

# The gLite Software Development Process

*Alberto Di Meglio*  
CERN



- **Software configuration management and tools**
- **Release process**
- **QA Metrics and Process Auditing**

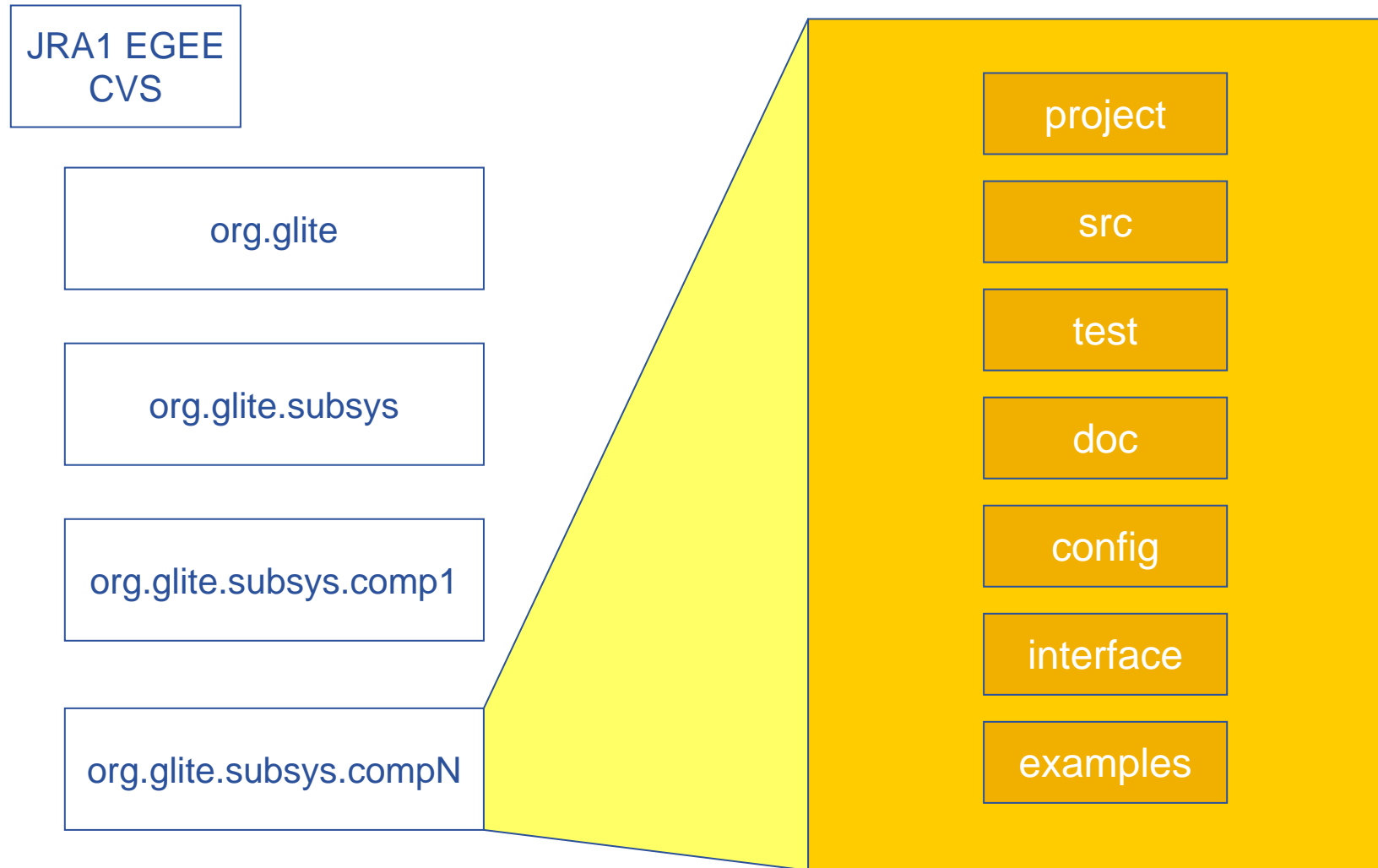
- JRA1 Software Process is based on an **iterative method** loosely based on **RUP** and some **XP** practices
- It comprises two main 12-month development cycles divided in shorter ***development-integration-test-release*** cycles lasting from 2 to 6 weeks
- The two main cycles starts with full Architecture and Design phases, but the architecture and design are **periodically reviewed and verified**.
- The process is **fully documented** in a number of standard document:
  - Software Configuration Management Plan (SCM)
  - Test Plan
  - Quality Assurance Plan
  - Developer's Guide

- The **SCM Plan** is the core document of the Software Process
- It contains a description of the processes and the procedures to be applied to the **six SCM activity areas**:
  - Software configuration and versioning, tagging and branching conventions
  - Build Tools Systems
  - Bug Tracking
  - Change Control and the Change Control Board (CCB)
  - Release Process
  - Process Auditing and QA Metrics
- It is based on a number of **standard methods and frameworks** including:
  - ISO 10007:2003 - Quality management systems -- Guidelines for configuration management, ISO, 2003
  - IEEE Software Engineering Guidelines (<http://standards.ieee.org/reading/ieee/std/se>)
  - The Rational Unified Process (<http://www-306.ibm.com/software/awdtools/rup/>)
- In addition it adopts **best-practice solutions**<sup>1</sup> to guarantee the highest possible quality in a very distributed and heterogeneous collaboration

<sup>1</sup>S.P. Berczuk, Software Configuration Management Patterns, Software Patterns Series, Addison-Wesley, 2002

A. Di Meglio et al., A Pattern-based Continuous Integration Framework for Distributed EGEE Grid Middleware Development, Proc. CHEP 2004

