



**eTICS2**  
The Grid Quality Process

## **JRA2**

# **Test Management Tools**

Eva Takacs (4D SOFT)

ETICS 2 Final Review  
Brussels - 11 May 2010

# Contents

Objectives

Major Achievements

Metrics and Statistics

Lessons learned and future work

Conclusions



# Objectives



## Main Objective of JRA2 activity

### Project objective: Enhancement of the services

- The main objective of the JRA2 activity is to design and implement a new approach for **distributed testing** by applying advanced workflow technology and distributed computing in a flexible, user friendly way



# Objective 1

- Develop the necessary **test interface standards** for proper collaboration with the job submission system, to provide metrics for A-QCM in well-defined formats and to provide users „readable” reports
- Year 1:
  - definition of standards
- Year 2:
  - provision of plugins (integrated test tools) having the proper interfaces
  - Integration of the Workflow Designer with Job Submission service having the proper xml format



## Objective 2

- Provide advanced test tool registration functionalities equipped with an advanced information system - as part of the ETICS Test System – allowing users an intuitive plugins management and information center
- Year 1:
  - Beta version having core functionalities
- Year 2:
  - Final version with information system and intuitive registration process focusing on the usability of the system



## Objective 3

- Provide an **advanced interface (Workflow Designer)** for the users to construct their complex multi-node deployments scenarios and testing process in a convenient, easy of use way even for non-professional system integrators
- Year 1:
  - Having a running instance on Sztaki's server, defining the interfaces with other ETICS services (Job Submission)
- Year 2:
  - Having integrated in ETICS framework, producing the well-defined output and having implemented the interpreter of the workflow



# Major Achievements





# ETICS Test System

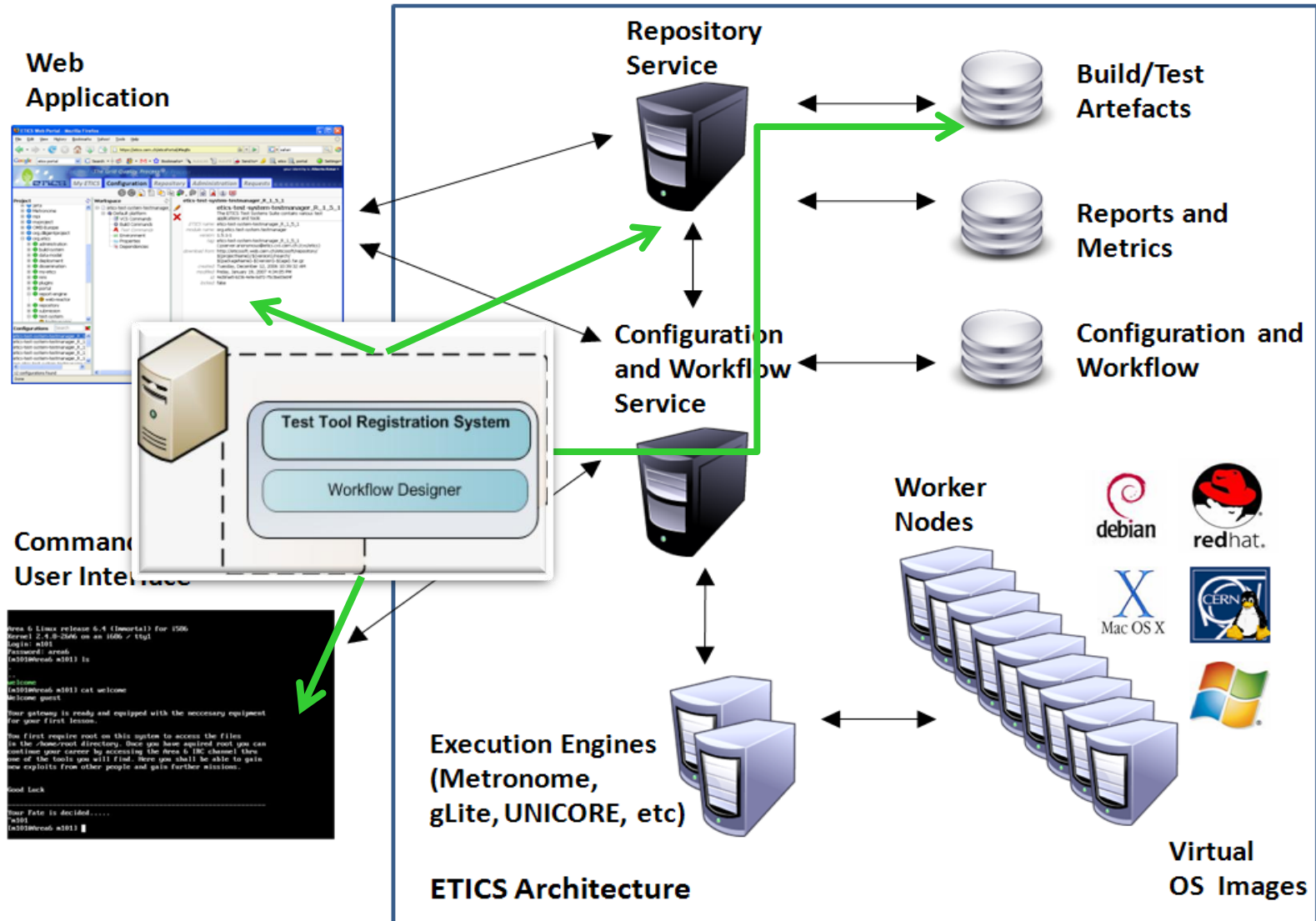
ETICS Test System is a portlet of the ETICS Web Portal comprised of the following elements:

