Software Engineering Overview

DTI International Technology Service-Global Watch Mission
"Mission to CERN in Distributed IT Applications"
28-30 June 2004

Alberto Aimar
alberto.aimar@cern.ch

SPI: Software Process & Infrastructure
http://spi.cern.ch

Overview of software development activities

- **Users profiles**
  - Most of the developers and users develop software for their work, for themselves or for their group
  - Thousands of end-users write their own software and have a lot of hands-on experience in their domain
- **Experiments infrastructures**
  - Each experiment has its software infrastructure and standards
  - To suit specific experiment's needs
- **Tools and languages**
  - All languages are used C++, Java, Python, Perl, Fortran, shell programming, etc
  - All products in all domains are used
  - Widely used frameworks (Geant 4, Root, etc)
- **Software organization**
  - Each experiment has its own software organization
  - Human factor is extremely important, projects are made of people
  - Different organizations and with different cultures

  - Is a very diverse and heterogeneous environment to support
  - But is also an intellectually rich environment, where users know a lot and can give a lot of help and advice

Example: Context of LCG App.Area software

LHC end- user applications
Thousands of users, writing software
Hundred institutes, universities

LHC core packages and frameworks
(LHC experiments, ~ hundred developers)

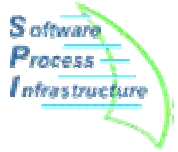
LCG Application Area

SPI:LCG Infrastructure

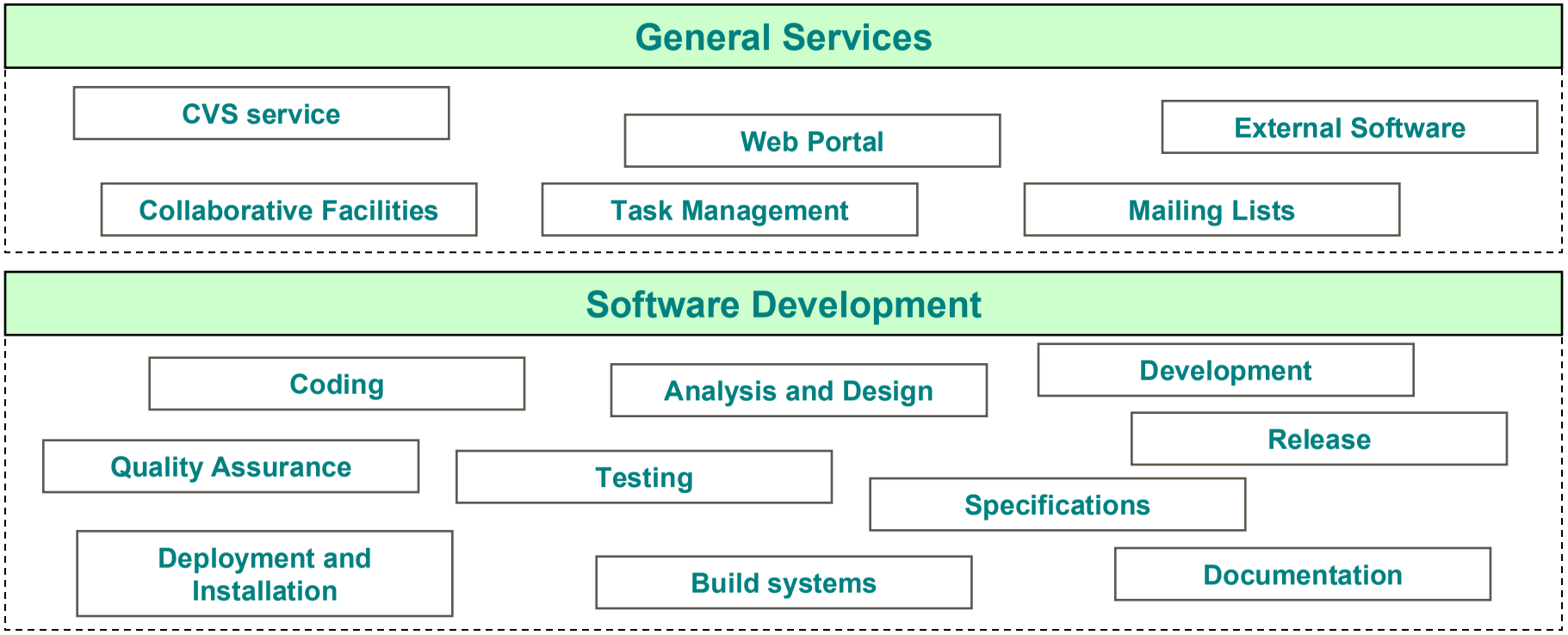
- Common services
- Similar ways of working
- Tools, templates, training
- General QA and tests

