



Open-Source RTOS Space Qualification

An RTEMS Case Study



Scott Zemerick
Systems Engineer
TMC Technologies
NASA's IV&V Program





NASA RTOS Usage



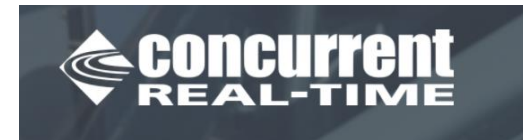
VxWorks

VxWorks 653



INTEGRITY

μC/OSTM
RTOS and Stacks





Open-Source RTOS Considerations



Open-Source Advantages

- Projects desire to own/control all source code forever
- Not locked into specific vendor/version for many years
- Not locked into proprietary build environment / tools
- Don't have to pay third-party for upgrades / new features
- Successful heritage and flight experience
- No budget for purchases


Open-Source Barriers

- Unable to be (easily) flight qualified
- No artifacts, little or limited documentation, no test cases
- Lots of internal development / testing may be needed
 - Example: custom drivers
- Nothing is guaranteed to work “out of the box”
- Could require more testing than COTS
- Many forks, no central/core version, fixes/features not fed back to project





Terminology

- Qualification vs Certification
 - *The process of developing and documenting quality software by utilizing a formal process and artifact generation*
- **Pre**-Qualification is *jump start on qualification with core artifacts and processes*  focus
- Final “Flight” Qualification
 - Performed on specific flight board/system
 - Qualified to a chosen standard
 - Examples: DO178-B/C, NASA 7150.2B
 - Tested and documented





Open-Source Qualification Example

- Core Flight System (CFS) Class A Certification
 - Performed by JSC for the Orion Program
 - LEON3/VxWorks
- Certification Included
 - Full coverage UT-assert unit test cases
 - API unit tests
 - Vertical integration tests
 - Test matrix, test plan, procedures, test report
 - VDD, User's Guide
 - Code inspections, static analysis
 - Coverage analysis results

Lorraine Prokop, Ph.D.
Software Manager, Advanced Exploration Systems
Avionics & Software Project NASA
Johnson Space Center (JSC)
October 2015





Research Goals

- **Increase the quality and maturity of open-source RTOS by identifying a lean, mean, PRE-qualification process**
 - Process should be driven by standard(s)
 - Process has to be simple, not overwhelming, leverage existing / free tools, and not scary
- **Processes are scary for open-source projects**
 - Limited resources
 - No time, money, expertise, or manpower
 - Not agile – too rigid for open-source paradigm



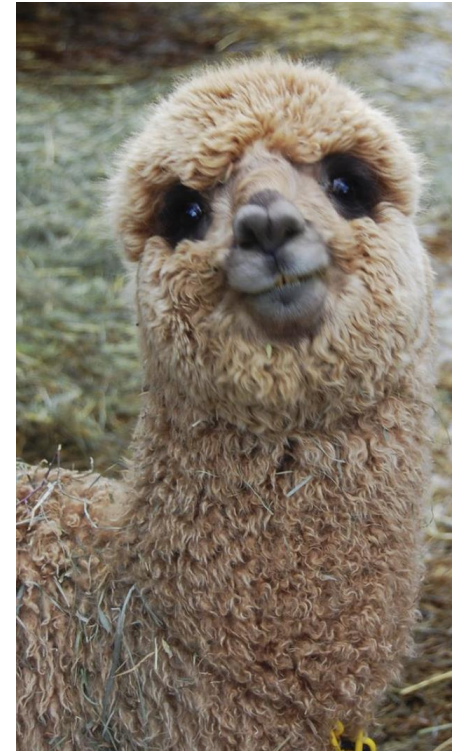
7150.2B



Research Goals



- **Imagine: FSW Lead on New Mission**
 - What RTOS? Open-source or COTS?
 - IF Open-Source:
 - Flight heritage?
 - Hardware profiles? LEON3/4, RAD750?
 - Maturity?
 - Flight Qualification Possible?
- **Maturation Metrics**
 - How to measure maturity?
 - How to measure software quality?
- **Choose open-source RTOS that is pre-qualified**
 - Ease and jump-start the qualification process
 - Review state of the open-source RTOS
 - What is complete? Tested?
 - What holes are missing? What needs tested?



