FUSION FOR ENERGY

MARTe2 Users Meeting
Quality Assurance & collaboration models with-in the community

A. Neto, F. Sartori

May 2019

Why a new MARTe project?

- Fast controller prototyope project
  - Integration of fast plant systems in ITER
- Development of ITER specific integration components
- Imposes the implementation of a Quality Assurance (QA) strategy that is appropriate for ITER

Development and upgrading of framework components will be distributed to different teams

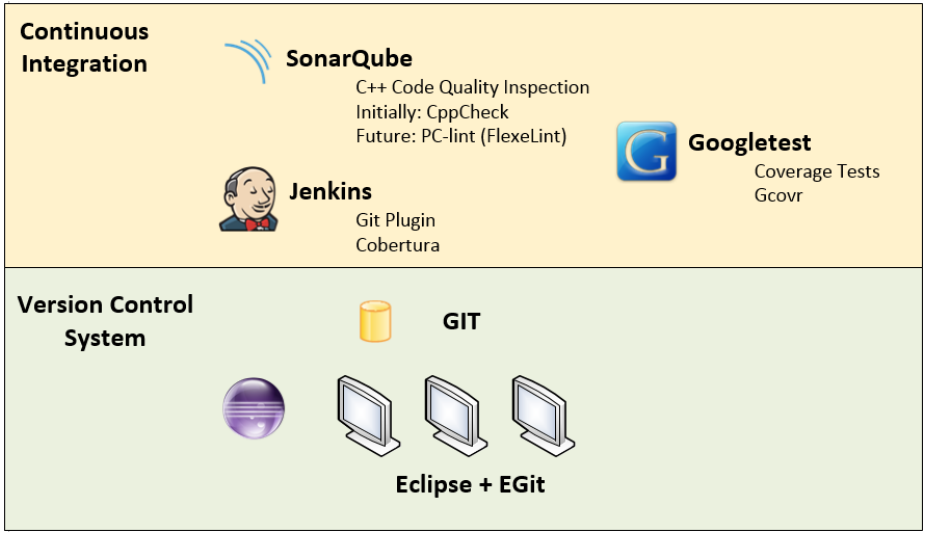
F4E internal resources	IO resources	Academic institutions	Industrial suppliers

Guarantee consistency of

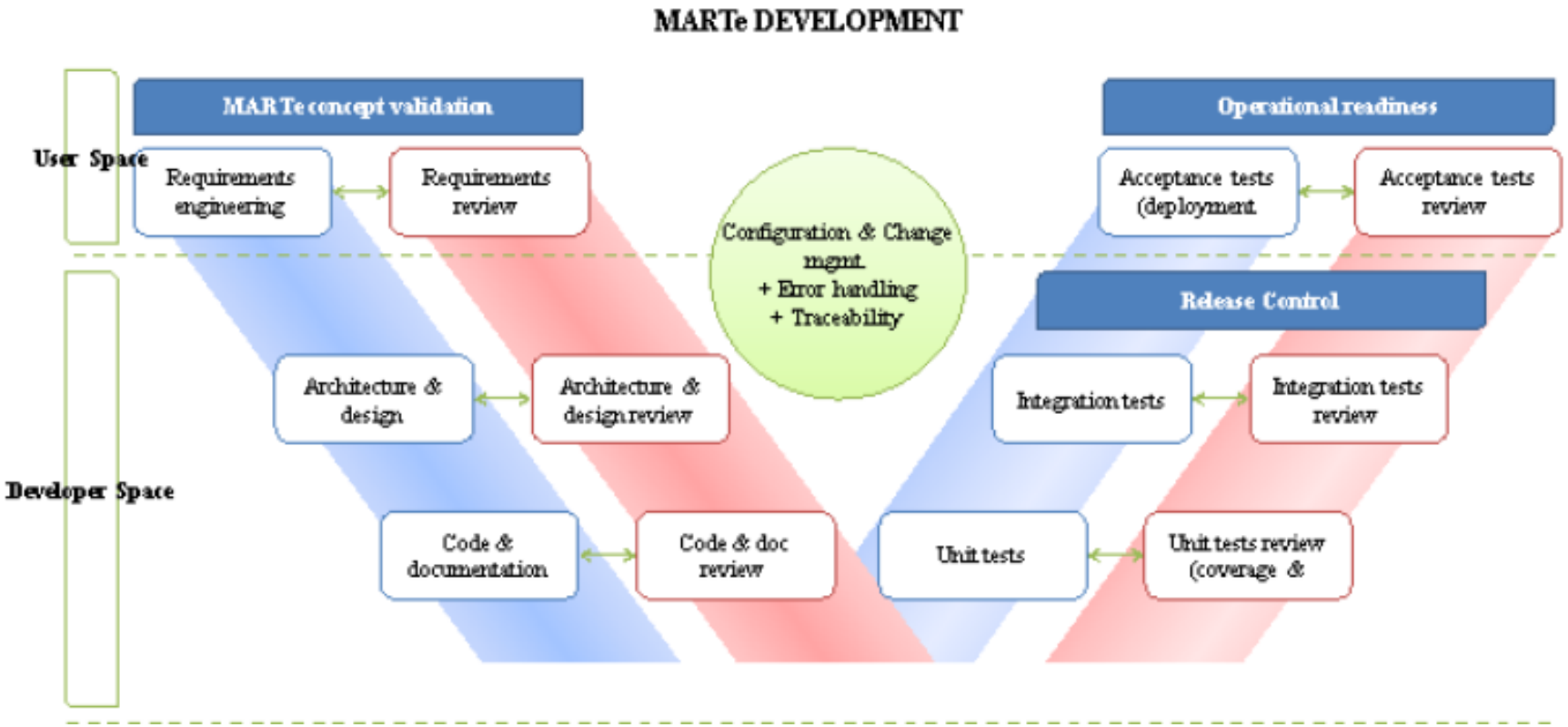
Implementation of coding and application of coding standards	Documentation	Testing standards

