Qualification of RTEMS Symmetric Multiprocessing (SMP)

- **Final Presentation Days**

07 & 08 December 2021

Qualification of RTEMS Symmetric Multiprocessing (SMP)

- Budget: 700K
- Duration: 2.8 years
- Consortium: Edisoft, Embedded Brains, LERO, Jena-Optronik, CISTER
- Main Objective:
  - This activity provides a QDP for the open source RTEMS real-time operating system with symmetric multi-processing capabilities. It is compliant with ECSS applicable standards for software engineering and software product assurance.

Background

- GSTP activity started in February 2019, consortium is composed of:
  - EDISOFT (Portugal – consortium lead) → RTEMS qualification experience, strong ties with industry
  - Embedded Brains (Germany) → RTEMS SMP development expertise, strong ties with community
  - LERO (University of Limerick, Trinity College Dublin, Ireland) → formal methods expertise
  - Jena-Optronik (Germany) → end user in space domain, application qualification expertise
  - CISTER (ISEP/P, PORTO, Portugal)→ real-time software and software qualification expertise

Objectives

- Facilitation of RTEMS SMP Qualification
- Reduce lifecycle of each release of “Qualified” RTEMS
- Apply Formal Methods Verification (e.g. for OMIP and MrsP Algorithms)
- Port RVS3000 to RTEMS SMP (Coordinate transformation of point cloud)

Qualification RTEMS SMP

Task: RTEMS SMP Qualification Data Package (Embedded Brains)

Sebastian Huber (Embedded Brains)

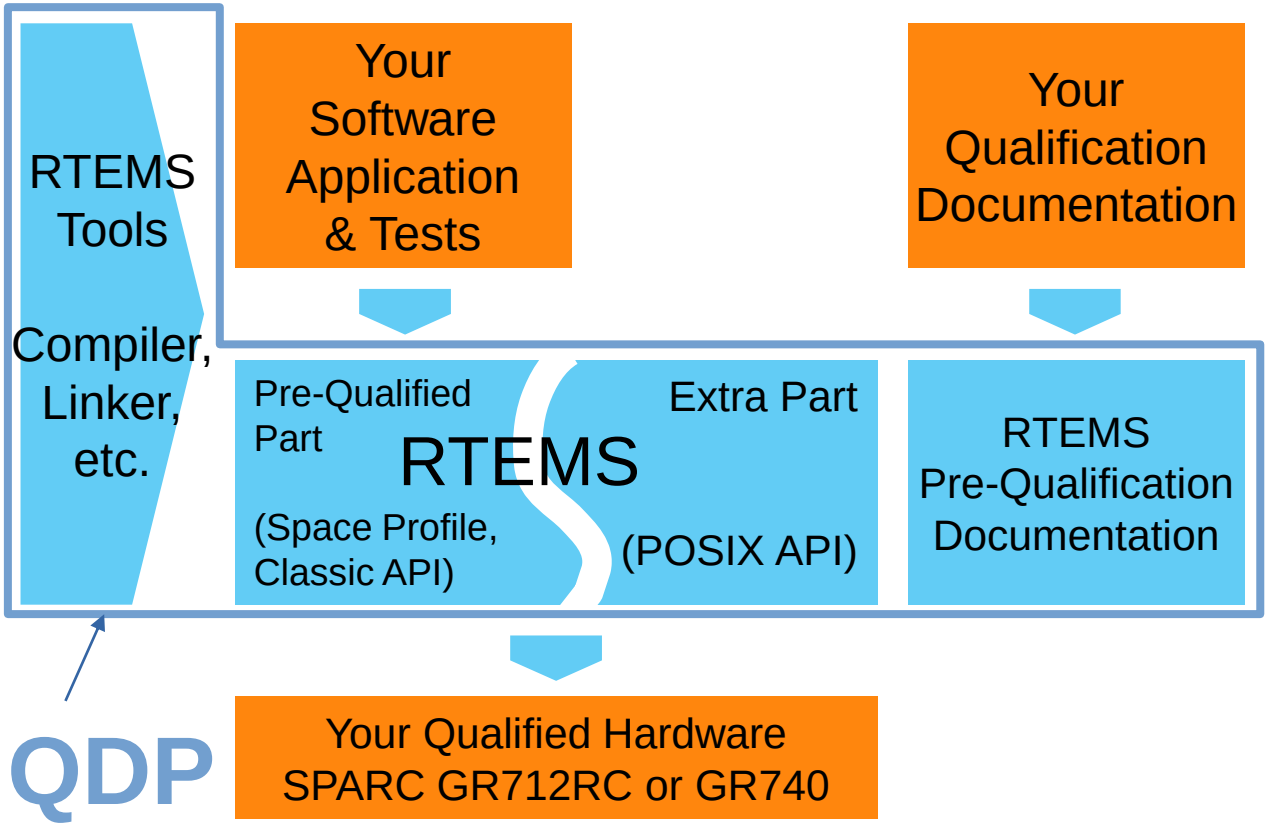
What is in the QDP?