Pre-Qualification
Provides a Warm-and-Fuzzy RTOS Choice

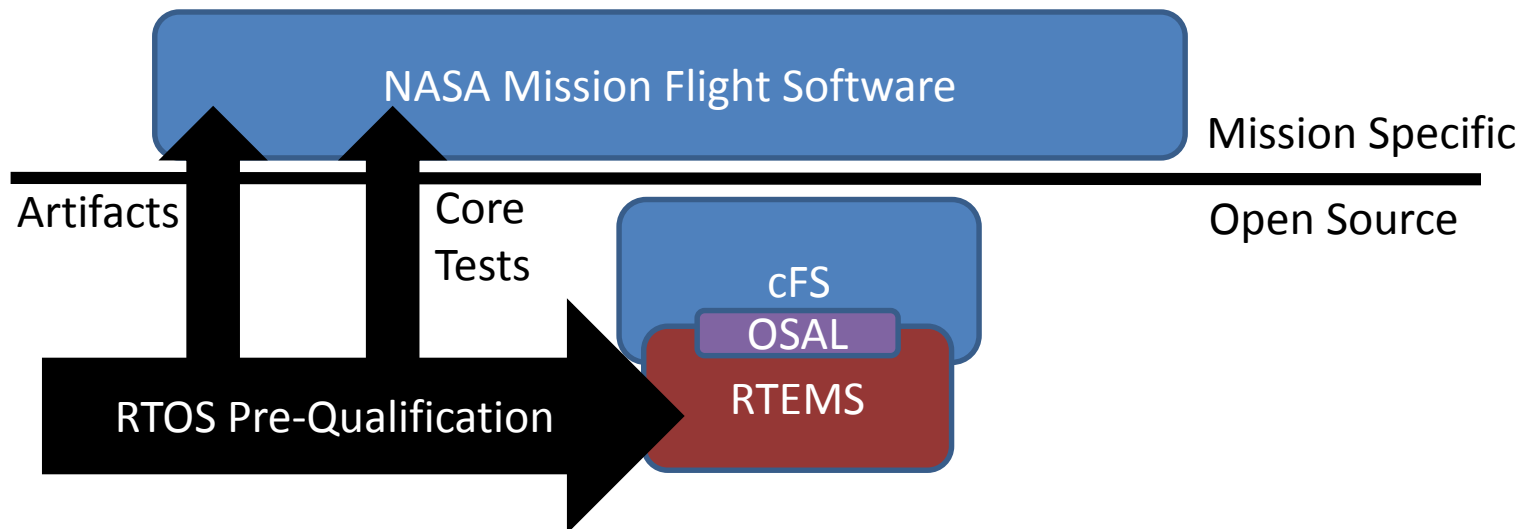




Research Goals

GSFC Flight Software Open Source Flight Software Stack

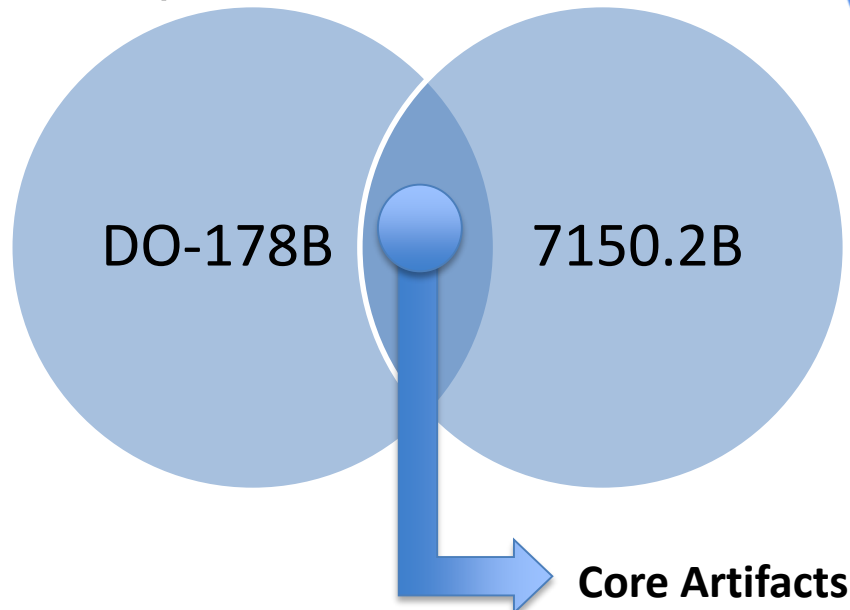
- Completely Open Source Flight Software
- “Qualifiable” due to this research
- Applicable to both small and large NASA missions