Safely integrate contributions from a large and heterogeneous development community

Manage changes to the configuration items and baselines

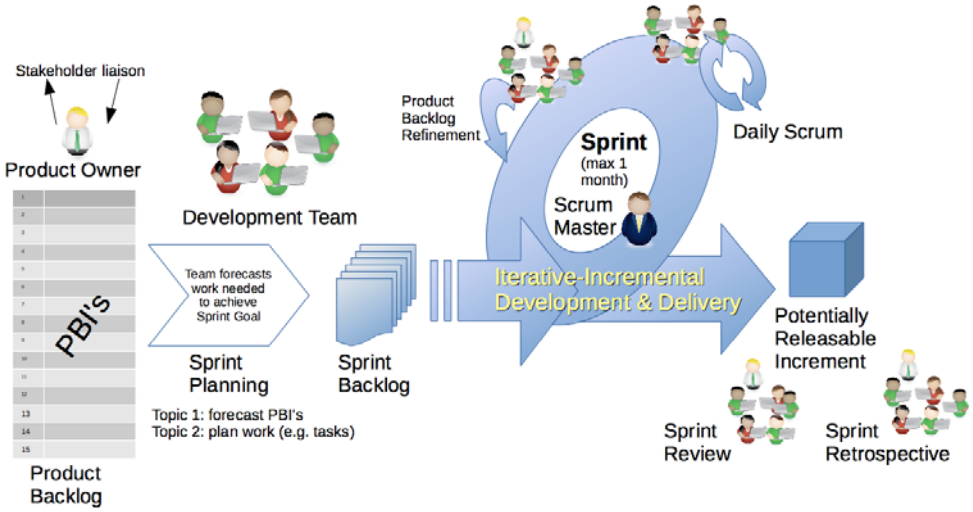


V & V


MARTe DEVELOPMENT
User Space
Developer Space
MARTe concept validation
Operational readiness
Requirements engineering
Requirements review
Configuration & Change mgmt.
+ Error handling
+ Traceability
Acceptance tests (deployment
Acceptance tests review
Release Control
Architecture & design
Architecture & design review
Integration tests
Integration tests review
Code & documentation
Code & doc review
Unit tests
Unit tests review (coverage &

Agile life-cycle method

- Requirements
  - Translated into user-stories
- User-stories
  - Architecture
  - Implementation
  - Unit-testing
  - Integrated testing
- Golden (and expensive!!!) rule
  - The Q&A review for each of the activities **shall** be done by a different person
    - …but this person can be a developer in another story
    - …or can be the coder for the implementation and the reviewer for the testing
- Every scrum…
  - …ends with a new release and with a QA audit
  - and with a sprint review for lessons learned regarding the QA process (very important!)



Issue tracking

- Redmine to implement Agile development
  - Full tracking which facilitates Auditing
  - Every story has the same structure:

Issue tracking

Architecture & design review

Date of the review: 29/07/2015

Person who did the review: Riccardo Vitelli

Version of architecture & design document: N/A. As per the original MARTe implementation, this is a support function and does not require formal design in UML.

Result of review: N/A

List of non-conformities: N/A

Code and documentation review

Date of the review: 29/07/2015

Person who did the review: Riccardo Vitelli

Result of review: PASS (MISRA non compliant)

List of non-conformities:

- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:92:85: Note 9025: More than two pointer indirection levels used for type 'ThreadInformation ***' [MISRA C++ Rule 5-0-19]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:92:85: Note 929: cast from pointer to pointer [MISRA C++ Rule 5-2-7]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:195:78: Note 925: cast from pointer to pointer [MISRA C++ Rule 5-2-8], [MISRA C++ Rule 5-2-9]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:207:107: Note 9025: More than two pointer indirection levels used for type 'ThreadInformation ***' [MISRA C++ Rule 5-0-19]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:207:107: Note 929: cast from pointer to pointer [MISRA C++ Rule 5-2-7]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:207:118: Note 925: cast from pointer to pointer [MISRA C++ Rule 5-2-8], [MISRA C++ Rule 5-2-9]

NOTE: ThreadsDatabase should be converted to the Singleton design pattern.

Issue tracking

Unit test review

Date of the review: 29/07/2015

Person who did the review: Ivan Herrero

Result of coverage tests review: PASS

Result of functional tests review: N/A

Result of review: FAIL

List of non conformities:

ThreadsDatabase class has not specific unit tests, but it is actually tested through the unit tests of Threads class.

Some error paths have not been exercised on unit tests.