- **Test Tool Registration System** – portlet and web service for plugins management and information center
- **Workflow Designer** - workflow tool and the associated services and workflow interpreter for complex, multinode test design
- **Collection of plugins** – added test tools, third party utilities producing appropriate reports and providing data in well-defined format for further analysis



# ETICS System architecture

## Test System in the main architecture



# Well-defined Interfaces

## Objective 1 – test interface standards

ETICS Test System provides

- complex test descriptions in a well-defined format – enabling the communication by job submission service in a standardised way
- representation of metrics/summary metrics in the ETICS build/test reports provided by test plugins are standardised – metrics to be available for trend analysis, A-QCM



# Test Tool Registration System

## Objective 2: advanced test tool management

It is a GWT based web portlet

- Providing a generic web based registration, authorisation procedure for plugins/test tools, /third party utilities management
  - Step-by-step approach
- It serves as an information center on the availability of test tools that can be applied during build/test processes for ETICS configurations
  - tag based search
  - test tool categories
- Provides an authorisation mechanism for plugins administration



# Test Tool Registration System

## Objective 2: advanced test tool management

- Addig a new plugin – step by step approach
- Information system – test tool categories
- Information system – tag based search

The screenshot displays the 'TestSystem' web interface. The top navigation bar includes 'Manage plugins' and 'Available Plugins'. A search filter is set to 'java; static; bugs; analysis; coding\_standard; standard; c/c++; cpp; dynamic; memory\_leak; java; statiunit; coverage; metric; complexity; coding\_stddependency; object-oriented; dynamic; CCocPlugin; metric; co; object-ori'. The 'JCKjmPlugin' version 1.0.1.1 is selected in the left sidebar.

The main content area shows the 'Version Details of "JCKjmPlugin"'. A green banner states: **Version is in production state! The Users can access it.**

**Version Description informations:**

*Displayable Plugin Name:* ckjm - Chidamber and Kemerer Java Metrics

*Description:* The program ckjm calculates Chidamber and Kemerer object-oriented metrics by processing the bytecode of compiled Java files. The program calculates for each class the following six metrics proposed by Chidamber and Kemerer.