Research Tasks

- **Task 1:** Find DO-178B and 7150.2B Overlap
- **Task 2:** Review Overlap and Trim to Core Artifacts



- **Task 3:** Generate *Core-Artifacts-List*
- **Task 4:** Apply *Core-Artifacts-List* to Open-Source Project





Core-Artifacts-List

Category	Artifact	Artifact Intent
Requirements	Software Requirements Specification	Documentation of software requirements
	Requirements Test and Traceability Matrix	Maintain bidirectional traceability between the software requirement and the higher-level requirement.
	Software Assurance Plan / Validation	Requirements validation to ensure that the software will perform as intended in the targeted environment.

- Provides a pre-qual starting point
- Friendly names
- Should not be surprising
- Category-based
- “Hidden” pre-qual with focus on process and testing

Design and Implementation	Software Development or Management Plan	The Software Development Plan includes the objectives, standards and life cycle(s) to be used in the software development process.
	Software Configuration Management Plan	To identify and control major software changes, ensure that change is being properly implemented, and report changes to any other personnel or clients who may have an interest.
	Implementation	Implement the software design into code which is maintained in a version control system.
	Coding Standards Report	Software coding methods, standards, and/or criteria are adhered to and verified.
	Version Description Document (VDD)	Document that provides release information including versions, change history, and dependencies.

Testing and Software Assurance Activities

Software Test Plan	Document describing the testing scope and activities.
Software Assurance / Testing Procedures	To define the techniques, procedures, and methodologies that will be used.
Software Change Report and Problem Report	Reviews of software activities, status, and results with the project stakeholders and track issues to resolution.
Software Schedule	Project milestone and schedule is updated accordingly.
Software Test Report / Verification Results	Record, address, and track to closure the results of software verification activities.

Usability

Software User's Manual	Software User Instructions
------------------------	----------------------------

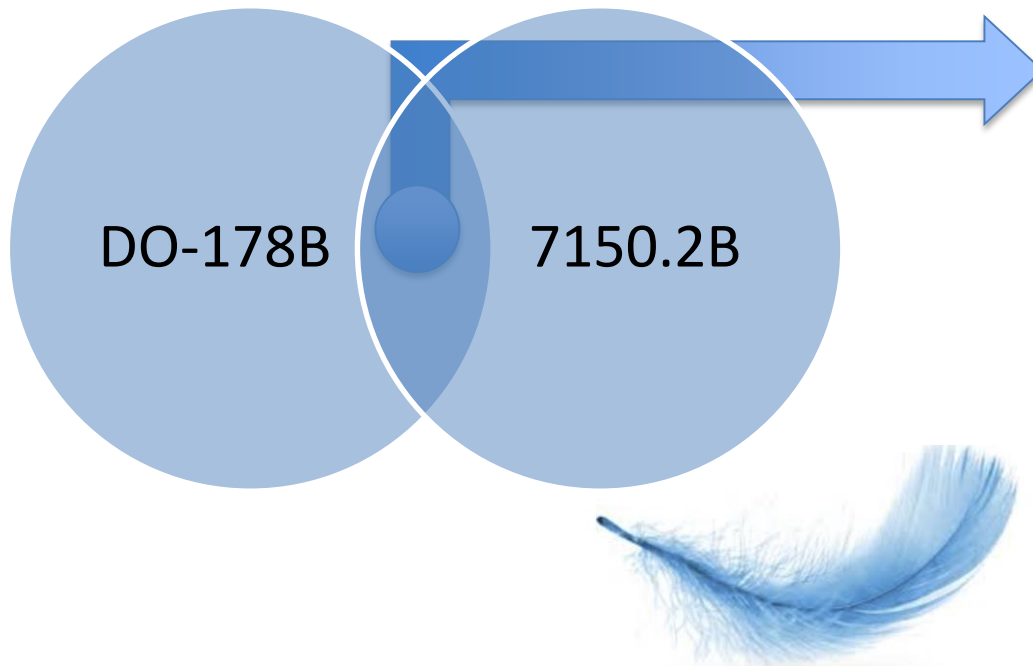




Applicable to RTEMS



- **Task 4:** Apply *Core-Artifacts-List* to Open-Source Project
- Chose RTEMS
 - Significant NASA / ESA flight heritage
 - Professional, well-managed open-source project
 - Desires to incorporate pre-qualification into their open-source process – but can't be a burden – wants a lightweight process
 - Has some existing processes, tests, documentation in place