Subtasks Add

Related issues Add

History

Updated by Riccardo Vitelli 3 months ago #1

- **Target version** set to *Backlog*

Updated by Riccardo Vitelli 3 months ago #2

- **Target version** changed from *Backlog* to *0.2*

Updated by Riccardo Vitelli 3 months ago #3

- **Status** changed from *New* to *Code: Impl*

Updated by Riccardo Vitelli 2 months ago #4

- **Status** changed from *Code: Impl* to *Code: Rev*

Updated by Ivan Herrero 2 months ago #5

- **Assignee** set to *Riccardo Vitelli*

Updated by Riccardo Vitelli 2 months ago #6

- **Description** updated (diff)

Associated revisions

Revision 7fce86c3
Added by André Neto 2 months ago

Lint of Threads #160, #161 and #162 is ready for review.

Revision bd448fd6
Added by André Neto 2 months ago

Simplified priority setting interface (#160, #161 and #162).

Revision e97f6c93
Added by Riccardo Vitelli 2 months ago

#162 - Minor formatting corrections (following template rules), removed unused includes.

Agile board

Agile board

Filters

Status: open

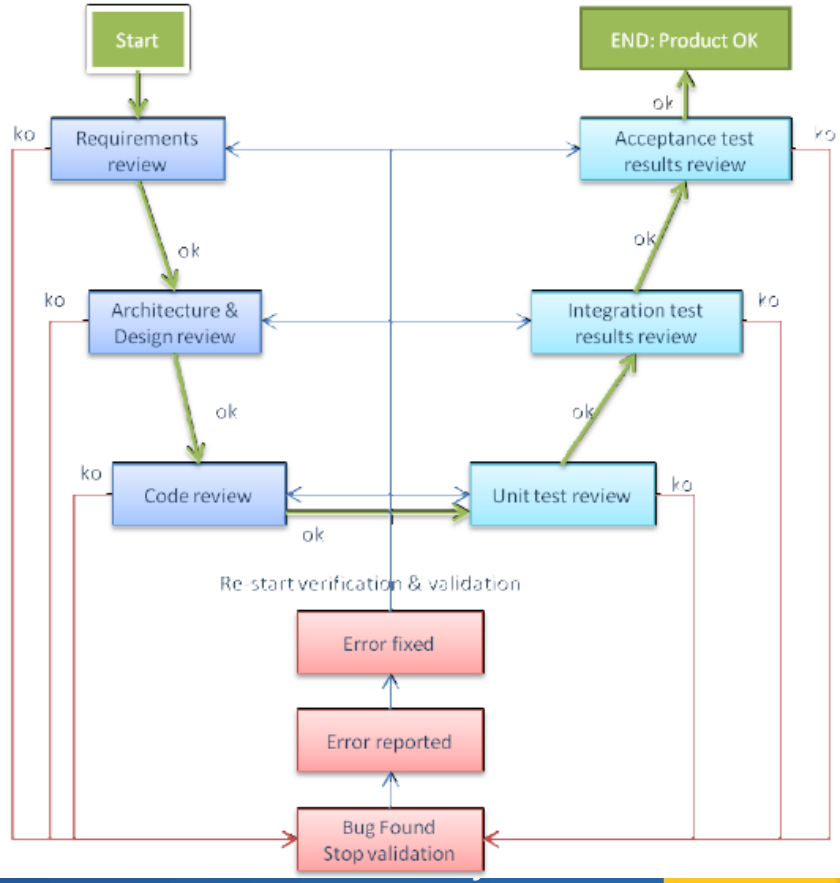
Target version: is MARTe2 - 0.6

Add filter

Options

Apply Clear

New (1)	Arch: Impl (0)	Arch: Rev (0)	Code: Impl (7)	Code: Rev (1)	Unit: Impl (0)	Unit: Rev (0)	Int: Impl (0)	Int: Rev (0)
Meeting #234 MARTe 2 Sprint report (v0.6)			**User story #188 Object introspection base infrastructure** Filippo Sartori	**User story #169 Merge into the develop branch the contents of the "windows" branch**				
			User story #235 ClassRegistryDatabase unique identifier type					
			User story #236 Specify C++ version in Makefile					
			User story #237 IOBuffer Giuseppe Ferro					



Q&A Audit

- Every scrum ends with a new release and with a QA audit

MARTe2 QA-audit (v0.3)

« Previous | 18 of 66 | Next »

Added by André Neto about 1 month ago. Updated about 1 month ago.

Status:	New	**Start date:**	23/08/2015
Priority:	Normal	**Due date:**	
Assignee:	-	**% Done:**	100%
Category:	-	**Spent time:**	-
Target version:	0.3		

Description Quote

Requirements review

Date of the review: 30/07/2015

Person who did the review: André Neto

Version of requirements: 0.3

Result of review: N/A

List of non-conformities: N/A

Comments: The requirements for this sprint are tracked in the EA model and are fully based on features that were already available in the previous version of the framework.

Architecture & design review

Date of the review: 30/07/2015

Person who did the review: André Neto

Version of architecture & design document: 0.3

Result of review: N/A

List of non-conformities: N/A

Comments: As per the original MARTe implementation, the user stories of this sprint are support functions and do not require formal design.

Note: No design document was produced but this was thoroughly discussed in several meetings and reported in #191. Discuss if this should be captured in EA, as a source of documentation (see #192).

