

Deployment

Markus Schulz markus.schulz@cern.ch CERN-IT-GD

Overview



- gLite-3.0
 - History
 - Problems
- New Process
 - Description
 - Status
 - Experience

gLite-3.0 History



- What is gLite-3.0?
- LCG used in production LCG-2 middleware distribution
 - With some gLite-1.x components
 - FTS , VOMS, R-GMA
 - Process for software lifecycle
 - Certification, packaging, configuration management, bug tracking, CVS,
 - Operated on 160+ production sites
 - Used by approx 70 Vos
 - Focus: Stable production environment
 - Very little development work
 - Functionality gaps
- EGEE developed new middleware components
 - gLite-1.x distribution
 - Own process for software lifecycle
 - Operated on a preproduction testbed
 - Focus: Rapid development progress
 - Significant development work
 - Fills many gaps

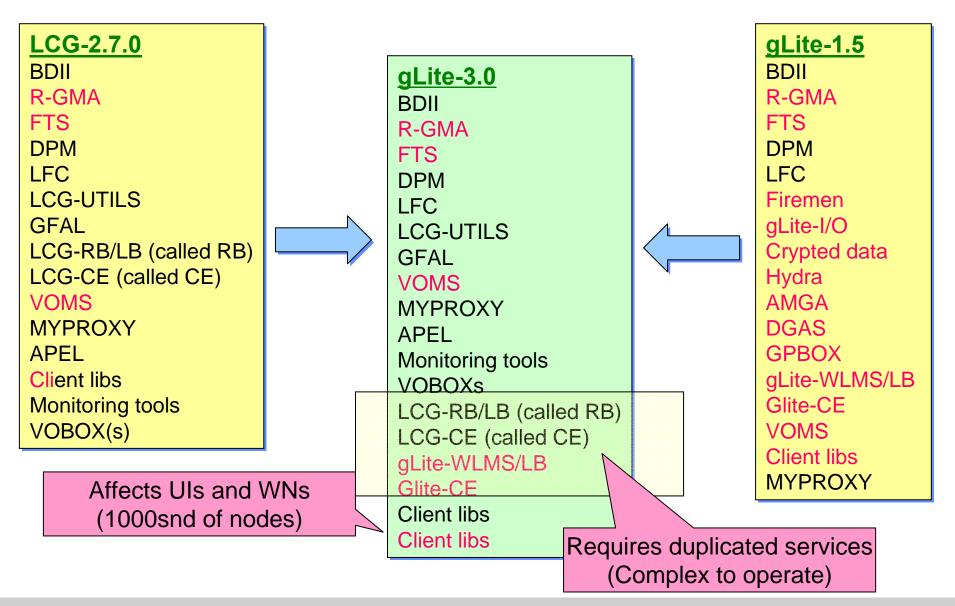
gLite-3.0 History I



- Why not replace LCG-2 with gLite-1.x?
 - Not all gLite-1.x components have been mature enough at the end of 2005
 - Experiments needed time to migrate to new APIs and use new functionality
- Solution: Merge LCG-2.7 and gLite-1.5
 - All LCG-2.7 components
 - Guarantees backward compatibility
 - Mature and critical gLite-1.5 components
 - Workload management first
 - Add more components later
 - Name: gLite-3.0
 - Not next version of LCG-2.7
 - Not next version of gLite-1.5

gLite-3.0 History II





gLite-3.0 History III



Merging processes, tools, and teams gLite-3.0 LCG-build system The harder part...... gLite build system LCG Documentation LCG-2.7.0 gLite-1.5 gLite documentation LCG-build system gLite build system Configuration management LCG Documentation gLite documentation **Configuration management** Procedures Configuration management Dependency management Certification testbed Dependency management Dependency management **Deployment oriented Bug Tracking Deployment oriented** + Testing and certification **Test&Certification** Test&Certification Certification testbed **Configuration management** Testing and certification Dependency management **Bug tracking** Certification testbed Site manager expertise **Developer oriented Bug tracking** Procedures Certification testbed **Bug Tracking** Naming conventions Naming Conventions **Procedures** Compact local team **Distributed Team** Procedures

