AdaEurope 2012

SW RELIABILITY PANEL – A PERSPECTIVE FROM INDUSTRY

INNOVATING SOLUTIONS

Reliable Software -1

- We know how to do 99% reliable software systems
  - Standards, recommendations, rules…
  - Guidelines, Methods and Techniques for the assessment of the software dependability and safety.
- Growing SW complexity, cost and schedule constraints!
- Where do the failures come from?

Reliable Software -2

- Failures are caused by different reasons:
  - Software requirements are incorrect, incomplete or ambiguous.
  - Software requirements have not been implemented, validated and verified properly.
  - Software has not been tested enough or has been tested inadequately.
  - Software Defects.
  - Software is used incorrectly.
  - Poor design or implementation.
  - Rare events can lead to uncontrolled states.

Reliable Software -3

- Failures due to inadequate designs or implementations are easier to detect and solve.
- Unexpected behaviours or states due to different failures are more difficult to detect and therefore to solve
- Analysis of consequences of failures:
  - Any single or combination of failures does not cause critical or catastrophic consequences.
- How to make software as fault tolerant as possible?

Reliable Software -4

- Single Points of Failure (SPF), a part of a system which, if it fails, will stop the entire system from working, shall be minimised.
- The implementing fault tolerant measures can be done at the following levels:
  - System-level measures: redundant networking equipments, redundant storage, a carefully planned monitoring, operational and maintenance strategy...
  - Component-level measures: using fault tolerant components, Error Correction Code (ECC) memory, redundant fans and networking components ....
  - Operating Systems and SW environment measures
  - SW development

Reliable Software -5

- DO-178B Partitioning: (1) independence of software components and processes, and (2) error containment.
- Design and coding practices:
  - Design partitions: different safety levels software running on the same processor
  - Design diversity: different failure strategies
  - Design as simple as possible
  - Defensive layers
  - Dynamic reconfigurations
  - Specify recovery actions
  - Structured programming, modularization, etc. Etc. Etc

Reliable Software -6

- Failures exist
- 99,xxx% SW Reliability
- Minimise Single Points of Failures (SPF)
- Design and coding practices for fault tolerance