Q&A Audit

Code and documentation review

Date of the review: 23/08/2015

Person who did the review: Ivan Herrero

Version of source code: 0.3

Result of review: PASS

List of non-conformities: N/A

- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:92:85: Note 9025: More than two pointer indirection levels used for type 'ThreadInformation ***' [MISRA C++ Rule 5-0-19]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:92:85: Note 929: cast from pointer to pointer [MISRA C++ Rule 5-2-7]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:195:78: Note 925: cast from pointer to pointer [MISRA C++ Rule 5-2-8], [MISRA C++ Rule 5-2-9]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:207:107: Note 9025: More than two pointer indirection levels used for type 'ThreadInformation ***' [MISRA C++ Rule 5-0-19]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:207:107: Note 929: cast from pointer to pointer [MISRA C++ Rule 5-2-7]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:207:118: Note 925: cast from pointer to pointer [MISRA C++ Rule 5-2-8], [MISRA C++ Rule 5-2-9]
- MARTe2-dev/Source/Core/L0Portability/ThreadInformation.h:47:25: Note 9109: type 'ThreadInformation' previously declared at location 'line 47' [MISRA C++ Rule 3-2-3]
- MARTe2-dev/Source/Core/L0Portability/ThreadInformation.cpp:93:49: Note 929: cast from pointer to pointer [MISRA C++ Rule 5-2-7]
- MARTe2-dev/Source/Core/L0Portability/ThreadInformation.h:47:25: Note 9109: type 'ThreadInformation' previously declared at location 'line 47' [MISRA C++ Rule 3-2-3]
- MARTe2-dev/Source/Core/L0Portability/ThreadInformation.h:47:25: Note 9109: type 'ThreadInformation' previously declared at location 'line 47' [MISRA C++ Rule 3-2-3]
- Warning 459: Function 'SystemThreadFunction(ThreadInformation *)' whose address was taken has an unprotected access to variable 'ThreadsDatabase::maxNOfEntries'
- Warning 459: Function 'SystemThreadFunction(ThreadInformation *)' whose address was taken has an unprotected access to variable 'ThreadsDatabase::nOfEntries'
- Warning 459: Function 'SystemThreadFunction(ThreadInformation *)' whose address was taken has an unprotected access to variable 'ThreadsDatabase::entries'
- Warning 459: Function 'SystemThreadFunction(ThreadInformation *)' whose address was taken has an unprotected access to variable 'ThreadsDatabase::internalMutex'
- Threads::name declares that it returns a C style string (char8*) without specifying who is the responsible for the management of the memory used by this char array (it actually returns a pointer to the char array hosted by an instance of ThreadInformation calling its ThreadName method). Perhaps it should be created a copy of the char array by means of Memory::StringDup or changed the prototype putting the char array as an out parameter of the method.
- Warning 1502: defined object 'ObjectHeap_' of type 'Heap' has no nonstatic data members
- Warning 1502: defined object 'ReferenceContainerHeap_' of type 'Heap' has no nonstatic data members
- Note 974: Worst case function for stack usage: 'ReferenceContainer::Find' is recursive, loop contains call to 'ReferenceContainer::Find'. See stack for a full report. [MISRA C+ Rule 7-5-4]
- The global FastPollingMutexSem in Object.cpp could be a class attribute.
- In the function DecrementReferences it would be better to initialise ret to referencesNumber.
- The implementation of the Heap was the minimum required to be able to compile, but this will be reviewed in v0.4

Q&A Audit

Unit test review

Date of the review: 23/08/2015

Person who did the review: Andre' Neto

Version of unit tests: 0.3

Result of coverage tests review: PASS

Result of functional tests review: PASS

Result of review: PASS

List of non-conformities:

- ThreadsDatabase class has not specific unit tests, but it is actually tested through the unit tests of Threads class.
- ThreadInformation class has not specific unit tests, but it is actually tested through the unit tests of Threads class.
- Unit testing coverage note: Error paths not exercised on:
  - ThreadsOS.cpp (missing test of priorities and ProcessorType::GetDefaultCPUs())
  - StringPortable.cpp (missing test for Concatenation with a NULL string)

- Unit testing coverage note: errors which require forcing an operating system fault are not tested: SleepOS.cpp, ThreadsOS.cpp, MutexSemOS.cpp, LoadableLibraryOS.h, EventSemOS.cpp.
- Unit testing coverage note: The following can only be tested when DLL support, configuration and introspection are implemented: ClassRegistryDatabase.cpp, ReferenceT.h, Object.cpp.
- Unit testing coverage note: The following are not tested because they have a private interface to disallow usage by the end-user: ClassRegistryItem.cpp, Object.cpp, ReferenceT.h. These have been removed from lcov using the markers //LCOV_EXCL_START and //LCOV_EXCL_STOP
- Unit testing coverage note: The delete object destructors (mangled with *D0Ev, see http://stackoverflow.com/questions/6613870/gnu-gcc-g-why-does-it-generate-multiple-dtors) are not being tested. This results in a low coverage function for many classes: Iterator.h, ReferenceContainerFilterReferences.h, SearchFilter.h and SortFilter.h
- Unit testing coverage note: In Processor::Family() the sentences inside the block protected by "if (family == 0xf)" has not been exercised, because the processor's family of the processor used in tests has not reached 0xf, so it does not need to use the Extended Family ID.
- Threads unit tests fails at:
  [FAILED] ThreadsGTest.TestPriority
  [FAILED] ThreadsGTest.TestGetThreadInfoCopy
  The reason why these tests fail, is that in Linux a regular user is not allowed to change the priority. This can be solved by either running the tests as the root user (not advisable), or by editing the file /etc/security/limits.conf and adding the following lines (change aneto to your username):

```
@aneto soft rtprio 100
@aneto hard rtprio 10
```

  [FAILED] BasicConsoleGTest.TestOpenModePerformCharacterInput (only happens in the continuous integration server)
- Testing note: No proper test can be done for the LoadableLibrary::Close() function. It will have to be discussed in the future.
- The following methods of BasicConsole do not have an explicit test, because the answer is different according to the target operating system:

Q&A Audit

Result of functional tests review: PASS

Result of review: PASS

List of non-conformities:

LCOV report (files containing functions not executed at all):
Note: execute with

```
lcov -l  Build/MARTe2.coverage.info
```

and compare with the latest master release using tkdiff.