- Delivery in **two parts**: SCF + archive file
- **SCF**: may be digitally signed, secure hash of archive, root of trust
- Binaries and sources of needed tools (compiler, linker, provided “as is”, ready to distribute)
- Pre-qualified part of RTEMS operating system
- Extra (not pre-qualified) part of RTEMS
- Documentation (RTEMS, ECSS, technical notes)
- Other stuff (sources, tests, Dockerfile)

What Do You Need to Get Your Application Qualified?

- Hardware: based on Gaisler GR712RC or GR740
- System requirements (what shall your application do?)
- Application software realizing those requirements
- Uses pre-qualified parts of RTEMS
- Built by tools from QDP
- Documents showing that ECSS standards are met by your application (may reference documents provided by QDP)
- Qualification authority to escort the development of your application (usually ESA)

How is the QDP used?



Pre-Qualfied RTEMS Features

- The Real-Time Executive for Multiprocessor Systems (RTEMS) is a multi-threaded, single address-space, real-time operating system with no kernel-space/user-space separation
- Thread synchronization and communication: mutexes, message queues, semaphores, events, barriers, signals, futex
- Locking protocols: transitive priority inheritance, priority ceiling, Multiprocessor Resource Sharing Protocol (MrsP), O(m) Independence-Preserving Protocol (OMIP)
- Clustered scheduling (SMP feature)
- Scalable timer and timeout support
- Lock-free timestamps (FreeBSD timecounters) with NTP support
- C11/C++11 thread-local storage

Which Documentation will be provided?

- **Software Configuration File (SCF)** – accompanies QDP, overview, content, tutorial, space profile – what has been pre-qualified?
- Standard RTEMS Documentation
  - **RTEMS Classic API Guide** – description of RTEMS and its API
  - RTEMS User Manual – how to use RTEMS?
  - RTEMS Software Engineering – how to maintain RTEMS?
- QT-109
  - Core document which includes the planning & content of documents
  - ECSS tailoring and compliance matrix
  - Analysis of other standards such as IEC 61508
- Interface Control Document (ICD) & Software Requirement Specification (SRS) – requirements
- Software Verification Report & SPAMR – verification documentation for qualification

Performed Software Engineering Activities

- Requirements engineering chapter for RTEMS Software Engineering manual
- Review of the complete source code of the pre-qualified RTEMS feature set
- Specification of pre-qualified feature set of RTEMS using the Easy Approach to Requirements Syntax (EARS)
  - EARS problem: lots of atomic requirements
  - Solution: table based specification with generated validation test code
- New build system for RTEMS using specification items
- Development of the RTEMS Test Framework
- Validation tests are embedded in the specification items, validation test code is generated
- More than 50000 atomic requirements are validated: 100% line and branch coverage at source code level on systems with at least three processors

Which Verification Activities have been performed?

Verification checks that all project activities meet ECSS standards

- Documents: svr.pdf and spamr.pdf in QDP
- Automated verification where possible
- Static Analyzer: Coverity, Clang Static Analyzer, Cppcheck
- Anticipated for 2022 (Kick-off December 2021): Independent Software Validation and Verification (ISVV) to meet Criticality Category B.

What is not Included in QDP?