- \* WMC: Weighted methods per class
- \* DIT: Depth of Inheritance Tree
- \* NOC: Number of Children
- \* CBO: Coupling between object classes
- \* RFC: Response for a Class
- \* LCOM: Lack of cohesion in methods

*Profile:* ckjm  
*Target:* test

*Properties:* ckjm.class.location  
 ckjm.jar.location  
 ckjm.lib.location

*Author:* 4dsoft  
*Vendor:*  
*HomePage:* <http://www.spinellis.gr/sw/ckjm/>  
*Support Email:* [info@4dsoft.hu](mailto:info@4dsoft.hu)  
*License:* open source  
*Logo:*  
*Tags:* java, object-orientec, metric

*Expected Input:* Any jar file or class files. The execution mode is automatically selected after checking the content of the given 'build path' director.

*Generated Output:* One XML and HTML report for every module containing the calculated metrics for every A± Java class.

*Version:* ver 1.0.1.1  
*Download Link:* <http://ltxetvm0016.cern.ch/plugins/JCKjmPlugin/JCKjmPlugin-1.0.1.tar.gz>

The bottom of the page shows the 'Install Commands' section.



# Metrics - Test Plugins

## Objective 1,2: registered tools having standard interfaces

Metrics	Type	Programming languages/ technologies	Tool	Etics Plugin
code checks	static	C++ Java	CppCheck Vera++ Checkstyle	CCppcheckPlugin CVerappPlugin JCheckstylePlugin
complexity	static	Java Python	Javancss	JCcnPlugin PyComplexityPlugin.py
design quality	static	Java	Jdepend	JDependPlugin
nr of „possible bugs”	static	C/C++ Python Perl PHP Java	Flawfinder, RATS PMD Findbugs	CFlawfinderPlugin CPyPhpRatsPlugin JPmdPlugin JFindbugsPlugin
nr of „possible bugs”	dynamic	C/C++	Valgrind	CValgrindPlugin
lines of code	static	All	SLOCCount	SLOCCountPlugin
coverage	dynamic	Java	Emma Cobertura	JUnitemmaPlugin JCoberturaPlugin
unit tests success rate	dynamic	Java Python	JUnit PyUnit	JUnitPlugin JUnitreportsPlugin.py PyUnitPlugin.py
Standards compliance	static	IPv6 web service	WSI	IPv6Plugin WSInteroperabilityPlugin
Object-oriented	static	Java C++	Ckjm Cccc	JCkjmPlugin CcccPlugin
metrics provided by profilers	dynamic	C/C++ Java	Jrat Valgrind	JRatPlugin CValgrindPlugin



# Workflow Designer

## Objective 3: advanced graphical user interface for test processes

- Web and Java based graphical application accessible through the ETICS Test System and using the ETICS configuration information
- It serves as a high-level front-end for the command-line based multinode test mechanism
- It allows multi-node test design, when the required services are deployed on different machines (the message passing between different phases of service deployment on different nodes is crucial)
- ETICS web service interprets the xml output of the design through a workflow interpreter



# Workflow Designer

## Objective 3: advanced graphical user interface for test processes

- Design of the graph representation
- Configuration of the graph
- Execution of the graph
- ETICS configuration elements generation

The screenshot displays the ETICS Workflow Designer interface. The top navigation bar includes 'Welcome', 'Submissions', 'Configuration', 'Repository', 'QA', 'TestSystem', 'Administration', and 'Requests'. The main workspace is divided into three panes:

- Project:** A tree view showing a project structure with nodes like 'multi-node', 'SeleniumTestNg02', 'client', 'HUB-RC-deploy', 'RC1-Node', 'RC2-Node', 'server', and 'TestSystem-Node'.
- Workspace:** A tree view showing the current workspace structure, including 'SeleniumTestNg02-2010-04-14\_09\_55', 'HUB-RC\_R\_1\_0\_0\_1', 'RC1\_R\_1\_0\_0\_1', 'RC2\_R\_1\_0\_0\_1', and 'TestSystem\_R\_1\_0\_0\_1'. A red box highlights the 'SeleniumTestNg02-2010-04-14\_09\_55' node.
- Test Commands:** A text area containing shell commands for setting up and running tests. A red box highlights the 'test' section of the commands.