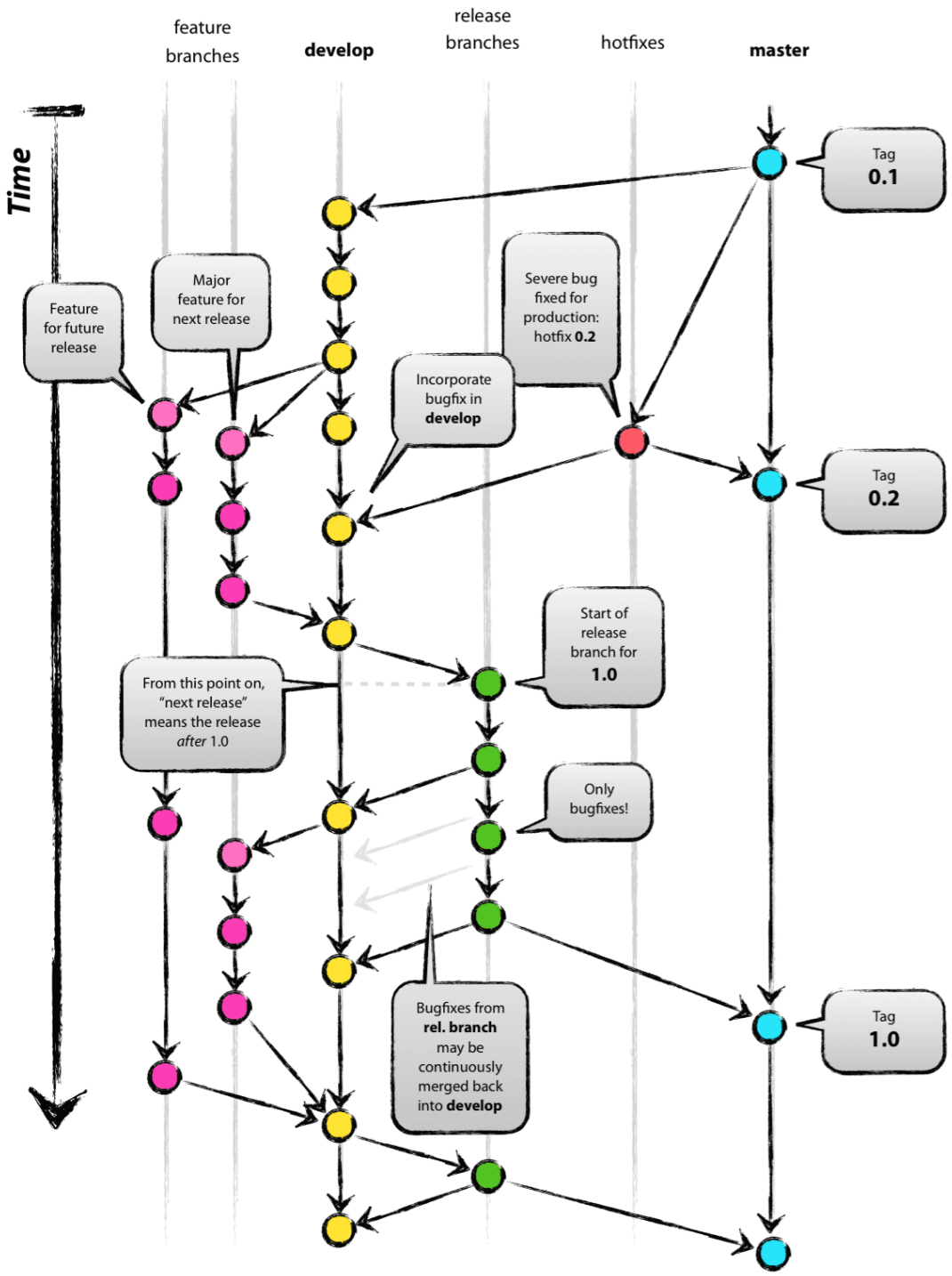
Files containing functions with coverage < 90% or which have changed w.r.t. to last master release. Exclamation marks inform of differences w.r.t. to the last master.:_

```
1                                                                    |Lines      |Functions|Branches       1                                                                    |Lines      |Functions|Branches
 2  Filename                                                         |Rate    Num|Rate  Num|Rate   Num      2  Filename                                                         |Rate    Num|Rate  Num|Rate   Num
 3  ================================================================================================      3  ================================================================================================
 7 ! [/home/aneto/Projects/MARTe2-dev/Source/Core/BareMetal/]                                                !
38  L1Portability/Environm...Linux/BasicConsole.cpp|79.4%   223|76.2%  42|    -    0     34  L1Portability/Environm...Linux/BasicConsole.cpp|79.4%   223|76.2%  42|    -    0
54  L1Portability/FastResourceContainer.cpp        |83.5%    97|71.4%   7|    -    0     50  L1Portability/FastResourceContainer.cpp        |83.5%    97|71.4%   7|    -    0
90  L2Objects/Introspection.cpp                    |73.3%    15|85.7%   7|    -    0     86  L2Objects/Introspection.cpp                    |73.3%    15|85.7%   7|    -    0
91 ! L2Objects/IntrospectionEntry.cpp              |87.3%   102|93.3%  15|    -    0     87 ! L2Objects/IntrospectionEntry.cpp              |88.3%   111|93.3%  15|    -    0
94  L2Objects/Object.cpp                           |88.8%   250|96.8%  31|    -    0     90  L2Objects/Object.cpp                           |88.8%   250|96.8%  31|    -    0
107 ! L2Objects/ReferenceT.h                       |94.2%    86|67.2%  1k|    -    0    103 ! L2Objects/ReferenceT.h                       |94.2%    86|67.8%  1k|    -    0
117  L3Streams/DoubleBufferedStream.cpp            |82.7%   110|84.0%  25|    -    0    113  L3Streams/DoubleBufferedStream.cpp            |82.7%   110|84.0%  25|    -    0
119 ! L3Streams/IOBuffer.cpp                       |95.1%   669|96.7%  30|    -    0    115 ! L3Streams/IOBuffer.cpp                       |95.0%   655|96.7%  30|    -    0
122  L3Streams/IOBufferIntegerPrint.cpp            |85.2%   460|93.5%  93|    -    0    118  L3Streams/IOBufferIntegerPrint.cpp            |85.2%   460|93.5%  93|    -    0
124  L3Streams/SingleBufferedStream.cpp            |84.1%   176|83.3%  24|    -    0    120  L3Streams/SingleBufferedStream.cpp            |84.1%   176|83.3%  24|    -    0
148 ! L4Configuration/TypeConversion.cpp           |94.8%   601|95.2%  21|    -    0    142 ! L4Configuration/TypeConversion.cpp           |95.8%   530|95.2%  21|    -    0
158  L4Messages/ObjectRegistryDatabaseMessageI.cpp |72.7%    11|88.9%   9|    -    0    152  L4Messages/ObjectRegistryDatabaseMessageI.cpp |72.7%    11|88.9%   9|    -    0
166 ! L5GAMs/GAMDataSource.cpp                     |93.9%   197|94.7%  19|    -    0    160 ! L5GAMs/GAMDataSource.cpp                     |94.0%   184|94.7%  19|    -    0