"RTEMS Software Engineering Standards" Template

- 1. Introduction to Qualification / Purpose**
- 2. Software Development Management**
 - a. Implementation Details
 - b. Coding Standards
 - c. Change Management
 - d. Issue Tracking
- 3. Software Test Plan Assurance and Procedures**
 - a. Scope, Procedures, Methodologies, Tools
- 4. Software Release Management**
 - a. Software Change Report Generation – Review process, workflows, etc
 - b. Version Description Document generation (generated by Issue Tracker)
- 5. User's Manuals**
- 6. Licensing Requirements**





Applicable to RTEMS



"RTEMS Software Engineering Standards" Template

1. Introduction to Qualification / Purpose ✓
2. Software Development Management
 - a. Implementation Details
 - b. Coding Standards
 - c. Change Management ✓
 - d. Issue Tracking
3. Software Test Plan Assurance and Procedures
 - a. Scope, Procedures, Methodologies, Tools
4. Software Release Management
 - a. Software Change Report Generation – Review process, workflows, etc ✓
 - b. Version Description Document generation (generated by Issue Tracker)
5. User's Manuals
6. Licensing Requirements ✓

Next Steps

- Dive into more details
- Provide scorecard on areas that can be improved
- Leverage open-source tools to generate artifacts
- Think about requirements more



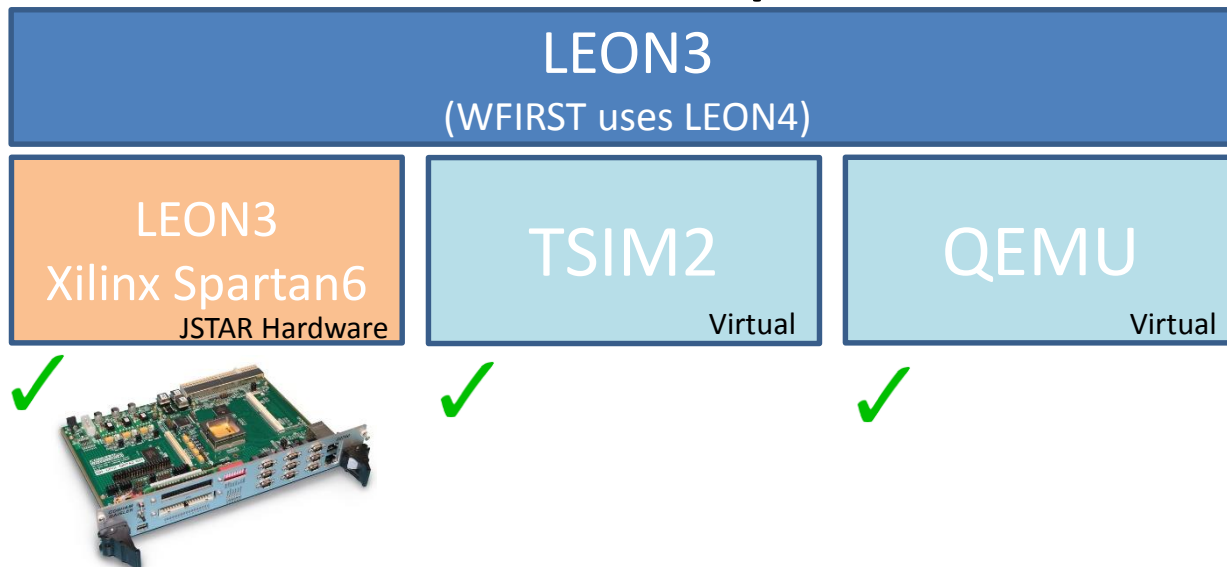


RTEMS Space Qualification Test Bed (REST)



- REST is a virtual environment with LEON3 instruction set simulator
- Goal will be a basic set of checkout tests and suitable for pre-qualification testing
- Repeatable test results

Pre-Qualification Space Profile



This is a non-ITAR presentation, for public release and reproduction from FSW website.





Ongoing FY18 Work

- Continue working with RTEMS community on pre-qualification
- Investigate RTOS security and how to assess
 - How much should we care about embedded RTOS security?
- Mature cFS CryptoLib and Release
 - CryptoLib implements SDLS procedures and allow for easy integration into existing CFS command ingest and telemetry output applications
 - Integrate into NOS3 – <http://www.nos3.org>