LCG Application Area software projects

- SEAL: Core common software
- POOL: Persistency
- PI: Physics Interfaces
- SIMU: Simulation
- …etc…



Software Engineering activities



General services needed by each project

- CVS repository, Web Site, Software Library, Procurement
- Mailing Lists, Bug Reports, Task Management, Collaborative Facilities

Solutions for the software development phases

- Tools, Templates, Policies, Support, Documentation and Examples

Our approach to Software Engineering/Development

- **Have different and separated services**
  - Simple solutions, easy to learn, commonly needed services
  - Leave any process for later
- **Develop as little as possible**
- **Establish simple deliverables**
- **Everything is done starting from existing infrastructure**
  - LCG and LCG projects (Pool, Seal, etc)
  - LHC experiments
  - IT division
  - Big projects (Geant4, Root, etc)
- **We did not start from tools for requirements, design, etc.**
- **We started from development-related work**
  - repository, releases, testing, bug report, etc
- **Work with the users, learn from them**

Example of services (provided for LCG App.Area)

- External Software Installations
- Savannah Project Portal
- Testing Frameworks
- Development of policies, templates
- QA checklists and reports
- Software Distribution
- CVS server and release area management
- Code Documentation (doxygen, lxr, viewcvs)
- Software Configuration
- Software Librarian, builds and releases
- Documentation and Workbook Standards

