Advances in Category-A Flight Software Production

Application to I-HAB ASW

Andoni Arregi

EDHPC, 2025-10-17

Content

1. Background

2. ECSS Cat-A Guidelines & Tools

3. Challenges & Advances

4. Takeaways

Background

Category A FSW Qualification in the Past

No systematic approach in the past

- The ATV MSU and the Orion ESM PDE
- Both successful in their function
- Some wrong assumptions in their Cat-A FSW approach

Category A FSW Qualification in the Past

What was done wrong?

- If object code coverage, then MC/DC
  $\rightarrow$ **false!**
- If I have a trace from object code to a source line I verified object code traceability
  $\rightarrow$ **false**

ECSS Cat-A Guidelines & Tools

ECSS Category A Guidelines

What was missing: some clarity and tools

- MC/DC structural coverage
- Object to source code traceability and coverage

2023: GTD published ECSS Cat-A Guidelines and Tools

- Guidelines to promote Cat-B flight-software to Cat-A
- Prototype open-source tools to apply the guidelines

Challenges & Advances

Challenges the Guidelines and Tools Faced

The real world of Cat-A software qualification

- Only SPARC, maybe ARM, and RISC-V?
  → No! we also have PPCs without an FPU, TI MSP430, etc.
- Compiler and OS particularities
  → For some targets we need to switch to old compilers
- Many separate object files instead of a monolithic FSW
- No access to SVF for the software developers
  → Emulators are the only option for daily work
- Integration with mandatory proprietary tools
  → Proprietary tools not always happy to cooperate outside their ecosystem

A very different computing environment

- A processor infrequently used in European space
- VxWorks operating system
- Older GCC compiler
- NASA cFS based FSW architecture
- Propietary unit testing environment
- Proprietary emulator based SVF

I-HAB application software current status:

- Many individual modules instead of a monolithic FSW binary
- Code size in relation to ATV MSU:
  - ~ x10 requirements
  - ~ x10 functions
  - ~ x5 executable statements
- ~3000 test cases

I-HAB application software current status:

- Many individual modules instead of a monolithic FSW binary
- Code size in relation to ATV MSU:
  - ~ x10 requirements
  - ~ x10 functions
  - ~ x5 executable statements
- ~3000 test cases

Good news

Continuous Integration pipeline runs nightly in 3 hours including:
→ MC/DC verification and object code coverage gathering

How the guidelines and tools can adapt

- Adapted the toolchain to a very different processor
- Set up a representative open-source emulator based on QEMU
- QEMU based object code coverage in Continuous Integration
- Adapted the guidelines to flight software composed by many non-linked object files
- Integrated proprietary unit-testing environment to enable object code coverage assessment

Advances

The open-source tool landscape has evolved significantly:

Towards an affordable Cat-A FSW process

- Available open-source tools for MC/DC:
  - GTD MC/DC checker
  - `gcov` now capable of MC/DC (since GCC 14)
  - Clang/LLVM also capable of MC/DC
- And for object to source code traceability verification and coverage:
  - OCCTRE and OCGRAPH for on-target object coverage tracing and verification
  - QEMU based coverage tracer for Continuous Integration

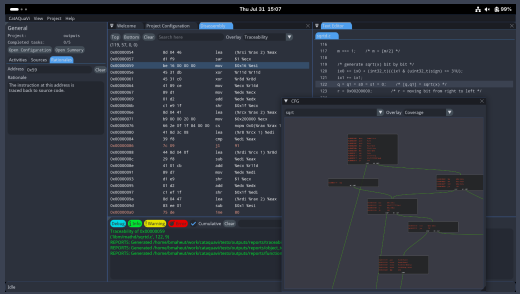
Advances

The maturity and applicability of the tools too:

Towards an affordable Cat-A FSW process

- DO-330 tool qualification ongoing to Tool Qualification Level 5
- Tools can be integrated in Continuous Integration systems
- We are expanding the support for more target architectures

Advances



Towards an affordable Cat-A FSW process

- We are developing a GUI suite to enable a wider use:
  - It guides the user trough the process
  - Hides the CLI tools complexity
  - Enables SW product assurance engineers, managers, and other stakeholders to assess the Cat-A process

Takeaways

Takeaways

Challenges

- Real world Cat-A projects might differ a lot from the nominal European SPARC+RTEMS combination

Solutions and Advances

- Both the Guidelines and the Tools can be adapted to many software environments
- More open-source tools than ever
- Tools being qualified to DO-330
- GUI suite available to enable all stakeholders

Takeaways

Challenges

- Real world Cat-A projects might differ a lot from the nominal European SPARC+RTEMS combination

Solutions and Advances

- Both the Guidelines and the Tools can be adapted to many software environments
- More open-source tools than ever
- Tools being qualified to DO-330
- GUI suite available to enable all stakeholders

Cat-A is not so expensive if you do it in Continuous Integration!