Experiences with Software Quality Metrics in the EMI middleware

Maria ALANDES, CERN

CHEP 2012, New York

EMI is partially funded by the European Commission under Grant Agreement RI-261611

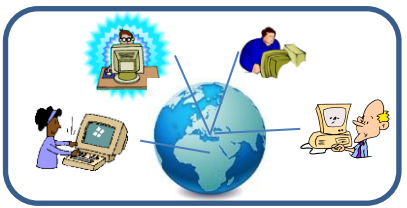
Outline

- EMI project context
- EMI quality Model
  - Why? How?
- EMI metrics
  - Products & Processes
- Tools
  - ETICS plugins & Dashboards
- Measurement Plan
  - Metrics reports
- Lessons learned

EMI INFSO-RI-261611

Our particular context

4 major Middleware Providers
- Developing middleware for the last decade
- Limited resources for QA



28 independent Development Teams
- Geographically distributed
- Well established processes and tools

56 interdependent software products
- Different technologies and programming languages
- > 2 million SLOC

Common QA policies

EMI INFSO-RI-261611

Why a Quality Model?

- **Software quality refers to non-functional requirements**
  - Reliability, Maintainability, Stability, …
- **A quality model helps to evaluate**
  - Software product quality
  - Software process quality

How ?

- **Defining quality goals and software characteristics**
- **Measuring whether the characteristics are actually present in the software.**

And this is what we did…

EMI INFSO-RI-261611

EMI Quality Model

What are the non functional requirements of the EMI middleware?

EMI Quality Requirements

UMD Quality Criteria

EMI general quality objectives

ISO/IEC 9126 software characteristics

Suitability
Maturity
Understandability
Stability
Testability
Adaptability
Instalability

If we want to release middleware to Distributed Computing Infrastructures, we need to meet their quality criteria.

EMI INFSO-RI-261611

Example

Software Characteristic: Testability

It is the capability of the software product to enable modified software to be validated

Importance for EMI: HIGH

Risks:

Failure to provide testable software may lead to not fulfill UMD requirement “Release changes testing: changes in a release of a product must be tested”.

Indicators:

The availability of test plans and test reports for released EMI software products are good indicators of the level of Testability.

Measures:

Test plan and test report availability, performed tests, regression tests for defect, functionality tests for new features

EMI INFSO-RI-261611

EMI Metrics

Now we know the quality requirements of the EMI middleware, but what do we want to measure?

Quality Model Definition

Software Characteristics fulfilling EGI UMD quality criteria

Project KPIs to be reported every quarter

Supporting project members

Release Manager

Quality Control team

Specific project needs

Sustainability plan

EMI metrics

- # technical objectives
  # user requirements
  ...
- # Incidents
  # Urgent Changes
  Incident Resolution Time
  ...
- # Immediate changes
  # High priority changes
  # successful builds
  ...
- # test plans
  # test reports
  # regression tests
  ...
- EPEL compliance
  ...

EMI INFSO-RI-261611

EMI Metrics

Software Process Metrics

technical objectives
user requirements
security vulnerabilities
Incidents
problems
Urgent Changes
Incident Resolution Time
Immediate changes
...

- Related to the way software changes are managed.
- Some metrics can be easily calculated from GGUS (incidents) or trackers in Savannah (user requirements and technical objectives).
- Software changes are tracked in 6 different tracking tools.

Bugzilla trac sourceforge LCG Savannah REDMINE

How can I extract information from so many different tracking tools?

Common QA policies

State	ARC State Description	dCache State Description	gLite State Description	UNICORE	StoRM State Description	MPI
Open	☑ Unconfirmed New Reopened	☑ EMI Open	☑ Open	☑ Open	☑ New	☑ New
Accepted	☑ Assigned	☑ Accepted	☑ Accepted	☑ Accepted	☑ In progress	☑ Accepted
Fixed	☑ Resolved/Fixed	☑ EMI Resolved	☑ Integration Candidate	☑ Fixed	☑ Resolved	☑ Fixed
Tested/Not Tested	☑ Verified/Fixed	☑ EMI Certified/Not Certified	☑ Fix Certified/Fix not Certified	⚠	☑ Tested/Not Tested	☑ Tested/Not Tested
Closed	☑ Closed (after released to production)	☑ EMI Closed	☑ Closed (after released to production)	⚠ Closed (after fix commited to VCS)	☑ Closed	☑ Closed (after fixed. Then ticket reopened and closed again with tested/not tested)
Resolution state	☑ Resolved/Fixed Resolved/Invalid Resolved/Wontfix Resolved/Duplicate Resolved/Worksforme	☑ Rejected	☑ Duplicate Won't fix Invalid Unreproducible Obsolete	☑ Fixed Invalid Rejected	☑ Rejected Closed (Positive State)	☑ Fixed Tested/Not Tested Invalid Won't Fix Duplicate