→ Examples of a few services follow

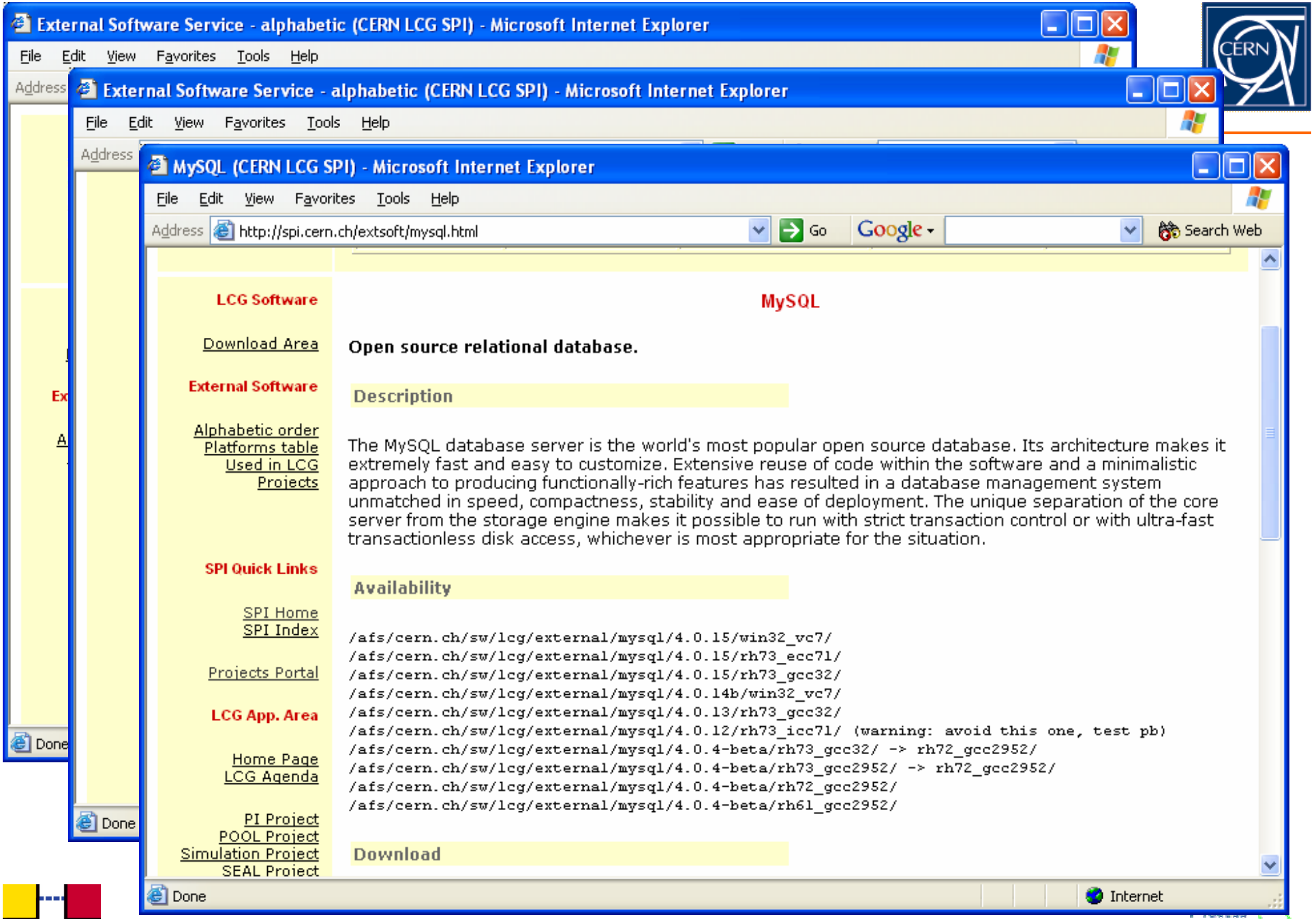
External Software Service

- **We install software needed by projects**
- Open Source and Public Domain software (libraries and tools) like:
  - Compilers (icc, ecc)
  - HEP made packages
  - Scientific libraries (GSL)
  - General tools (python)
  - Test tools (cppunit, qmtest)
  - Database software (mysql, mysql++)
  - Documentation generators (lxr, doxygen)
  - XML parsers (XercesC, gccxml)
- **There are currently 50** different packages**, plus others under evaluation. For more than 300 installations**

The platforms, are those needed by the users

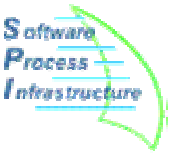
- Linux RedHat 7.3 and **compilers**
  - gcc 3.2 and 3.2.3 (*rh73_gcc32 and gcc323*)
  - icc 7.1 (*rh73_icc71*)
  - ecc 7.1 (*rh73_ecc71*)
- Windows
  - Vis. C++ 7.1: (*win32_vc71*).
- Mac OSX (*osx103_gcc33*)
- Platforms always been reviewed

- **We also provide configuration and installation area**
  - A unique location
  - Standard structure
  - *package_name/version/ platform/package_ content*


External Software Service - alphabetic (CERN LCG SPI) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
External Software Service - alphabetic (CERN LCG SPI) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
MySQL (CERN LCG SPI) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://spi.cern.ch/extsoft/mysql.html
Go
Google
Search Web
LCG Software
Download Area
External Software
Alphabetic order
Platforms table
Used in LCG
Projects
SPI Quick Links
SPI Home
SPI Index
Projects Portal
LCG App. Area
Home Page
LCG Agenda
PI Project
POOL Project
Simulation Project
SEAL Project
MySQL
Open source relational database.
Description
The MySQL database server is the world's most popular open source database. Its architecture makes it extremely fast and easy to customize. Extensive reuse of code within the software and a minimalistic approach to producing functionally-rich features has resulted in a database management system unmatched in speed, compactness, stability and ease of deployment. The unique separation of the core server from the storage engine makes it possible to run with strict transaction control or with ultra-fast transactionless disk access, whichever is most appropriate for the situation.
Availability
/afs/cern.ch/sw/lcg/external/mysql/4.0.15/win32_vc7/
/afs/cern.ch/sw/lcg/external/mysql/4.0.15/rh73_ecc71/
/afs/cern.ch/sw/lcg/external/mysql/4.0.15/rh73_gcc32/
/afs/cern.ch/sw/lcg/external/mysql/4.0.14b/win32_vc7/
/afs/cern.ch/sw/lcg/external/mysql/4.0.13/rh73_gcc32/
/afs/cern.ch/sw/lcg/external/mysql/4.0.12/rh73_icc71/ (warning: avoid this one, test pb)
/afs/cern.ch/sw/lcg/external/mysql/4.0.4-beta/rh73_gcc32/ -> rh72_gcc2952/
/afs/cern.ch/sw/lcg/external/mysql/4.0.4-beta/rh73_gcc2952/ -> rh72_gcc2952/
/afs/cern.ch/sw/lcg/external/mysql/4.0.4-beta/rh72_gcc2952/
/afs/cern.ch/sw/lcg/external/mysql/4.0.4-beta/rh61_gcc2952/
Download
Done
Internet