194 ! [/home/aneto/Projects/MARTe2-dev/Source/Core/FileSystem/L1Portability/Environment/Linux/]   188 ! [/home/aneto/Projects/MARTe2-dev-master/Source/Core/FileSystem/L1Portability/Environment/Linux/]
195  BasicFile.cpp                                  |87.8%   278|93.9%  33|    -    0    189  BasicFile.cpp                                  |87.8%   278|93.9%  33|    -    0
199  BasicUDPSocket.cpp                             |88.1%   118|95.7%  23|    -    0    193  BasicUDPSocket.cpp                             |88.1%   118|95.7%  23|    -    0
201  DirectoryScanner.cpp                           |88.0%    92|90.0%  10|    -    0    195  DirectoryScanner.cpp                           |88.0%    92|90.0%  10|    -    0

206 ! [/home/aneto/Projects/MARTe2-dev/Source/Core/FileSystem/L3Streams/]                    200 ! [/home/aneto/Projects/MARTe2-dev-master/Source/Core/FileSystem/L3Streams/]
207  File.cpp                                       |71.4%     7|57.1%   7|    -    0    201  File.cpp                                       |71.4%     7|57.1%   7|    -    0

214 ! [/home/aneto/Projects/MARTe2-dev/Source/Core/FileSystem/L6App/Environment/Linux/]   208 ! [/home/aneto/Projects/MARTe2-dev-master/Source/Core/FileSystem/L6App/Environment/Linux/]
215  Bootstrap.cpp                                  |41.9%    31|60.0%   5|    -    0    209  Bootstrap.cpp                                  |41.9%    31|60.0%   5|    -    0

225 ! [/home/aneto/Projects/MARTe2-dev/Source/Core/Scheduler/L1Portability/]              219 ! [/home/aneto/Projects/MARTe2-dev-master/Source/Core/Scheduler/L1Portability/]
226  ExceptionHandler.h                             | 0.0%     2| 0.0%   2|    -    0    220  ExceptionHandler.h                             | 0.0%     2| 0.0%   2|    -    0

228 ! [/home/aneto/Projects/MARTe2-dev/Source/Core/Scheduler/L3Services/]                  222 ! [/home/aneto/Projects/MARTe2-dev-master/Source/Core/Scheduler/L3Services/]

243 ! [/home/aneto/Projects/MARTe2-dev/Source/Core/Scheduler/L4LoggerService/]             237 ! [/home/aneto/Projects/MARTe2-dev-master/Source/Core/Scheduler/L4LoggerService/]

248 ! [/home/aneto/Projects/MARTe2-dev/Source/Core/Scheduler/L4Messages/]                  242 ! [/home/aneto/Projects/MARTe2-dev-master/Source/Core/Scheduler/L4Messages/]
249 ! QueuedMessageI.cpp                           |95.9%    49|90.0%  10|    -    0    243 ! QueuedMessageI.cpp                           |93.9%    49|90.0%  10|    -    0

253 ! [/home/aneto/Projects/MARTe2-dev/Source/Core/Scheduler/L4StateMachine/]              247 ! [/home/aneto/Projects/MARTe2-dev-master/Source/Core/Scheduler/L4StateMachine/]

257 ! [/home/aneto/Projects/MARTe2-dev/Source/Core/Scheduler/L5GAMs/]                     251 ! [/home/aneto/Projects/MARTe2-dev-master/Source/Core/Scheduler/L5GAMs/]

262  ================================================================================================      256  ================================================================================================
263 !                                        Total:|94.1% 21054|87.8%  7k|    -    0    257 !                                        Total:|94.5% 20656|87.9%  7k|    -    0
```

