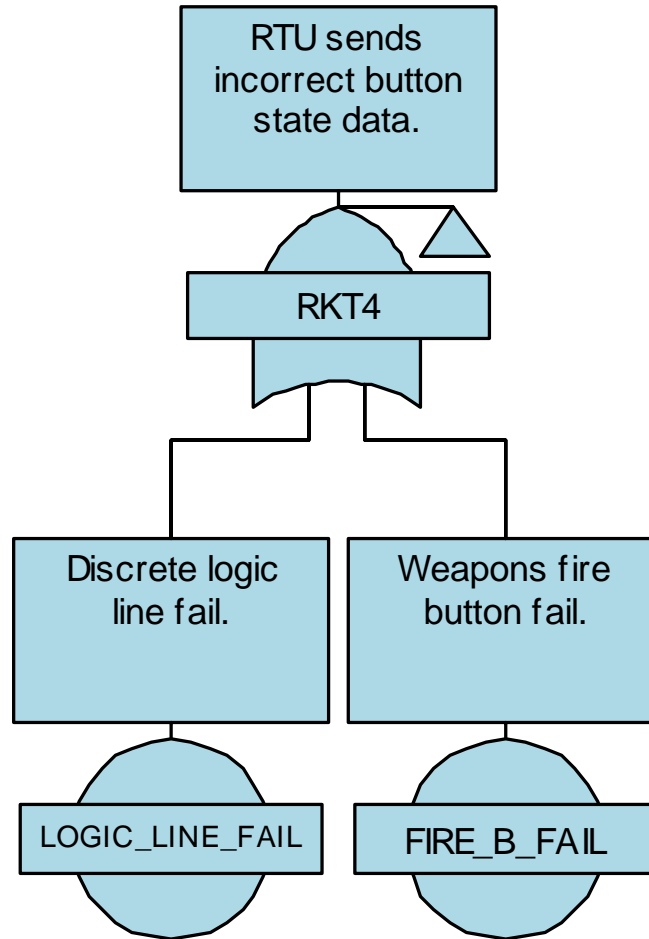
Slide 1111: Sw Fault Tree Example



Slide 10000: Sw Fault Tree Example



Slide 10001: Sw Fault Tree Example



Questions



Email: tcervenjak@ausaero.com.au