Build and software configuration

- **SPI provides a common configuration for all releases of LCG software**
  - Which versions of all packages that are used for any given release
- **This is done following the decisions of the LCG Application Area and the needs of the LHC experiments projects**
  - If the release does not provide a clear configuration the software layers above will even not be able to build
  - The role of a central software librarian is to coordinate the software builds and distribution for all supported platforms
- **The experiments build using different build systems developed in house**
  - We do the build using one of those systems
  - Work is going on to study other solutions not developed in house (autoconf/automake tools)
- **We support and generate information for all build systems used by the experiments**
  - Because the experiments need to have the configuration for building their software with their own tool

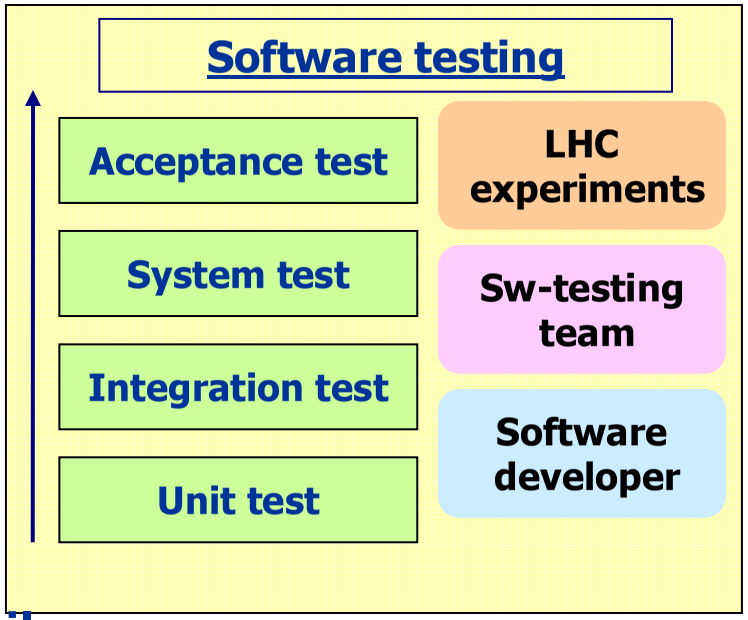


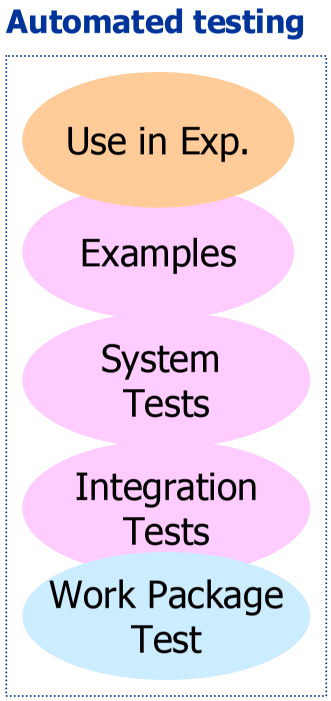
Testing and QA Services

- Software testing should be an integral part of the software development
- The goal was to provide something that can be run automatically as often as needed (releases, development, etc)

SPI provides

- Test frameworks
  - CppUnit, PyUnit, Oval
  - Qmtest
- Test support
- Test policies
- Test doc

