Experiences with Software Quality Metrics in the EMI middleware

Maria ALANDES, CERN

CHEP 2012, New York
Outline

EMI project context

EMI quality Model
  – Why? How?

EMI metrics
  – Products & Processes

Tools
  – ETICS plugins & Dashboards

Measurement Plan
  – Metrics reports

Lessons learned
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

Integrated Release
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 Quality Model

What are the non functional requirements of the EMI middleware?

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.
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
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
- ...

21st May 2012, CHEP 2012, New York
Software Process Metrics

- 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.

Common XML representation of the trackers information

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

https://twiki.cern.ch/twiki/bin/view/EMI/EMITrackerMappings
Software Product Metrics

- 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
Now we know **what we want to measure** and we have **uniform access** 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!**
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.

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

Input from all QA tools (ETICS logs and XML data) are processed to build trend diagrams using the chart generation framework.
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.
The verification dashboard is a tool automating quality control checks on software products included in a release.

**Benefits**

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

- Metrics templates are provided for each type of metrics report.
- Summary tables containing the thresholds of each metric.
Complex software projects with heterogeneous development teams need

Monitoring and controlling software quality in large software projects requires

Common QA policies

Uniform processes

Dashboards

Automation

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