- Training
- Support services
- Expert knowledge to
  - Customize the QDP
  - Support new architectures/BSPs
  - Extend the pre-qualified scope (e.g. POSIX, OpenMP)
- Long term maintenance

Service providers

Qualification RTEMS SMP

Task: Qualification Tool Chain (EDISOFT)

José Valdez (EDISOFT)

Objectives

- Preliminary work:
  - Define the space profile to be used for RTEMS
  - Standard analysis to assess the possibility to extend this work to other standards (GSWS, DO-178, ISO 26262 and IEC 61508)
  - Open source Tools identification that could be integrated in the Qualification Toolchain
  - Assess the possibility of reuse parts RTEMS Improvement QDP for this project
- Qualification Toolchain
  - RTEMS and RSB Compilation (provide to the end users already compiled binaries and testsuite automatic execution/analysis
  - Produce the ECSS documentation (as needed for Category B)
  - Produce a solution easy to maintain (with docker and CI)
  - Follow RTEMS community guidelines (to foresee a future integration)

Qualification Toolchain – Concept

Inputs:

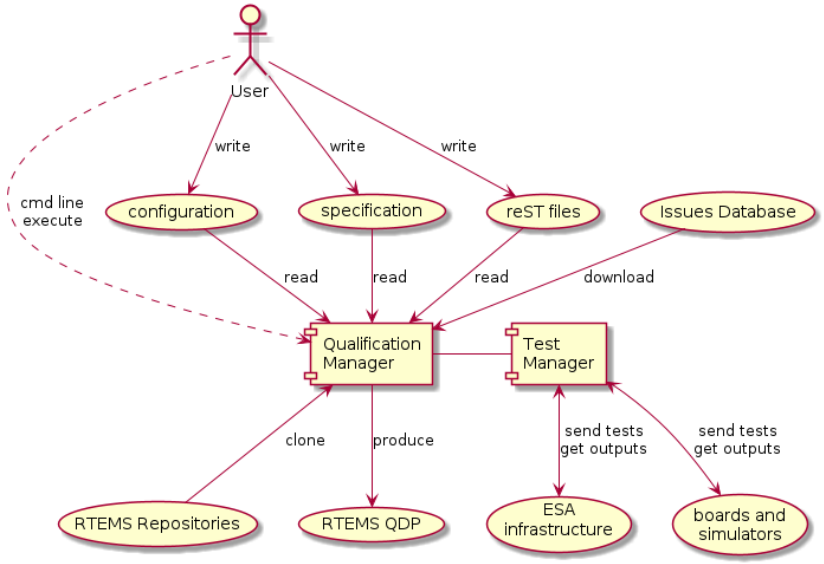
- Configuration files
- Specification files
- Sphinx files
- RTEMS Repositories

Output:

- RTEMS QDP

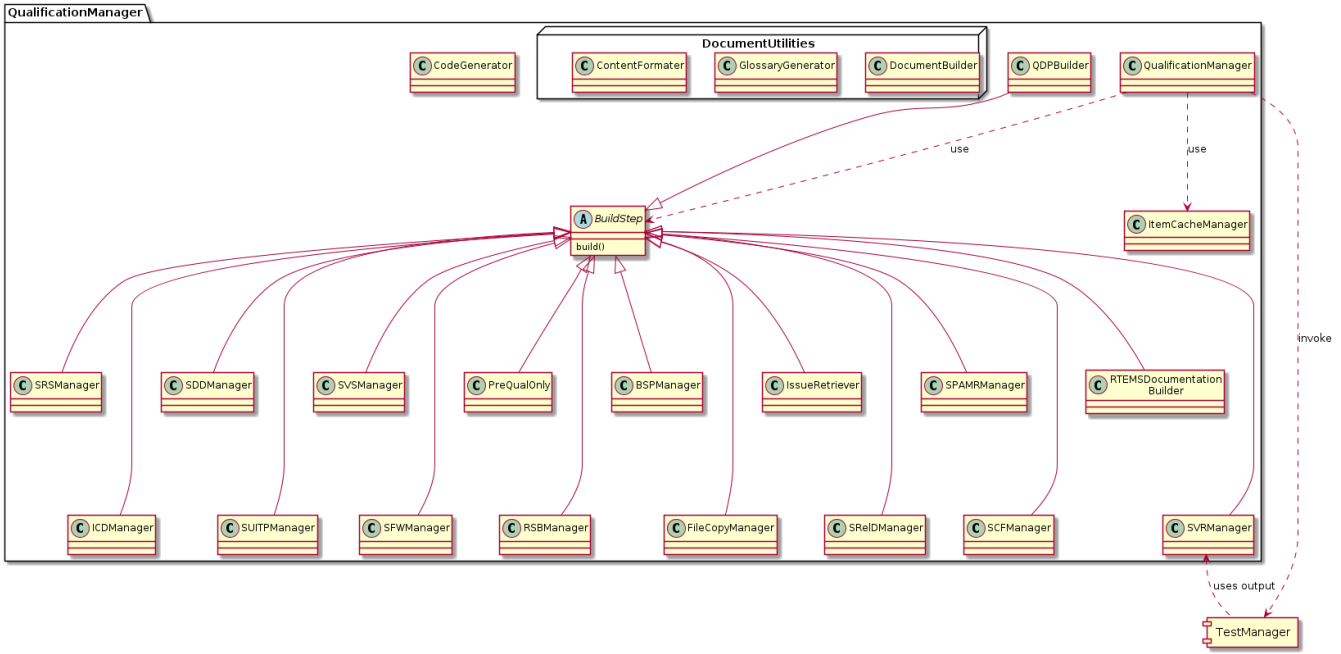
Main features:

- Qualification automatization
- Allows CI (via docker)
- Allows keeping up to date with RTEMS community
- Allows easy addition of features (ex: new BSPs)
- Automatic traceability
- Automatic RSB, RTEMS build, testsuite execution, result analysis and report generation



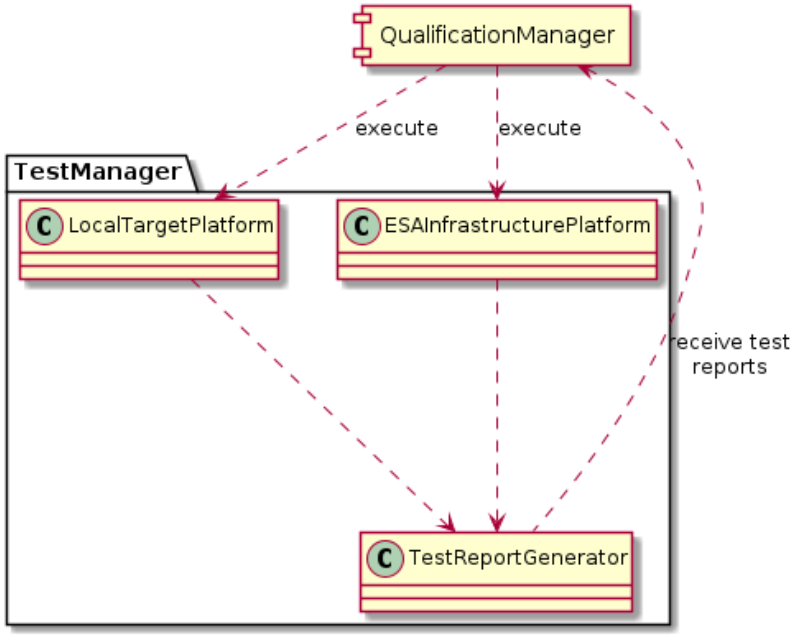
Qualification Toolchain – Qualification Manager

Components of the Qualification Manager (cat. D qualification)



Qualification Toolchain – Tests Manager

Components of the Test Manager (cat C. qualification)



Test Manager – Test Processing (1/3)

Requirements and test status summary table

Requirement	Test specification	Test name	Status
spec:/score/smp/req/fatal-multitasking-start-on-unassigned-processor	spec:/score/smp/val/fatal	ScoreSmpValFatal	*P*, *P*
spec:/rtems/task/req/perf-runtime	spec:/rtems/task/val/perf	RtemsTaskValPerf	*P*

By clicking on the status (“P”), it jumps to the full report

7.1.14.1 Test Case - ScoreSmpValFatal

In this test case 12 test steps were executed. All steps passed. The test case execution time was 0.000001s.

Listing 13: Test Log

```
B:ScoreSmpValFatal
E:ScoreSmpValFatal:N:12:F:0:D:0.000001
```

Test Manager – Test Processing (2/3)

Performance metrics status summary table

Requirement	Test Measurement	Status
/rtems/task/req/perf-construct	RtemsTaskReqPerfConstruct	*P*, *P*, *P*, *P*, *P*

By clicking on the status (“P”), it jumps to the full report

Runtime Measurement - RtemsTaskReqPerfConstruct (FullCache)

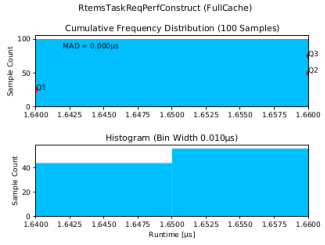


Table 1: Limits Specified by spec:/rtems/task/req/perf-construct vs. Actual Values