Common XML representation of the trackers information

https://twiki.cern.ch/twiki/bin/view/EMI/EMITrackerMappings

EMI INFSO-RI-261611

EMI Metrics

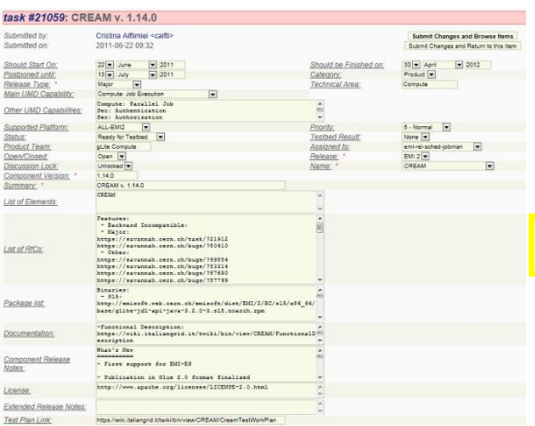
Software Product Metrics

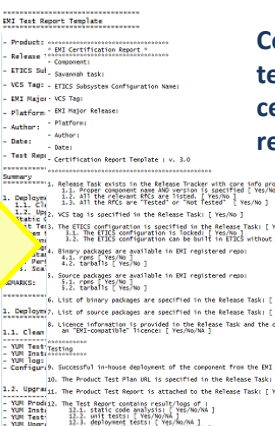
test plans
test reports
regression tests
functionality tests
EPEL compliance
Supported platforms
Source Packages
Reduced lines of code
...

- **Related to the released software product itself.**
- **It uses information stored in the release tracker.**

How can I extract information from 56 different software products?

Common QA policies





Common testing and certification report

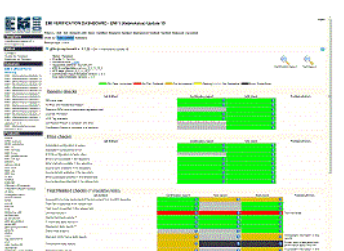
Tools

Now we know <u>what we want to measure</u> and we have <u>uniform access</u> to the information but …

Metrics reports need to be calculated regularly

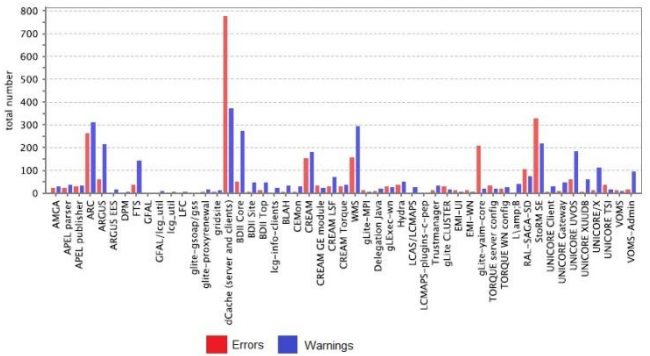
- **35 metrics defined in the Quality Model**
- **56 software products**
- **6 XML files containing software changes**
- **~100 software changes (medium, high, immediate)**
- **15 EMI 1 Updates, releasing 60 new product versions**

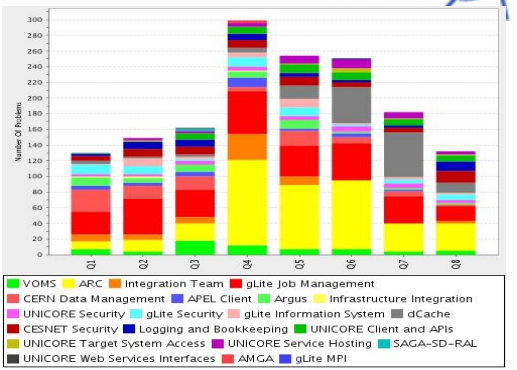
Automation is needed!