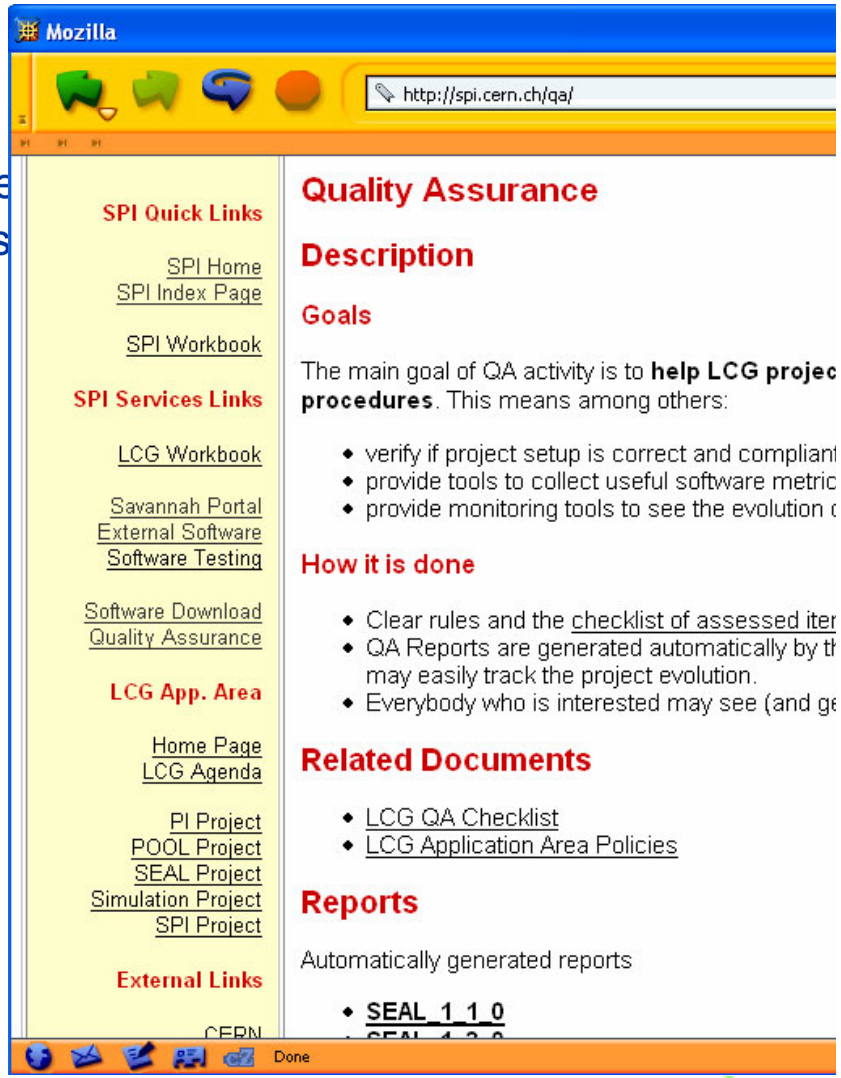




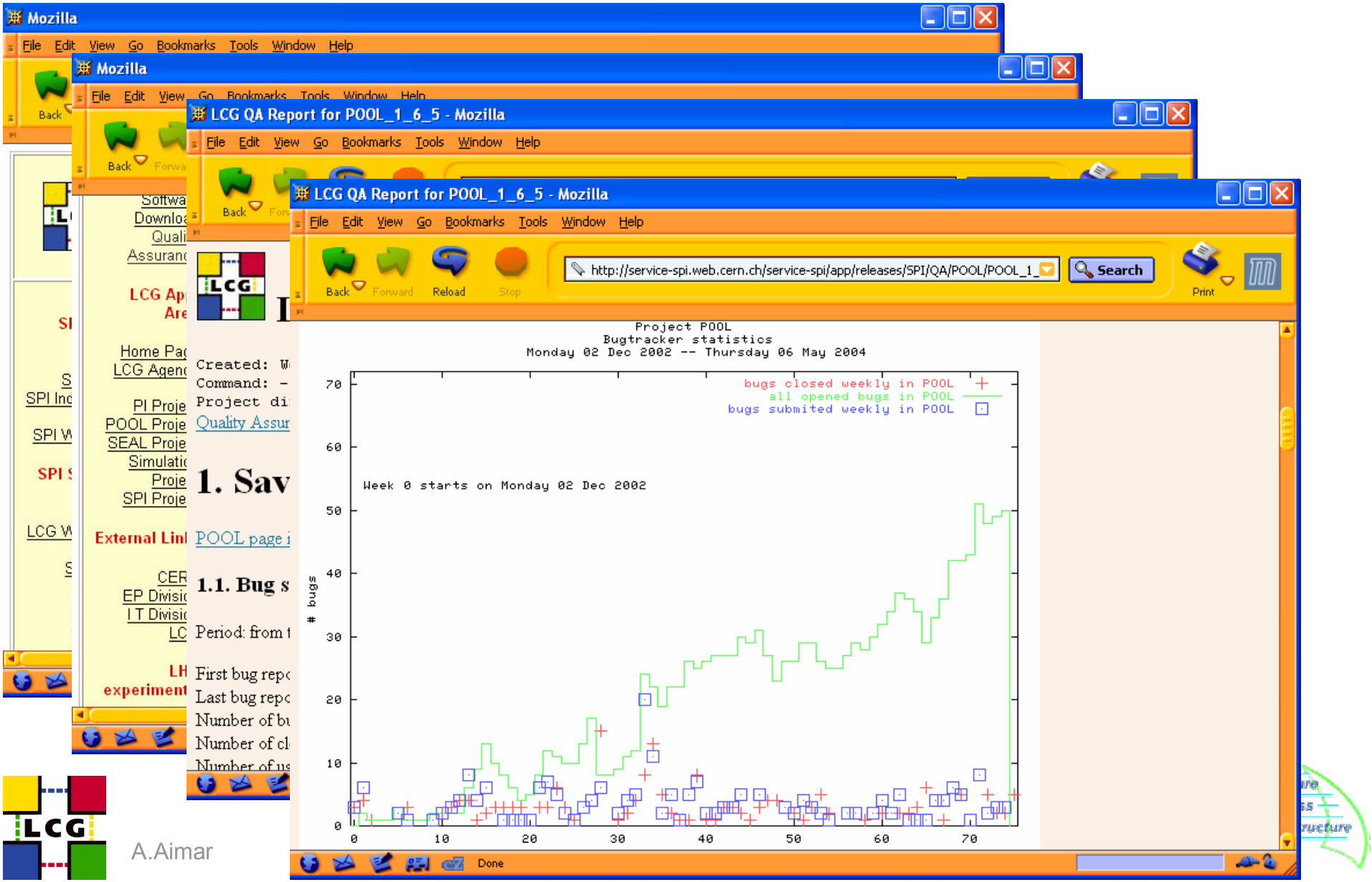
- Different platforms/compilers

Quality Assurance Service - http://spi.cern.ch/qa

- **QA activity is to help the projects not to control them**
  - assess and improve the quality of the software
  - provide tools to collect useful metrics/statistics which help to asses quality;
  - generate reports;
  - verify if project setup is correct with policies.
- **QA Tools and Focus**
  - Automatic reports
  - Development/integration of automatic tools
- **QA Checklist on each Release**
  - Build the release
  - Run automatic tests
  - Code and Test Inventory
  - Documentation/Examples Inventory
  - Savannah Statistics
- **LCG Policies**
  - Configuration of a build system
  - CVS directory structure