Limit Kind	Specified Limits	Actual Value	Status
Minimum	$1.312\mu s \leq$ Minimum	$1.640\mu s$	OK
Median	$1.328\mu s \leq$ Median $\leq 1.992\mu s$	$1.660\mu s$	OK
Maximum	Maximum $\leq 1.992\mu s$	$1.660\mu s$	OK

Test Manager – Test Processing (3/3)

Coverage summary html report:

GCC Code Coverage Report

Directory: ./
Date: 2021-12-02 00:49:36
Legend: low: >= 0% medium: >= 80.0% high: = 100%

	Exec	Total	Coverage
Lines:	8045	8059	99.8%
Branches:	2056	2061	99.8%

File	Lines			Branches		
bsps/include/bsp/fatal.h		100.0%	2 / 2	-%	0 / 0
bsps/include/bsp/irq-generic.h		100.0%	35 / 35	100.0%	6 / 6
bsps/shared/grlib/uart/apbuart_polled.c		100.0%	15 / 15	100.0%	4 / 4
bsps/shared/irq/irq-affinity.c		100.0%	20 / 20	100.0%	14 / 14
bsps/shared/irq/irq-default-handler.c		100.0%	2 / 2	-%	0 / 0
bsps/shared/irq/irq-enable-disable.c		100.0%	22 / 22	100.0%	12 / 12
bsps/shared/irq/irq-entry-remove.c		100.0%	25 / 25	100.0%	10 / 10
bsps/shared/irq/irq-generic.c		100.0%	69 / 69	100.0%	30 / 30
bsps/shared/irq/irq-handler-iterate.c		100.0%	13 / 13	100.0%	6 / 6
bsps/shared/irq/irq-lock.c		100.0%	8 / 8	100.0%	4 / 4
bsps/shared/irq/irq-raise-clear.c		100.0%	22 / 22	100.0%	14 / 14
bsps/shared/start/bootcard.c		100.0%	4 / 4	-%	0 / 0
bsps/sparc/include/grlib/io.h		100.0%	5 / 5	-%	0 / 0
bsps/sparc/leon3/clock/ckinit.c		100.0%	29 / 29	100.0%	2 / 2
bsps/sparc/leon3/console/printk_support.c		100.0%	21 / 21	-%	0 / 0
bsps/sparc/leon3/include/bsp/leon3.h		100.0%	30 / 30	100.0%	4 / 4
bsps/sparc/leon3/start/bspclean.c		100.0%	23 / 23	100.0%	16 / 16
bsps/sparc/leon3/start/bspsmp.c		100.0%	28 / 28	100.0%	4 / 4
bsps/sparc/leon3/start/bspstart.c		100.0%	9 / 9	-%	0 / 0
bsps/sparc/leon3/start/cache.c		100.0%	52 / 52	100.0%	3 / 3

```
48
49  ▶ 2/2   886   if ( attributes == NULL ) {
50              2     return RTEMS_INVALID_ADDRESS;
51                  }
52
53            884   memset( attributes, 0, sizeof( *attributes ) );
54
55  ▶ 2/2     884   if ( !bsp_interrupt_is_valid_vector( vector ) ) {
56               35     return RTEMS_INVALID_ID;
57                  }
```

Qualification RTEMS SMP

Task: RTEMS SMP Formal Verification (LERO)

Andrew Butterfield (Lero)

Objectives

- Explore the application of Formal Methods
- How would they best contribute to the QDP?
    - Which parts of RTEMS would benefit most?
- How would they best fit with RTEMS community principles?
    - Which formal methods and tools were most suitable?
- Deploy Formal Methods on a chosen set of features
    - Develop Formal Models
    - Perform Verifications
    - Develop supporting tools
- Focus: critical features such as synchronization primitives, multicore, atomics,..