- Based on **CVS** using CERN Central IT CVS service
- **Fixed directory structure** for each module
- Rules for tagging and branching (e.g. bug fix branches)
- Common naming conventions for baseline and release tags
- **Configuration files to automate creation of workspaces**
  - Used to enforce build reproducibility
  - Used to create private workspaces for developers
- **Rules apply also to external dependencies, all third-party packages and versions are controlled by the build systems, not the developers**



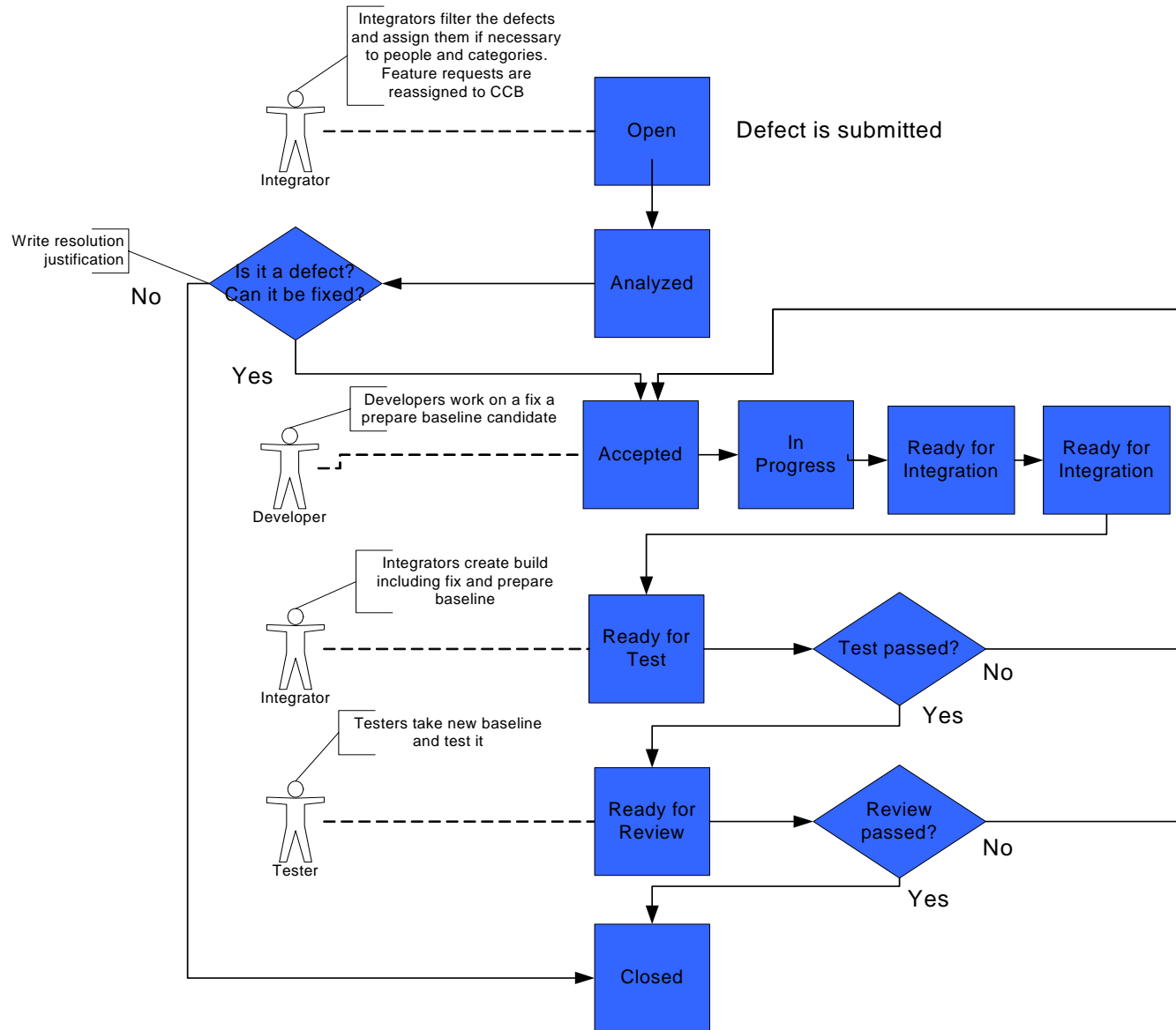
- **Ant**: used for the general build management and all Java modules.
- **Make + GNU Autotools**: for C/C++, Perl and other languages as necessary. Some effort has been done to port the Makefiles to Windows or use Cygwin, but with very limited success
- **CruiseControl**: used to automate the nightly and integration builds on the central servers
- An **abstraction layer** has been created on top of these tools to provide a common interface to all build tools independently of the language, platform and tool used

- **Two nightly build servers on RH Linux 3.0 (ia32)**
  - Clean builds out of HEAD and v. 1.x every night of all components
  - Results are published to the gLite web site
  - Tagged every night and totally reproducible
- **One continuous build server on RH Linux 3.0 (ia32)**
  - Incremental builds out of v. 1.x every 60 minutes
  - Results published to CruiseControl web site
  - Automated build error notifications to developers and Integration Team
- **One nightly build server on RH Linux 3.0 (ia64)**
  - Clean builds every night of all components
- **One nightly build server on Windows XP**
  - Clean builds every night of all components currently ported to Windows
- **Build system supported platforms:**
  - Red Hat Linux 3.0 and binary compatible platforms (SLC 3, CentOS, etc), 32 and 64-bit (gcc)
  - Windows XP/2003