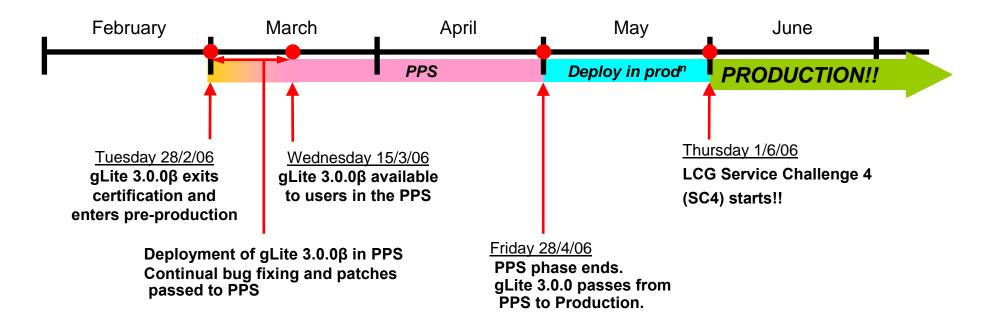
Two processes
With slightly different focus had to be integrated
Teams worked independently until late January
•gLite-1.5
•LCG-2.7.0
•No time for in depth integration of process

Bug Tracking Procedures Procedures Naming conventions Naming Conventions New team+ external developers

gLite-3.0 History IV



- Detailed planning started end January
- July 1st deadline
- Plan announced to sites in LCG-2.7 release notes
 - January 31st



gLite-3.0 History V



What happened

- 20th Feb; freeze
- 3rd March, beta released to PPS
- All of March, deployment on PPS
 - close monitoring and creation of RC2
 - PPS not available for users
- 11th Apr RC2 hits the PPS
 - too late, but what's the average bug lifetime until integration?
- Apr updates and patches
 - PPS sites are trying to run a stable service;
 - Less than ideal conditions for testers
- Before eastern: ROC deployment testing (5 ROCs volunteered)
 - CE ROC, IT ROC, UK ROC, EGEE-SEE.
 - #16388 #16355 submitted
- 4th May gLite 3.0 released to production
 - Staged deployment: 2 waves of Tier1s within 2 weeks

gLite-3.0 History V



- 5th June 8 'blah' CEs, >50 sites have installed WNs and Uis
- 2 weeks after release
 - 1st upgrade (configuration tools)
 - Fixes for relocatable UI/WNs
 - Many documentation glitches fixed
- Ind June: Full day <u>postmortem</u> (follow link for more details)
 - Aggregate fixes and release 'bundled' upgrades
- 16th June: gLite-3.0.1
 - 1st upgrade ready
 - □ gLite WLMs, CE, UI, WN
- 21th August: gLite-3.0.2
 - Bugfixes
 - Better localisation support
- 28th August: Configuration patch
- 5th September: Security patch (globus)
- 19th September: Set of patches (WMS, FTS, dpm, LFC..)

gLite-3.0 Problems



- (Fixed release date + Fixed set of components) := Problems
- Release had still significant deployment issues
- The release hit most sites unprepared despite:
 - Release had been announced in January + weekly status updates
- The release and upgrade notes confused sites
 - First large update since 2 years
- The staged rollout made everyone wait
 - No trailblazers (small to medium sites that upgrade rapidly)
- There was not enough time left for localization
 - One month not enough for large sites to integrate new services
 - Work planning, resource allocation
 - Adaptation to local fabric management and batch systems
 - Internal testing of new services/releases
- gLite-WMS and gLite-CE first time on large scale production
 - Stability, performance under high load, (as all new services)