GTest report

Tools and methods

- Source control version
  - Git and GitLab



Coding standard

- C++ version
  - ISO/IEC 14882:2003 aka as C++03
- MISRA C++:2008
  - Guidelines for the C++ language targeted towards critical systems
    - Applies to the C++ language defined by the standard ISO/IEC 14882:2003
  - Emerged from the automotive industry, and is widely accepted as a model for best practices in sectors like aerospace, telecom, medical devices, defense, railway and others
  - Part of the C++ stdlib cannot be used (e.g. cstring)
- Strict documentation standard defined
- Strict coding style standard defined

Static checking tools (FlexeLint)

- Devil is in the details
  - Adjusting the configuration of the linting files requires time and expertise...
  - Without integrating it in the code development environment, it can make life a living hell to the developers.
- It is not cheap
  - The level of pedanticity is huge
    - This definition has 5 “mistakes”: unsigned int a = 3;

```
48:5: Note 970: Use of modifier or type 'unsigned' outside of a typedef [MISRA C++ Rule 3-9-2]
48:14: Note 970: Use of modifier or type 'int' outside of a typedef [MISRA C++ Rule 3-9-2]
48:23: Note 9117: implicit conversion changes signedness [MISRA C++ Rule 5-0-4]
49:1: Warning 438: Last value assigned to variable 'a' (defined at line 48) not used [MISRA C++ Rule 0-1-6], [MISRA C++
48:1: Info 830: Location cited in prior message
49:1: Warning 529: Symbol 'a' (line 48) not subsequently referenced [MISRA C++ Rule 0-1-3], [MISRA C++ Rule 0-1-4]
48:1: Info 830: Location cited in prior message
```

- One of the major advantages is that the deviations to the standard can be put in the code itself

```
    */
    uint32 listSize_;

    /*lint -e{1712} This class does not have a default constructor because
     * the element type size must be defined on construction and remain constant
     * during object's lifetime*/
};
```

Testing, testing and testing…

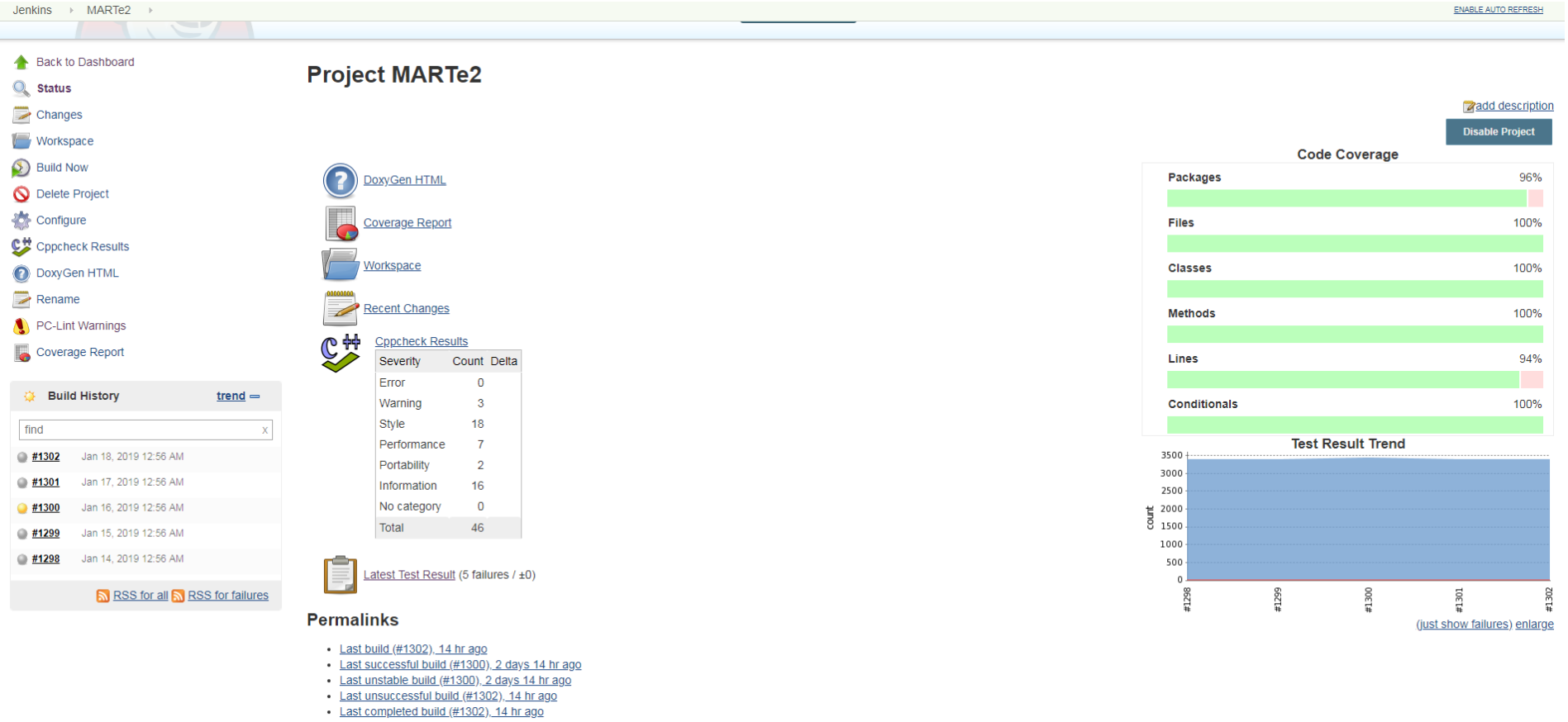
- Unit testing aiming at 100 % coverage
- Google test used as framework
  - But no explicit dependency on the framework
- Coverage with lcov frontend to gcov
  - Again, getting the configuration of this right is not trivial…