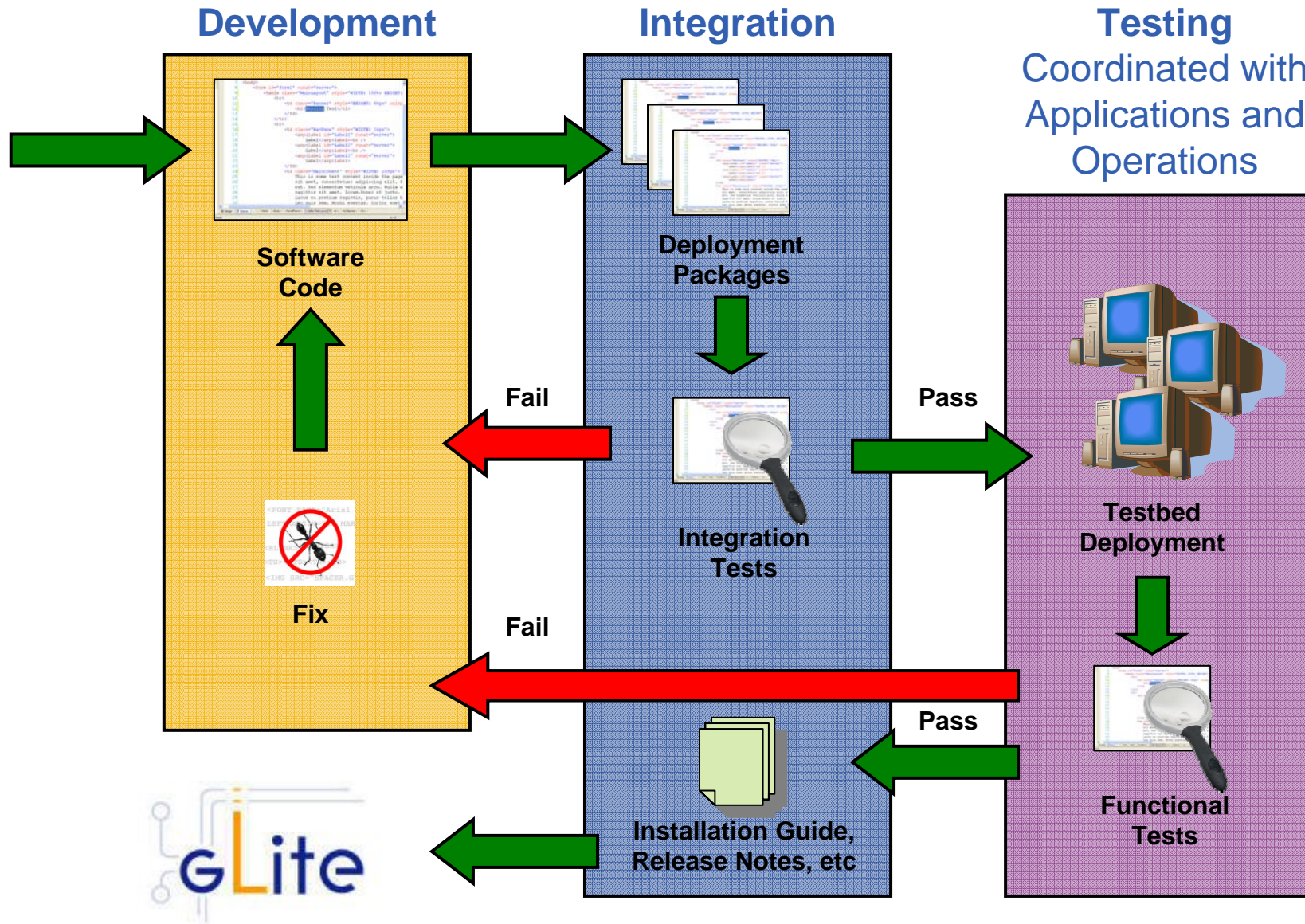


- Based on the **Savannah** project portal at CERN
- Used also for **change requests** (for example API changes, external libraries version changes, etc). In this case, request are assigned to the **Change Control Board** for further evaluation (explained later)
- Heavily customized to provide several **additional bug status** on top of the default Open and Closed (!)
- Each gLite subsystem is tracked as a separate category and related bugs are assigned to the responsible clusters

- Two **conditions**: Open, Closed
- Ten main **states**: None, Accepted, In Progress, Integration Candidate, Ready for Integration, Ready for Test, Ready for Review, Fixed, Duplicate, Invalid
- **Transitions** between two states is subject to specific **checks** (Is it a defect? Can it be fixed? Test passed, Review passed?)
- Not all transitions are allowed and additional automated changes can be triggered by changing the bug status (condition, assignment, etc)



- All public changes must go through a **formal approval process**
- The CCB is tasked to collect and examine the **change requests**
- Changes are tracked and handled as quickly as possible
- The CCB is not a physical team, but a **role** that is assumed by more than one team or group depending on the type of change (interface changes, bug fixes, software configuration changes, etc)



- **Software Metrics** are collected as part of the build process
- Failure to pass a **quality check** fails the build
- Additional checks are implemented in the version control system (coding style, documentation tags)
- Software Defect and QA Metrics are collected from the defect tracking system
- **Reports** and **graphs** are published on the project web site