gLite-3.0 Problems I



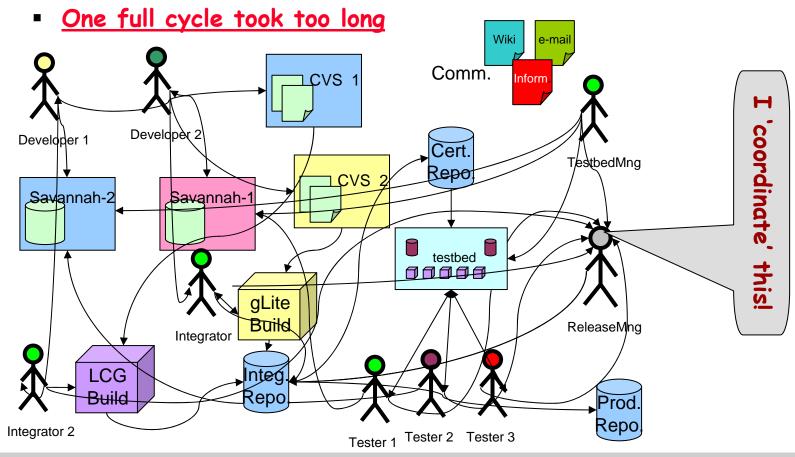
Communication

- Between sites and the release preparation team
- Between the release team and the developers
 - Ad hoc synchronization via Wiki page, frequent meetings
- Two processes (worse than none!!)
 - Non uniform tracking
 - Bugs, patches and release candidate tracking was unreliable
 - Naming conventions
 - Communication
 - Required: Temporary ad hoc process
 - Based on frequent informal communication
- Merging configuration tools
 - Complex failures due to configuration 'interference'
- Bundling of many fixes into one upgrade
 - Slowest patch holds back important patches

gLite-3.0 Problems II



- Two (incomplete) sets of tests
 - Linked manually + via Wiki
 - Tests require full setup of testbeds
- Time



Markus. Schulz @cern.ch

LHCC Review September 2006

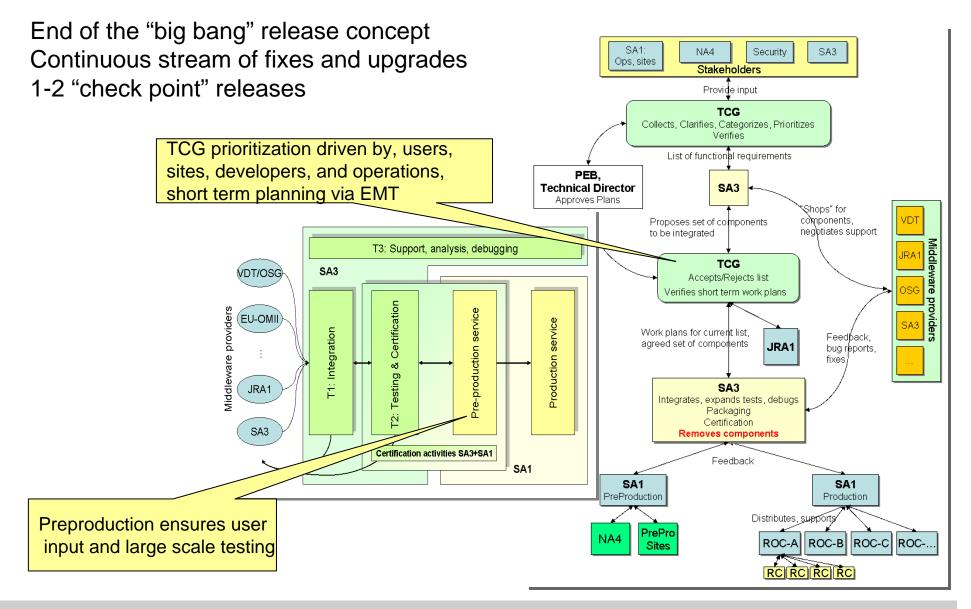
gLite-3.0 Problems III



- Where are we now?
- Communication
 - Short term coordination and planning via EMT
 - Twice a week, once with Condor developers
 - Medium term via TCG
 - Every two weeks
- One process
 - Single improved tracking system (Savannah)
 - Component centric
- Merging of tools started
 - CVS merged
 - Move to ETICS build system started
- Tests
 - Inventory
 - Missing tests have been identified
 - Assigned to partners
 - Tests are moving towards common analysis tool
 - Testbed restructuring started, using virtualization

Process





Markus. Schulz @cern.ch

Process I



- The Software Process
 - How we should be working
 - The different roles with defined responsibilities
 - The interaction between the different roles
- Creates a primary information source
 - The reference for all knowledge on problems and solutions
 - Implemented in Savannah
 - Traceability of the problems and the solutions
- Not written in stone
 - If we find a problem with the process
 - Analyze the problem and improve the process
 - Must follow the process correctly

