ESARR 6 SWAL3 Compliant Software Development

Lessons learned in the SDDS project
ES2 WS3 Software Safety and Degraded Modes of Operation
Hans de Haan

EUROCONTROL

The European Organisation for the Safety of Air Navigation

Project context

- Surveillance Data Distribution System (SDDS)
- High performance data distribution system (60.000 – 1.200.000 tracks per second)
- Data validation, filtering an conversion capabilities
- Build on multi-purpose IP communication platform

Lesson 1: Getting Safety Requirements

- It proved to be very difficult if not even impossible to get clear requirements on the reliability and availability of the SDDS
- Only very general statements were given:
  - No single point of failure
  - Availability of 99.999
  - SWAL 3 compliant
- None of these “requirements” were backed up by a quantitative analysis
- Especially SWAL 3 seemed to be the result of the following reasoning:
  - SWAL 4 is not good enough and SWAL 2 is to expensive, let’s go for SWAL 3

Lesson 2: Determination of SWAL

- The required SWAL depends highly on the local deployment such as:
  - The existence of a backup
  - Diversity of the backup solution
- Analysis showed that in the majority of cases, SWAL4 would be sufficient
- Nevertheless the system was developed according to SWAL3 guidelines for two reasons:
  - General acceptance for SWAL4 was low
  - SWAL3 gives the flexibility to deploy the SDDS both in the main as backup chain

Lesson 3: Third party software

- ESARR 6 and accompanying documentation gives insufficient guidance related to third party software.
- Result:
  - The use of third party products (i.e. Tomcat, Apache) was abandoned
  - Third party libraries are built from source code and subject to same quality process as application
  - Java virtual machine treated as part of OS
- Conclusions:
  - It is almost impossible to validate third party SWAL3 compliance
  - Conflict between safety and cost efficiency is highly visible in this area

Lesson 4: Design is Underestimated

- ESARR 6 focus is in the following areas:
  - Requirements
  - Validation
  - Configuration management
  - Traceability
- In general, however, the software quality is mainly determined by design and implementation
- In the SDDS project, traceability was extended to design and implementation was monitored by applying advanced metrics.

Lesson 5: Use the Right Analysis

- FTA and RMA are not sufficient
- Use FMECA to check design and implementation
- Use FTA to develop operation procedures
- RMA is of limited value as it is focussed on hardware failures

Lesson 6: How to Prove Software Safety

- Answer:
  - We don’t know
- Reason:
  - The nature of safety related events (low frequency)
- Approach chosen:
  - Advanced software metrics to measure quality (complexity, programming practices, documentation)
  - Test automation which allows to expand the variance of tests performed

Lesson 7: Cost of SWAL3 development

- SWAL3 development requirements are comparable to medium level commercial software development
- Additional cost factors are:
  - Safety documentation (app 5%)
  - Additional development as result of third party software usage restrictions (may add up to 20%-100+% depending on type of development)

Conclusions

- ESARR 6 compliant development is possible at a reasonable cost.
- Improvement areas:
  - Better initial safety requirements
  - Better guidelines for third party software use
  - Incorporation of design and implementation in SWAL3 process
  - Development of validation strategies