gLite coding style report

system  
**org.glite**

---

Summary			
Subsystems	SLOC	Errors	Errors / line
13	1091608	52792	0.0484

---

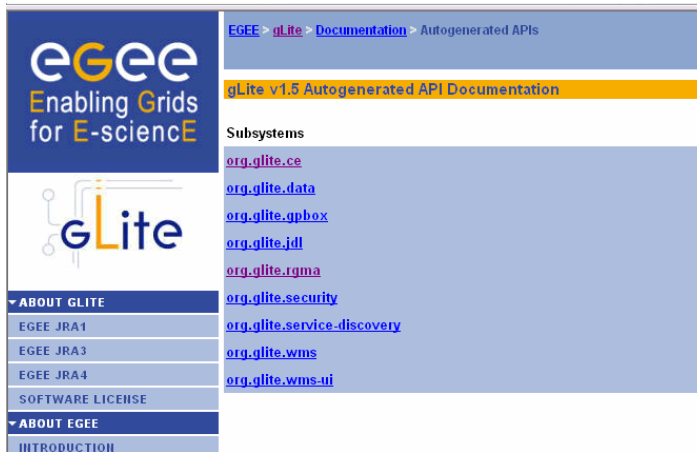
Subsystems			
Name	Errors	Lines	Errors / line
org.glite.jdl	2572	3847	0.6686
org.glite.ce	9575	26410	0.3626
org.glite.wms-ui	18785	93834	0.2002
org.glite.rgma	7677	110008	0.0698
org.glite.gpbox	6645	108168	0.0614
org.glite.service-discovery	336	7508	0.0448
org.glite.amga	1448	41802	0.0346
org.glite.security	3225	108056	0.0298
org.glite.wms	1888	297219	0.0064
org.glite.data	641	196888	0.0033
org.glite.dgas	0	31226	0.0000
org.glite.testsuites	0	63582	0.0000
org.glite.wms-utils	0	3060	0.0000

Coding conventions checked by [CHECKSTYLE](#) and [CODEWIZARD](#) using the gLite coding conventions.  
Line count by [SLOCCOut](#).

Internet

<span>Build Results</span> <span><b>Test Results</b></span> <span>XML Log File</span> <span>Control Panel</span>			
Name	Status	Time(s)	
.org.glite.rgma.ProducerPropertiesTest			
testIsHistory	Success	0.008	
testIsLatest	Success	0.000	
<a href="#">Properties »</a>			
.org.glite.rgma.QueryPropertiesTest			
testIsHistory	Success	0.008	
testIsLatest	Success	0.000	
testIsContinuous	Success	0.000	
testEquals	Success	0.000	
<a href="#">Properties »</a>			
.org.glite.rgma.StorageTest			
testEquals	Success	0.006	
testGetPassword	Success	0.000	
testGetLocation	Success	0.000	
testGetUserName	Success	0.000	
testIsDatabase	Success	0.000	
testIsMemory	Success	0.001	
testHasDetails	Success	0.000	
<a href="#">Properties »</a>			
.org.glite.rgma.TimeIntervalTest			
testValueAsMillis	Success	0.006	
testValueAsSeconds	Success	0.000	
testValueAsMinutes	Success	0.000	
testValueAsHours	Success	0.000	
testValueAsDays	Success	0.000	
<a href="#">Properties »</a>			





EGEE > [gLite](#) > [Documentation](#) > Autogenerated APIs

**gLite v1.5 Autogenerated API Documentation**

Subsystems

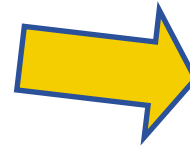
- [org.gLite.ce](#)
- [org.gLite.data](#)
- [org.gLite.gpbox](#)
- [org.gLite.jdl](#)
- [org.gLite.rgma](#)
- [org.gLite.security](#)
- [org.gLite.service-discovery](#)
- [org.gLite.wms](#)
- [org.gLite.wms-ui](#)

▼ ABOUT GLITE

- EGEE JRA1
- EGEE JRA3
- EGEE JRA4
- SOFTWARE LICENSE

▼ ABOUT EGEE

- INTRODUCTION




EGEE > [gLite](#) > [Documentation](#) > [Autogenerated APIs](#) > [org.gLite.rgma](#)

**gLite v1.5 Autogenerated API Documentation - org.gLite.rgma**

Components

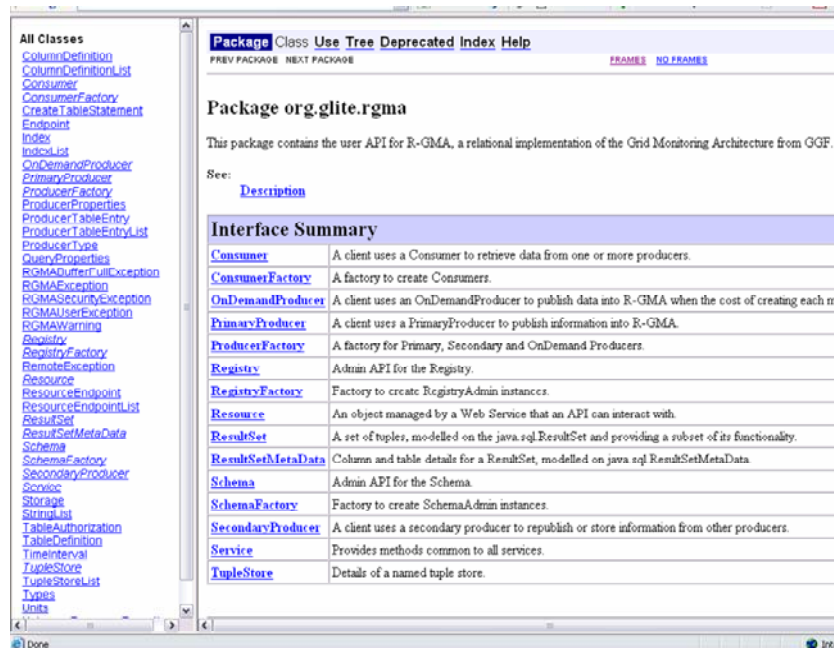
- [org.gLite.rgma.api.cpp](#)
- [org.gLite.rgma.api.java](#)
- [org.gLite.rgma.log4j.cpp](#)
- [org.gLite.rgma.log4j](#)
- [org.gLite.rgma.server.servlet](#)
- [org.gLite.rgma.services](#)
- [org.gLite.rgma.servicetool](#)
- [org.gLite.rgma.stubs.servlet.java](#)

▼ ABOUT GLITE

- EGEE JRA1
- EGEE JRA3
- EGEE JRA4
- SOFTWARE LICENSE

▼ ABOUT EGEE

- INTRODUCTION



Package Class Use Tree Deprecated Index Help

PREV PACKAGE NEXT PACKAGE [FRAME](#) [NO FRAME](#)

**Package org.gLite.rgma**

This package contains the user API for R-GMA, a relational implementation of the Grid Monitoring Architecture from GGF.