Process II



Terminology

- Component
 - The smallest self-contained package (e.g. one rpm)
- Subsystem
 - A logical group of components (e.g. R-GMA, WMS)
- Baseline
 - The full list of components that make up a release.
- Two distinct entities, Problems and Solutions \odot
 - Problems = Bugs
 - Solutions = Bug Fixes = Patches
 - New features are tracked as "Enhancement"
 - Missing feature = Problem
- Well defined roles and interactions
 - Subsystem Bug Manager, Developer ,Subsystem Integrator
 - Integration Manager, Certification Manager, Pre-production Manager,
 - Production Manager EMT TCG Release Manager
- Communication tracked via Savannah

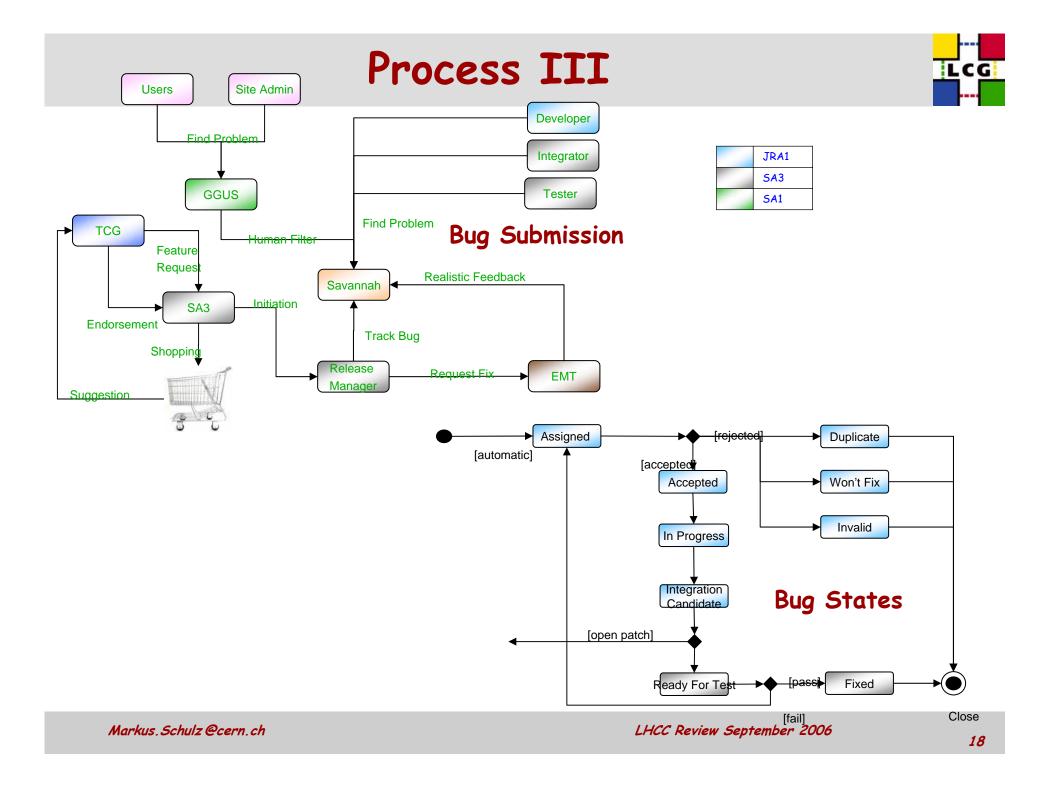
Process II



Operating System	Externals	External	Internal
	Packages	Middleware	Middleware

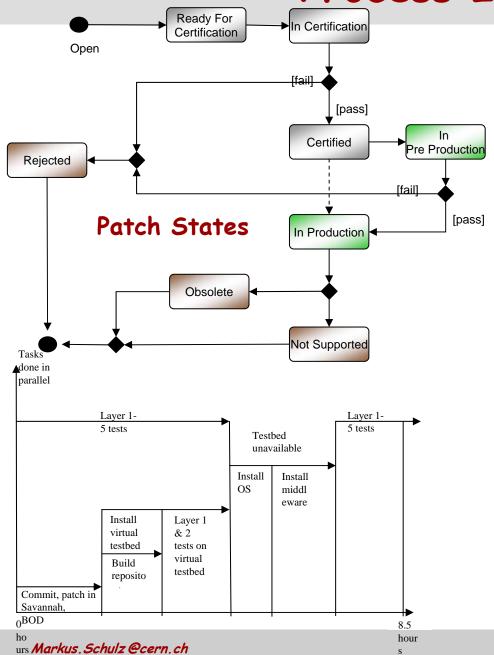
The software spectrum:

- Most software is provided as a package
 - Only internal middleware needs to be built from CVS
 - Require mapping rule from package name to CVS tag
- Need to integrate at the package level
 - View every thing as external --> decoupling of components
- Defined configurations (meta packages) for
 - Service Types
 - Nodes Types
- A release is a set of packages that define a **baseline**
 - Updates are defined relative to the baseline
 - The baseline contains a core (like kernel + gcc version for linux)
 - Changes to the core affect backward compatibility (not service)
 - Require new release
 - Preview testbed gives access to next baseline



Process IV





- Progression of patches depends on:
 - Actions (builds, etc)
 - Tests
 - Meeting criteria
 - Defined in check lists
- Test strategy
 - Multi level tests
 - To abort as early as possible
 - Virtualization
 - To save time
 - Upgrade and install
 - First local then external testbeds

Process Status

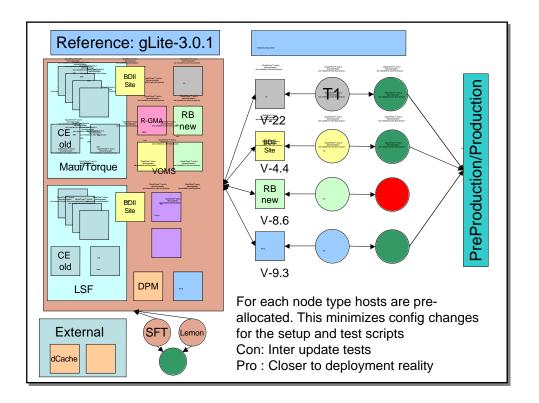


- Process tracking has been implemented and is in use
- Independent update of components
 - Practiced where possible, but:
 - Affects configuration mangement
 - Development under way (October)
 - Affects testbed management
 - In progress, relies heavily on virtualisation (cooperation with openlab)
 - Affects certification (running tests)
 - In progress
 - Affects build system
 - Will take effect with the move to the ETICS build system (October)
 - Affects repositories
 - Done
- Changes have to be introduced while providing service
 - Not: stop, restructure, restart

Process Experience



- Tracking works very well
- Testing and integration are slowing the process down
 - Still 'mini big bang' releases
 - Will improve with ETICS
 - Testing restructuring underway



Markus.Schulz@cern.ch

Process Experience I



- Biggest Problems
 - Process tailored for relative stable production environment
 - Many components still in development phase
 - Many shallow bugs
 - Need for fast change cycles
 - Testing on testbeds insufficient
 - Scale (Vos peak at 50k jobs/ day)
 - Localization creates many different deployment scenarios
 - Can't be all modeled

Current approach

- Special experimental production systems
 - 6 weeks of close collaboration with developers of WMS
 - Increased reliability and performance (x8)
 - Needs discipline and rigorous tracking
- External testbeds to cover localizations
 - Currently build up with EGEE SA3 partners
- Out of process patches for central services
 - Dangerous, but sometimes necessary

Next Steps



- In parallel, both finished by the end of the year
- Finish the implementation of the new process
 - Build system, tests, testbeds......
 - Formalize the scalability tests in the production environment
- Prepare next major release (was gLite-3.1)
 - Moving to SL4
 - Moving to VDT-1.3
 - Releasing all client libs ready for 64bit
 - Adding new components as agreed with the TCG

Summary



- gLite-3.0 ready for release on time
 - Despite CHEP, and kick offs (ETICS, EGEE-II, egee- extension projects...)
- gLite-3.0 contained agreed components
 - With almost full functionality, scalability and stability were still problematic
- Merging 2 stacks and processes was as hard as expected
 - Still not 100% finished
- New process implementation is progressing well and is used
 - CVS, tracker, build....
 - Addresses communication problems within release prep and with developers
- Testing is still not where it should be in terms of coverage and ease of use
 - Structural progress has started
 - Outside contribution started to arrive
 - Massive scalability tests problematic
- Documentation
 - Still needs much more effort.
- Several core components are still in development mode
 - A patch a week