Work Performed

- Task 3.1 Initial Investigation
  - RTEMS community issues: software footprint, and future maintainability
  - Promela/SPIN deemed most suitable
    - others investigated included Frama-C, TLA+, Isabelle/HOL
- Task 3.2 Detailed work
  - Explore different ways to produce Promela models
  - Aim: use models for test generation
  - Perform case-studies to develop the approach
    - Chains API – basic concepts, end-to-end, producing tests run on hardware
    - Event Manager – concurrency, multi-core, how to produce repeatable tests
    - MrsP ThreadQs – exploring modelling/testing for this critical component
- Task 3.3 Final Reporting

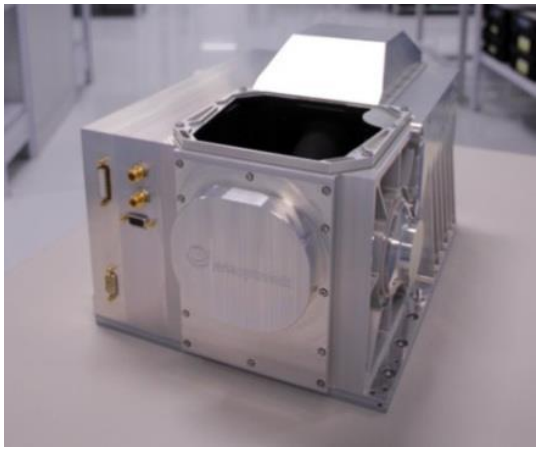
Qualification RTEMS SMP

Task: RTEMS SMP Application Porting (Jena-Optronik)

Olivier Ballereau (Jena-Optronik)

Objectives

- Migrate the RVS3000-3D device's software from EDISOFT RTEMS Improvement to RTEMS SMP to take advantage of the second gr712rc core
- Assess the ported application in terms of memory and performance
- Qualify RTEMS SMP in the Jena-Optronik gr712rc (run again the testsuite and re-generate the new Software Verification Report)



Work Performed (1/2)

- Task 4.1 - Application Description & Architecture
  - Description of the RVS3000-3D hardware and software model
  - Selection of the managers to use: Clock Manager, Event Manager, Interrupt Manager, Scheduler Manager, Semaphore Manager, Task Manager and Timer Manager
  - Selection of the scheduler: EDF scheduler with one-to-one and one-to-all thread to processor affinity
  - Selection of the locking algorithm: O(m) Independence-Preserving Protocol (OMIP)

Work Performed (2/2)

- Task 4.2 - Application Porting
  - Porting from GCC 4.2 to GCC 10.2, required some corrections (ex: double ‘const’ qualifier for a parameter, missing ‘extern’ qualifier when declaring a global variable in a header file,)
  - Migrate custom qualified math library to ESA MLFS
  - API Changes (ex: rtems_clock_get() replaced by rtems_clock_get_ticks_per_second()
  - Interrupt Locking: replace of rtems_interrupt_{enable,disable}() by rtems_interrupt_lock_{acquire,release}()/rtems_interrupt_local_{enable,disable}()
  - Conguration Changes (ex: disable Newlib re-entrancy)
  - Init Task: use new CONFIGCONFIGURE_INIT_TASK_CONSTRUCT_STORAGE_SIZE
  - Added SMP support in the application (ex: Communication and Synchronization)
  - Run the testsuite using GDB

Results and conclusions

- Task 4.3 Application Porting Report:
  - Memory overhead: newer versions of gcc introduce more code:
    - RTEMS 4.8 minimal application: 23812 bytes
    - RTEMS SMP minimal application: 68896 bytes
  - RTEMS SMP allowed a boost of 63% in time performance
  - QDP Testsuite run with test failures (under investigation) and the SVR generated successfully. Jena Optornik uses a gr712rc engineering model, whereas ESA uses a development board.

Qualification RTEMS SMP

Project Outcome

Sebastian Huber (Embedded Brains)

Project Outcome

Approach

- Code in line with the public (open-source) version of RTEMS 6
- Fully automated document generation and testing
- Application of Formal Methods for testing critical features

Results (for GR712 RC and GR740, based on “Space Profile”):

- Requirements added and code documentation completed
- Comprehensive validation test suite: Code optimization performed, line/branch coverage: 100%
- Tool chain to run tests and to produce QDP
- Use case test on GR712RC based OBC
- Criteria for pre-qualification according to Criticality Category C matched
- Formal Promela Models of selected RTEMS features used for Test Generation

Outlook

- Independent Software Validation and Verification started (→ Criticality Category B)
- Further support (Training, functional extensions) available by expert services

Contacts

- QDP maintenance and questions:
  - EDISOFT: nuno.ramos@edisoft.pt / jose.valdez@edisoft.pt
  - Embedded Brains GmbH: rtems@embedded-brains.de
- Formal methods questions: andrew.butterfield@scss.tcd.ie

Questions