Example of QA Report

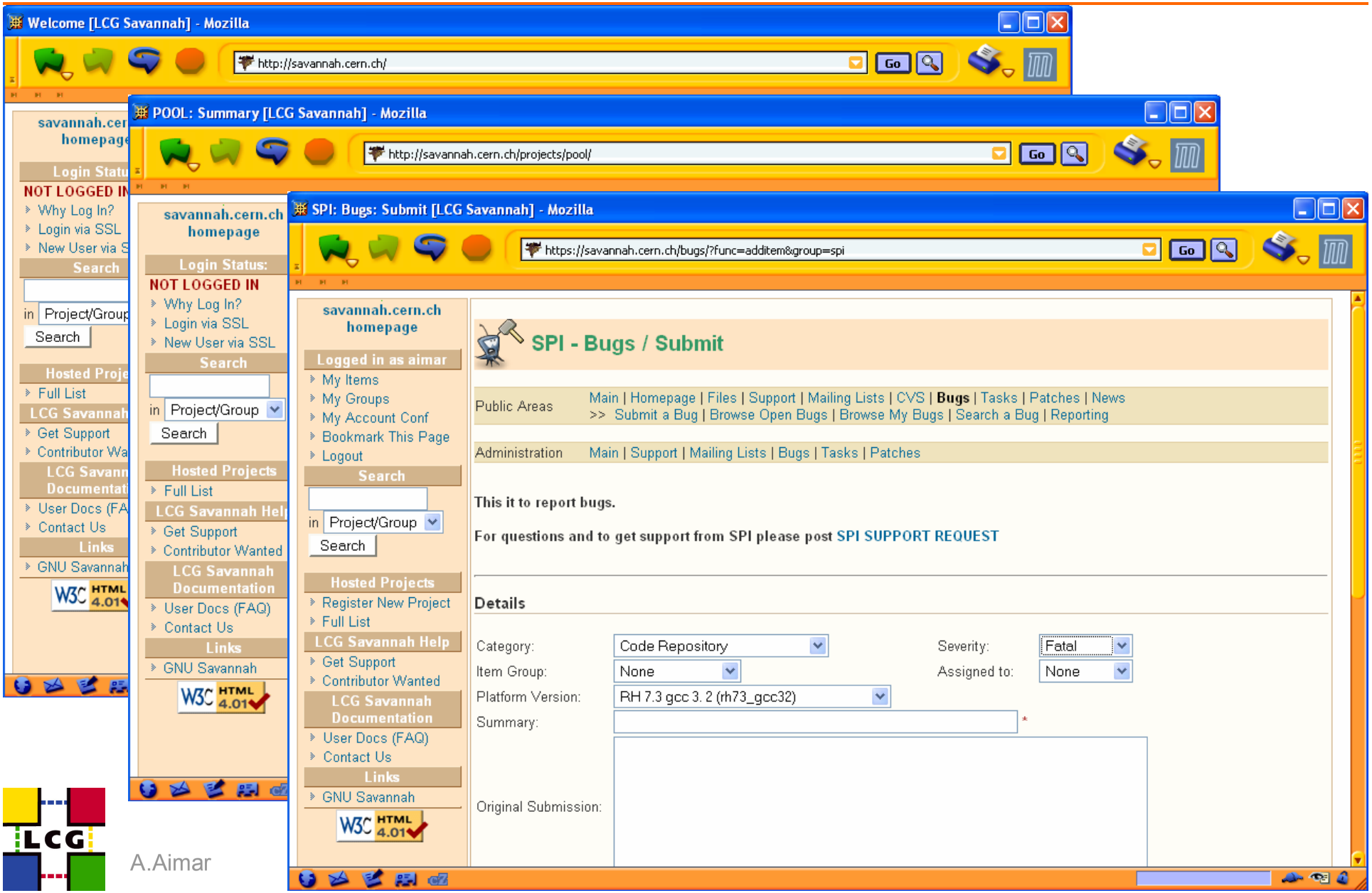


Savannah Portal Service

- **Functionality:**
  - Bug tracking
  - Task management
  - Mailing lists, news, faqs
  - Access to CVS repository
  - Download area, etc
- **The Web portal for software projects**
- **Customized from GNU (SourceForge as origin)**
- **Totally web based**
- **Single entry point to all projects**
- **Uniform access to project information**
- **Set up common web infrastructure for a project without coding**

- **What we changed**
  - installation from GNU, general bug fixing and improvements
  - integration with AFS authentication
  - Integration with standard services already available
- **What we do**
  - administration (project approval)
  - maintenance (submitted bugs)
  - development (support requests)
- **Status**
  - >90 hosted projects
  - >700 registered users

Savannah Service – http://savannah.cern.ch



A.Aimar

Experience with open source

- **Savannah is an example of using a tool from an open source project**
- **Our specific developments were all done by us**
  - One needs to invest manpower to participate and follow the project closely
  - You do not find 'volunteers' to implement what you need: you have to do it
- **Important to establish a collaboration with the open source developers**
  - A lot of these project depend on the "openness" of the developers outside our control: we invite the open source main developer once a year
- **One has to know that still there is work to keep up with the software**
  - No control on the products and on the features, little influence on how and when releases are done
- **But is adaptable to our needs and we can integrated it in our environment (user registration, files access, etc)**
- **Is a good starting point for specific improvements**
  - One would have discussed for months even before starting a savannah-like product
- **Another advantage is that the product/company will not disappear**
- **So even if we do not control it we know that will still be there**

Summary on Software Engineering services

- **Software Engineering is widely spread, in an non-uniform way**
  - Different projects focus on different aspects
  - Depends on the needs of the project and on the people involved
- **There is a set of services that recently have been set up and are used by several projects and experiments**
  - External Software Service
  - Quality Assurance and Policies
  - Savannah Project Portal
  - Software Testing
  - Software Configuration,…etc
- **Provide simple and modular solutions so that projects and experiments can choose and pick what to use**
- **The services are generic and to some extend customizable by the individual projects**
- **Follow a simple strategy**
  - Work with the users and ask for their help
  - Develop as little as possible in order to have little future maintenance

For more information on SPI: **http://spi.cern.ch**