Development environment

- Eclipse is configured to guarantee
  - Code compilation
  - Correct formatting against coding style
  - Automatically generate the documentation with doxygen
  - Manage the git branches
  - Integrated with flexelint to highlight problems
  - Run the tests
  - Check for the coverage

Integration environment

- Jenkins runs everyday against the development branch



Where are we now?

- QA integration system successfully running for almost 4 years
- 19 development releases
- Very high MISRA compliance (almost 100 %)
- > 3400 core unit-tests
- > 1500 components unit-tests

PC-Lint Warnings

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
6	0	0

Summary

Total	High Priority	Normal Priority	Low Priority
6	1	2	3

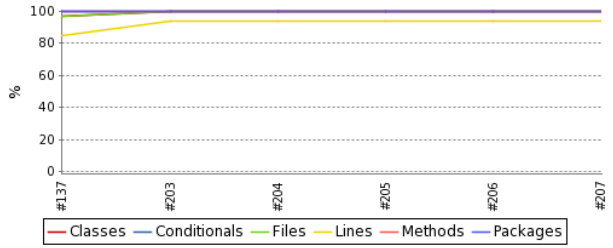
Details

Folders | Files | Categories | Warnings | Details | High | Normal | Low

Source Folder	Total	Distribution
MakeDefaults/Lint	1	
Source/Core/L0Portability	3	
Source/Core/L0Portability/OperatingSystem/Linux	2	
Total	6	

- High Coverage

Trend



Project Coverage summary

Name	Packages	Files	Classes	Methods	Lines	Conditionals
Cobertura Coverage Report	100% 5/5	100% 64/64	100% 64/64	100% 0/0	94% 3154/3370	100% 0/0

Lessons learned

- Finding the right balance between creativity and Q&A is an art
  - Modifications can be adapted but one must be extremely conservative
    - **This is also true for compiler versions, new versions of the standard language**
  - Every modification to the way of working requires significant energy
    - Infrastructure
    - Software
    - People's time
- Once a Q&A strategy and development environment is agreed stick with it
  - Be prepared to handle frustration
    - Linting and documentation are hard tasks which take time for people to get used to
  - Automate as much as possible

Lessons learned

- Linting forces people to think and guarantees some uniformity in addressing problems
  - But it is not a silver bullet!
  - It is not easy to configure
  - Some errors are extremely pedantic and developers need time to get used to it
  - Might introduce actually introduce bugs (when trying to solve linting issues)
    - Test!

Lessons learned

- Agile is a great tool
  - Makes sure that people talk everyday
  - People can share ideas and address immediate problems
    - Technical authority to take decisions is crucial
  - We have it mixed with some waterfall which can lead to dead-locks in the development (e.g. all the stories are waiting to be reviewed and only one Q&A reviewer is available)

Lessons learned

- Eclipse
  - Having all the Q&A process embedded in the tool is very beneficial
  - In particular having the IDE integrated with lint is very important
    - Otherwise you might be dealing with linting reports that have tens or hundreds of warnings and then having to look for the line numbers in the files

Lessons learned

- Unit testing
  - Try to aim at 100 % coverage and try to have as many tests as possible
  - We have decided not to apply the coding standard to tests
    - Simply could not allow the effort
  - Unit testing the classes that handle the operating system abstraction was not easy
    - Main fault conditions are hard to simulate

Lessons learned

- Q&A
  - Having each user-story water-falled allows to easily develop the sprint audit
  - Aligning sprint with releases is very helpful
    - Allows to have all the Q&A reports with the same release numbers and to have everything consistent
  - Allow developers to vent their frustration
    - Usually there is the opportunity to make small improvements in the processes …
    - **… but do not allow for clumsiness**
  - Documentation is a (very difficult) art
    - Make sure this is reviewed several times during the development
  - Make sure that the framework documentation (API and code) is aligned to the actual code

And the numbers are…

Item	Lines of code
Core	47 k
Core (test)	138 k
Official components	28 k
Official components (test)	122 k

- For every unit of development expect:
  - ~4.5x of QA
  - ~0.3x of QA review
- For every new release expect:
  - 1 day of QA

Collaboration models

- **Open discussion and feedback from audience**

FUSION FOR ENERGY

Thank you for your attention

Follow us on:

www.f4e.europa.eu

www.twitter.com/fusionforenergy

www.youtube.com/fusionforenergy

www.linkedin.com/company/fusion-for-energy

www.flickr.com/photos/fusionforenergy