See: [Description](#)

**Interface Summary**

<a href="#">Consumer</a>	A client uses a Consumer to retrieve data from one or more producers.
<a href="#">ConsumerFactory</a>	A factory to create Consumers.
<a href="#">OnDemandProducer</a>	A client uses an OnDemandProducer to publish data into R-GMA when the cost of creating each me
<a href="#">PrimaryProducer</a>	A client uses a PrimaryProducer to publish information into R-GMA.
<a href="#">ProducerFactory</a>	A factory for Primary, Secondary and OnDemand Producers.
<a href="#">Registry</a>	Admin API for the Registry.
<a href="#">RegistryFactory</a>	Factory to create RegistryAdmin instances.
<a href="#">Resource</a>	An object managed by a Web Service that an API can interact with.
<a href="#">ResultSet</a>	A set of tuples, mo-delled on the java.sql.ResultSet and providing a subset of its functionality.
<a href="#">ResultSetMetaData</a>	Column and table details for a ResultSet, modelled on java.sql.ResultSetMetaData
<a href="#">Schema</a>	Admin API for the Schema.
<a href="#">SchemaFactory</a>	Factory to create SchemaAdmin instances.
<a href="#">SecondaryProducer</a>	A client uses a secondary producer to republish or store information from other producers.
<a href="#">Service</a>	Provides methods common to all services.
<a href="#">TupleStore</a>	Details of a named tuple store.



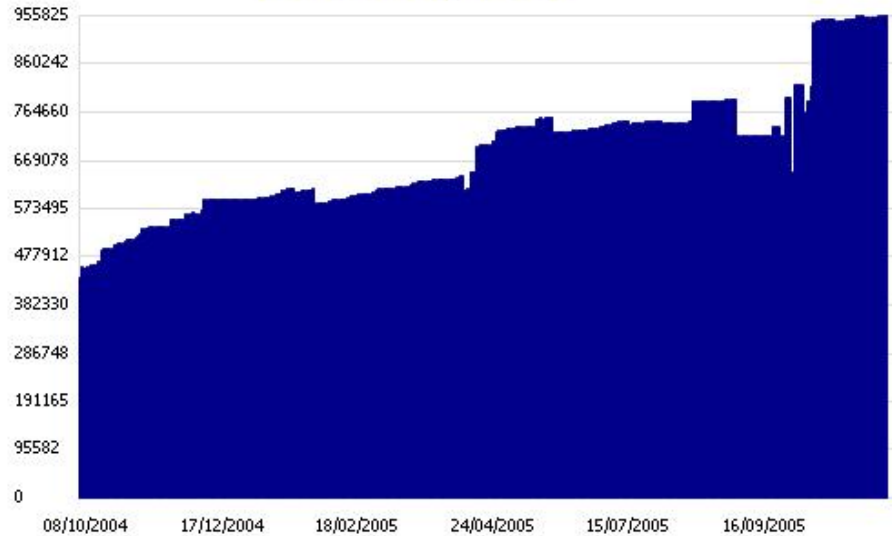
## Total Physical Source Lines of Code (SLOC)

- SLOC = 955,825 (as of 21 November 2005)

## Total SLOC by language (dominant language first)

- **Java**            **285271 (29.85%)**
- **C++**            **266828 (27.92%)**
- **Ansi C**        **209326 (21.90%)**
- **Perl**            **75386 (7.89%)**
- **sh**              **70904 (7.42%)**
- **Python**        **43459 (4.55%)**
- **Total complete builds: 641 (all 1.x branches), 236 (HEAD)**
- **Number of subsystems: 21**
- **Number of CVS modules: 454**
- **Pre-Release Defects/KSLOC = 2.75**
- **Post-Release Defects/KSLOC = 1.10**