EMI INFSO-RI-261611

ETICS plugins and Charts





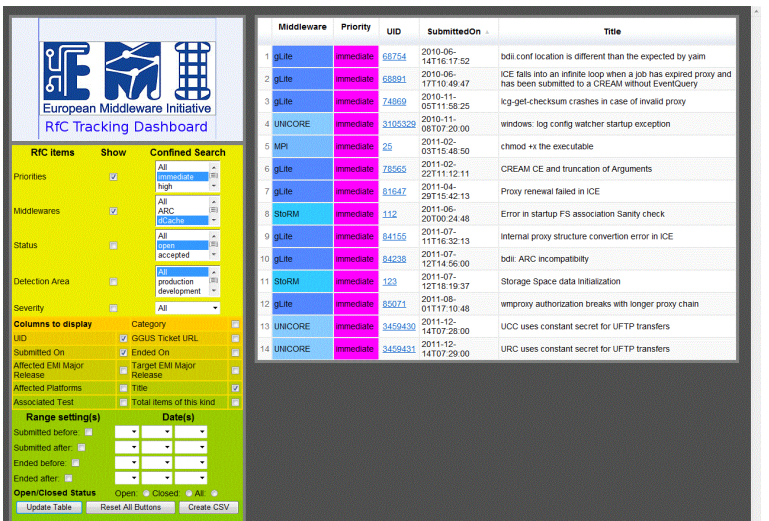
- **ETICS is the tool used to build and package EMI middleware.**
- **The ETICS plugin framework collects metrics during build and test execution like RPMlint.**
- **It enables the automation of product metrics generation.**

- **Input from all QA tools (ETICS logs and XML data) are processed to build trend diagrams using the chart generation framework.**

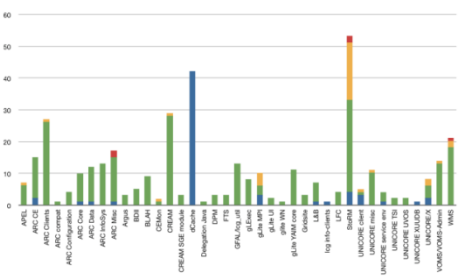
Benefits

- **Monitor whether project goals are being achieved.**
- **Availability of metrics charts for report generation.**

RfC Dashboard



- The RfC Dashboard offers a unique entry point to track software changes from 6 different tracking tools.
- It enables the automation of process metrics generation.



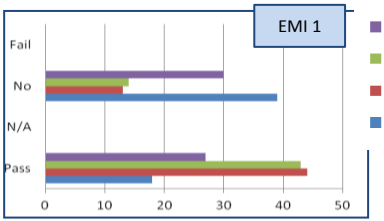
Benefits

- Support release manager by offering a single view of the software changes for all products.
- Report Generation.