The 'Test Commands' pane contains the following code:

```

description: Generated from Workflow: SeleniumTestNg02.

clean:
init:
test:
export HUB-CONSOLE-URL1='etics-get -b HUB-CONSOLE-URL1';
echo " STARTING RC1...";

mkdir selRC;
cd selRC;
etics-workspace-setup;
etics-get-project org.etics.testsuites;

etics-checkout --noask --verbose -c selenium-grid-rc01_R_1_0_0 selenium-test;

IP_ADDR= "ifconfig |grep 'inet addr:' |grep -v '127.0.0.1' |cut -d: f2 |awk '{print $1}";

etics-build -p selenium.host=${IP_ADDR} selenium.hub.url=${HUB-CONSOLE-URL1} --force --verbose -c
selenium-grid-rc01_R_1_0_0 selenium-test;

RC1-READY=True;

etics-set RC1-READY $RC1-READY;
export TERMINATE-RC1='etics-get -b TERMINATE-RC1';
echo " STARTING...";

directory as in the phase before;

cd selRC;
etics-workspace-setup;

etics-test -t clean --verbose -c selenium-grid-rc01_R_1_0_0 selenium-test;
  
```





# Testing Scenarios for ETICS Test System

Taking into account technical and marketing aspects showing ETICS capabilities several testing scenarios have been created

- Build process having unit tests and static analysers
- Run time Web service testing with coverage extension
- Deployment test of the portlet
- User acceptance and browser compatibility test scenario on a local or multi-node environment



# Desktopgrid Project

## EDGeS - ETICS collaboration

- The targets of the EDGeS (Enabling Desktop Grids for e-Science) project are scientific user communities that require large computing power and ready to use alternative e-Infrastructure solutions, e.g. volunteer or Desktop Grid systems.
- EDGeS needs a framework like ETICS because
  - volunteer computer owners accept only trustable software
  - requires understanding of many different environments and multiple platforms
- EDGeS started using ETICS framework for
  - Building and integration of the new applications to be added into the desktop grid environment
  - Validation and testing of the integrated applications focusing on **platform compatibility testing** and **dynamic tests** (Valgrind plugin)
  - AQCM for core Desktop Grid middleware components



# Deliverables and Milestones



# Deliverables

DJRA2.5	Test Management System - prototype	RELEASED
DJRA2.6	Test Management System - prototype	RELEASED



# Milestones

MJRA2.2	Test management system (prototype) delivered to SA1	OK
MJRA2.3	Complete test management system (final release) delivered to SA1	OK



# Lessons learned and future work



# Lessons Learned

- Having an environment that integrates builds and tests is more efficient than actually having separated tools
- Applying high level workflow technology for the orchestration of individual but related test tasks in a distributed environment proved to be indispensable

Lessons learned from the EDGeS – ETICS collaboration:

- Frameworks such as ETICS have a substantial learning curve, but using it in software projects reveals the existing technical problems easily in the early phase of the project contributing to a better final software product.



# 4D SOFT's contribution and benefits

INFSO-RI-223782





# Next steps for 4D SOFT

## Concrete steps

### Continuing technical developments after the project:

- Local ETICS instance installation (<https://etics3.4dsoft.hu:8443/eticsPortal>)
- Preparing a simplified ETICS installation package for single user purposes on local machine
- authorisation
- Windows compatibility

### 4D SOFT Customers:

- First customer (robotics industry) signed with 4D SOFT and will use ETICS for software engineering in C to develop avionics systems for unmanned aircraft.



# Conclusions

The major achievement of the **JRA2** activity is that it has created the possibility to have a full orchestration of large complex test cases across disconnected administrative domains. This is generally not possible to do even with high-end commercial tools and its the first open source software



# Thanks!



**eTICS2**  
The Grid Quality Process

<http://www.eticsproject.eu>