**Code Size (SLOC)**



Copyright (c) 2004 EGEE

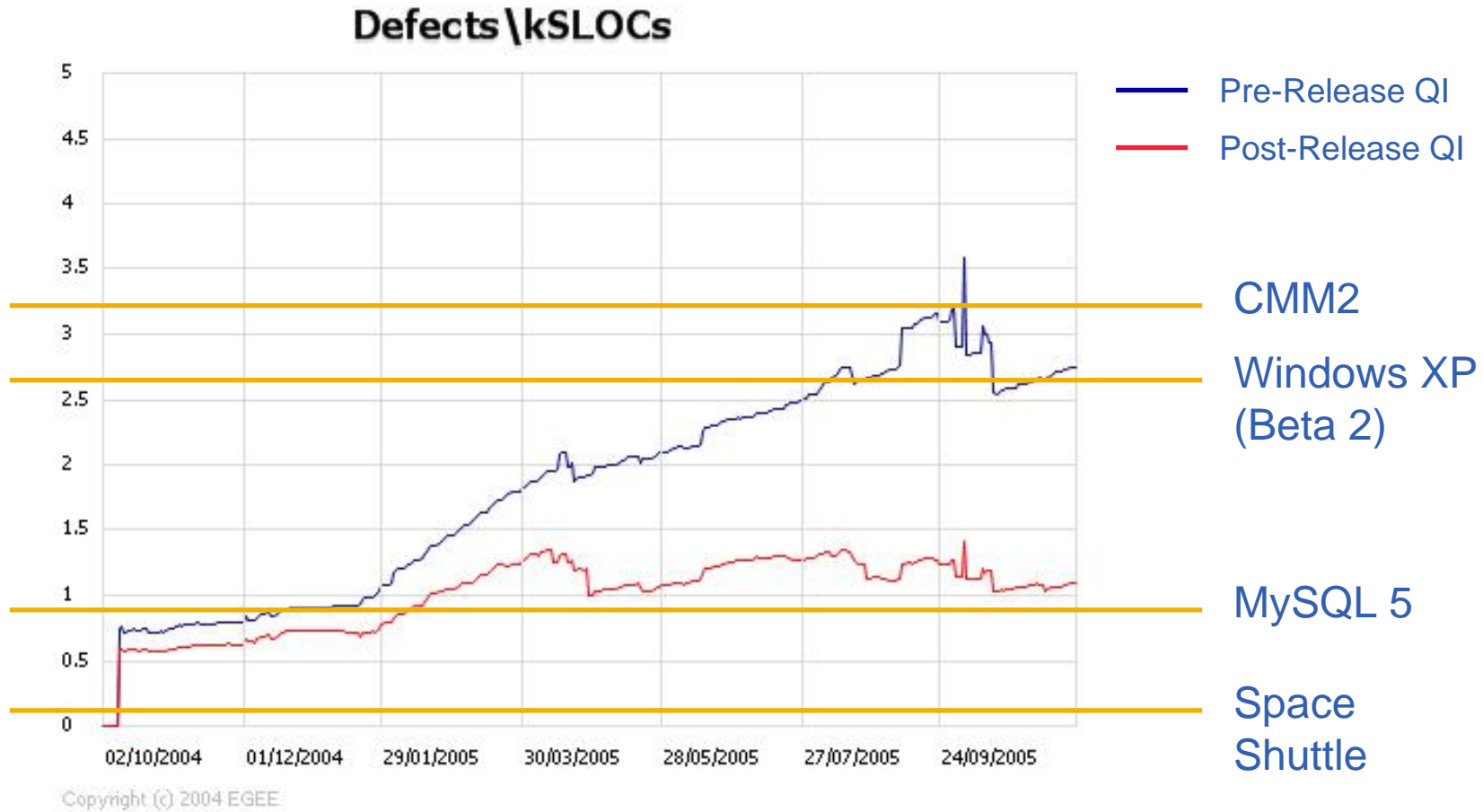
The Code Size chart shows the changes in total number of SLOCs during the life of the project

The Code Stability chart shows the change rate of code size during the life of the project. As the project nears completion the rate should approach 0

**Code Stability (dSLOC/dt)**



Copyright (c) 2004 EGEE



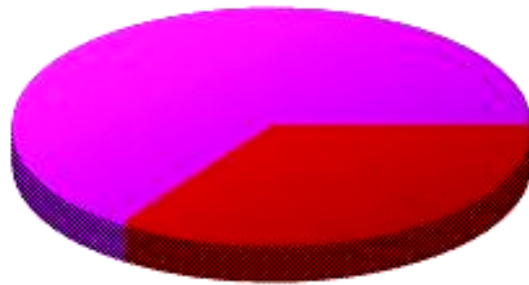
CMM2

Windows XP (Beta 2)

MySQL 5

Space Shuttle

## Open and Closed Bugs



- Open (1076 - 40.83%)
- Closed (1559 - 59.17%)

Copyright (c) 2004 EGEE

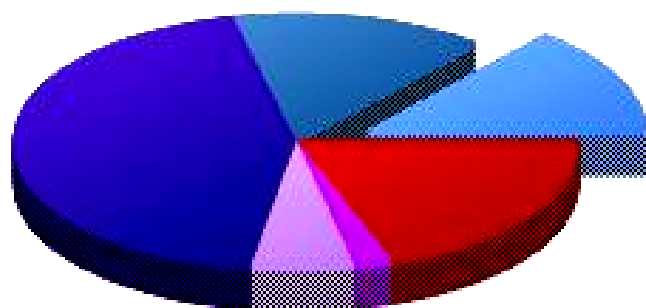
## Bugs Status



- Fixed (1095 - 41.56%)
- Ready for Test (358 - 13.59%)
- Invalid (211 - 8.01%)
- None (193 - 7.32%)
- Duplicate (159 - 6.03%)
- Accepted (142 - 5.39%)
- Ready for Integration (139 - 5.28%)
- Remind (123 - 4.67%)
- Wont Fix (67 - 2.54%)
- In progress (60 - 2.28%)
- Integration Candidate (50 - 1.9%)
- Unreproducible (28 - 1.06%)
- Ready for Review (10 - 0.38%)

Copyright (c) 2004 EGEE

## Bugs Severity



- 1 - Enhancement (375 - 14.23%)
- 2 - Cosmetic (100 - 3.8%)
- 3 - Minor (331 - 12.56%)
- 4 - Normal (973 - 36.93%)
- 5 - Major (648 - 24.59%)
- 6 - Critical (208 - 7.89%)