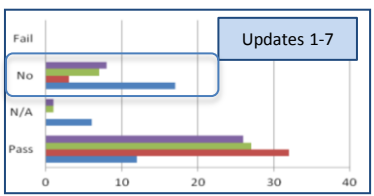
EMI INFSO-RI-261611

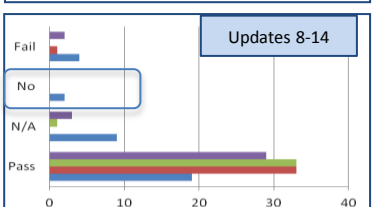
Verification Dashboard

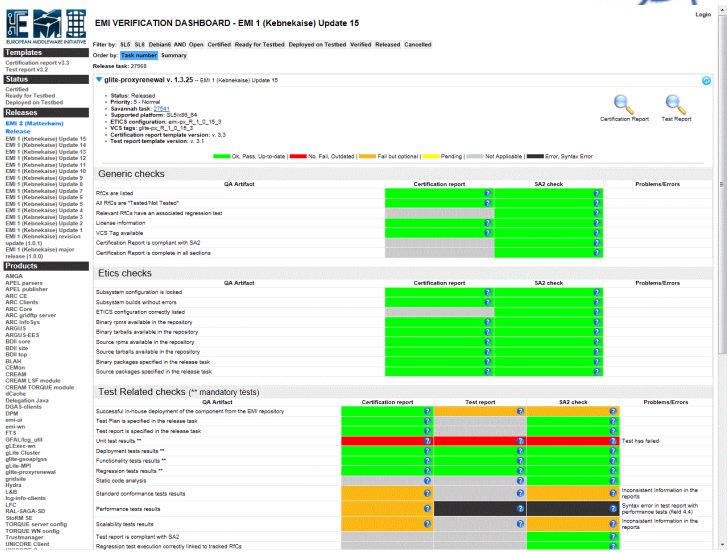
- **The verification dashboard is a tool automating quality control checks on software products included in a release.**



- Automated Regression
- Basic Functionality
- Clean Deployment
- Unit Tests







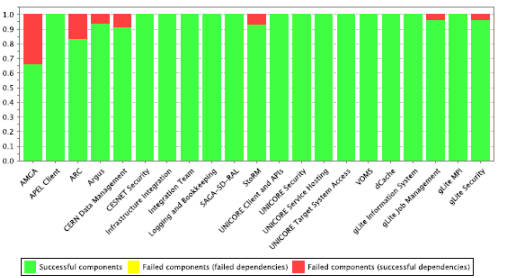
Benefits

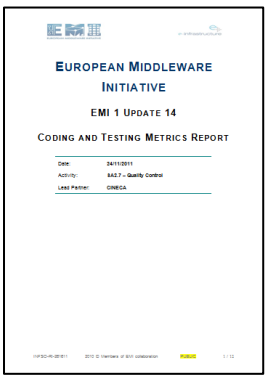
- **Support quality control activity.**
- **Support developers when preparing the release showing QA policy compliance.**

Measurement Plan

The measurement plan defines when metrics reports are generated:

- Planning phase of a major release
- Major release
- Release updates
- Weekly report for release manager
- KPIs for project quarterly reports





EUROPEAN MIDDLEWARE INITIATIVE

EMI 1 UPDATE 14

CODING AND TESTING METRICS REPORT

Date: 24/11/2011
Activity: SA2.7 – Quality Control
Lead Partner: CINECA

- Metrics templates are provided for each type of metrics report.
- Summary tables containing the thresholds of each metric.

Quality characteristic	Metrics	Required Level	Assessment Actual Result
Testability	Number of Test Plans per released software product.	One per software product.	
	Number of Test Reports per released EMI software product.	One per released software product.	
	Number of mandatory tests per EMI software product.	Ideally 100%, however an improvement per product is also a good indicator.	
	Number of RfCs tracking a defect with an associated regression test.	Ideally 100%, although an increased value per product per release is also a good indicator.	
	Number of RfCs tracking a new feature with an associated functionality test.	Ideally 100%, although an increased value per product per release is also a good indicator.	
	Number of development tasks tracking a new feature with an associated functionality test.	Ideally 100%, although an increased value per product per release is also a good indicator.	
	Number of passed Certification Checks	100% for the checks considered in the Production Release Criteria.	
Maintainability	KPI KJRA1.3 *Number of Reduced lines of code.*	> 33% (1/3) reduction over the three-year activity. The reduction can be consequence of	

EMI INFSO-RI-261611

Lessons learned

Complex software projects with heterogeneous development teams need

Common QA policies

Uniform processes

Monitoring and controlling software quality in large software projects requires

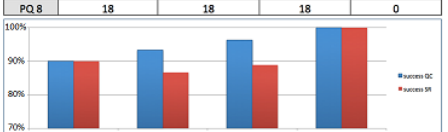
Automation

Dashboards

Software Provisioning

EMI software quality

	Finished #	Passed QC	Passed SR	Failed doc. QC
PQ 5	30	27	27	0
PQ 6	30	28	26	2
PQ 7	27	26	24	2
PQ 8	18	18	18	0



Benefits

- Enables evaluation of software process and product quality.
- Contributes to achieve project goals.

EMI INFSO-RI-261611