Copyright (c) 2004 EGEE

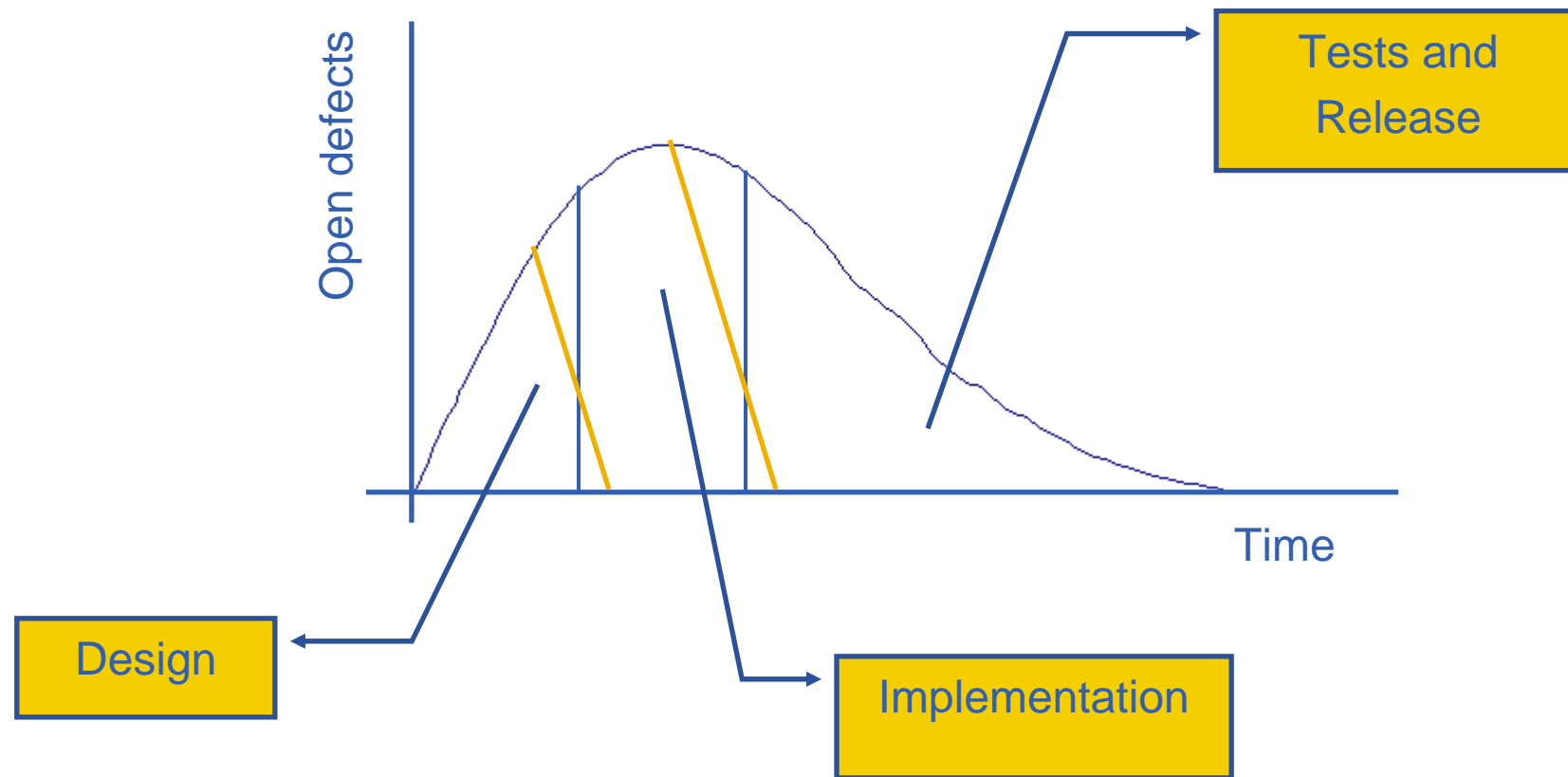
## Bugs Type

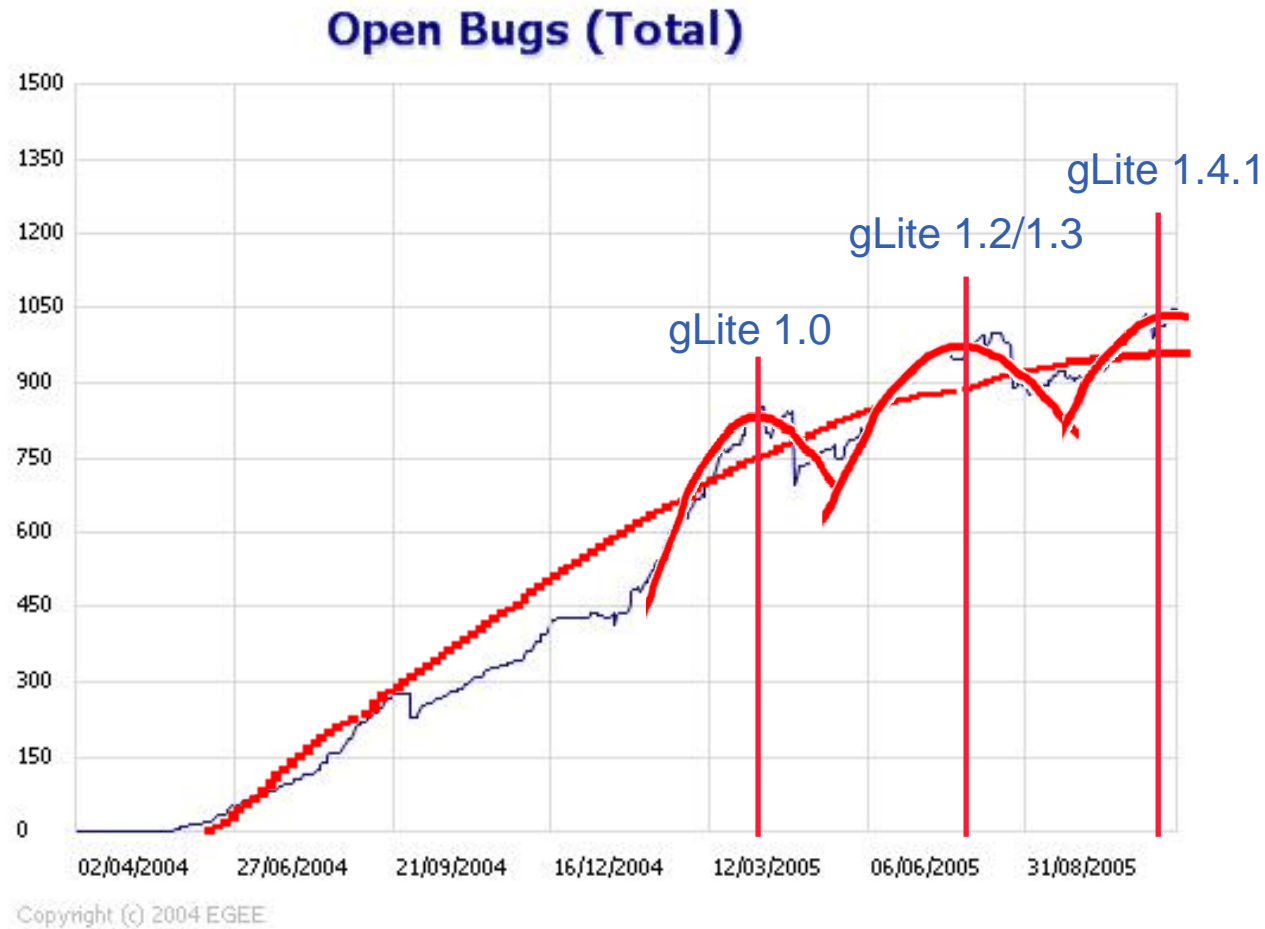


- Execution error (917 - 34.8%)
- Configuration error (479 - 18.18%)
- Documentation error (285 - 10.82%)
- Installation error (242 - 9.18%)
- Crash error (197 - 7.48%)
- Design error (184 - 6.98%)
- Enhancement Request (143 - 5.43%)
- Build error (140 - 5.31%)
- Processes and tools error (48 - 1.82%)

Copyright (c) 2004 EGEE

## The Rayleigh Defect Prediction Model

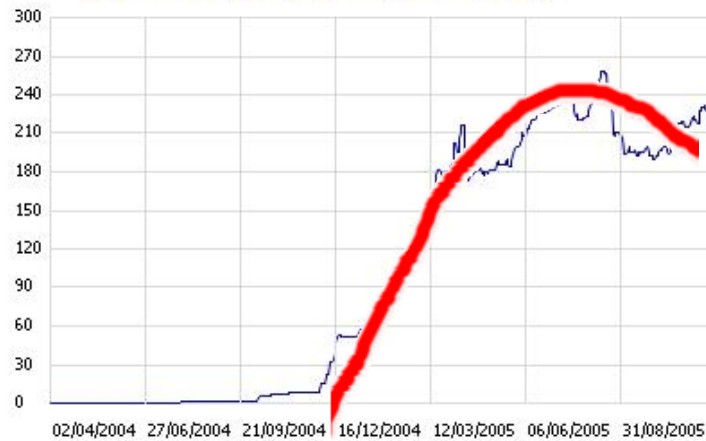




The Rayleigh Defect Prediction Model applied to gLite

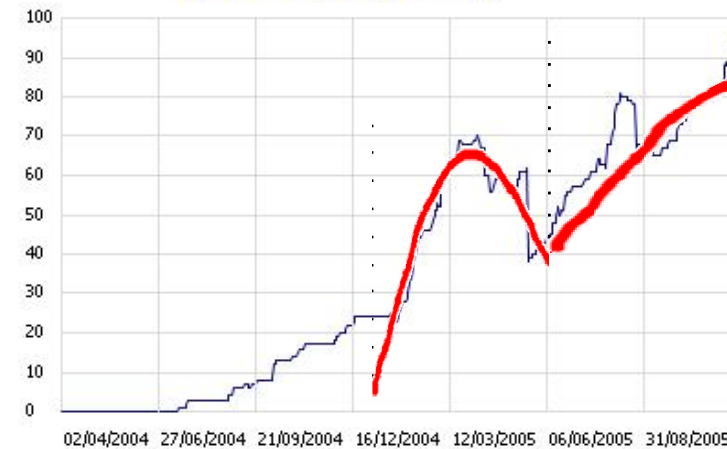


### Open Bugs (Configuration)



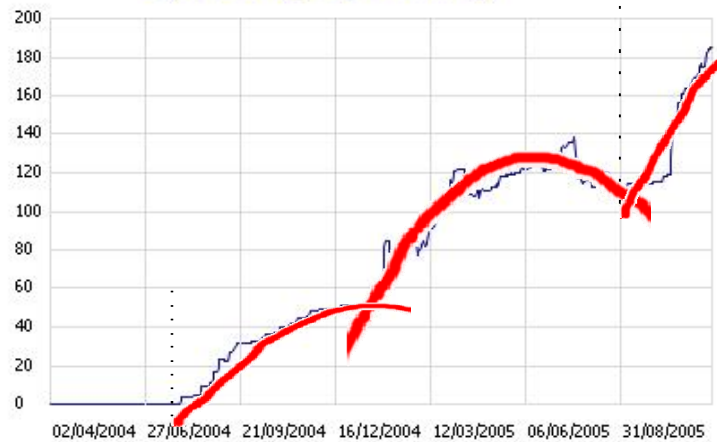
Copyright (c) 2004 EGEE

### Open Bugs (WMS)



Copyright (c) 2004 EGEE

### Open Bugs (R-GMA)



Copyright (c) 2004 EGEE

### Open Bugs (Data Management)



Copyright (c) 2004 EGEE

<http://www.glite.org>

<http://cern.ch/egee-jra1>