

Table of Contents

WLCG EGEE Pre Production: Service Description.....	1
1 Scope of this document.....	2
2 General description of the EGEE Pre-Production Service.....	3
3 Actors and roles.....	4
4 Functional tasks and workflows.....	5
4.1 Middleware Pilot Services.....	5
4.1.1 Fully backward compatible client update.....	6
4.1.2 Non-backward compatible client update.....	7
4.1.3 Backward compatible server update.....	8
4.1.4 Non-backward compatible server update.....	11
4.2 Middleware Quality Services.....	13
4.2.1 Overview.....	13
4.2.2 Pre-deployment test.....	13
4.2.3 Release Testing.....	14
5 Non-functional tasks and workflows.....	15
5.1 Coordination.....	15
5.2 Support Area.....	15
5.2.1 Release Management.....	15
5.2.1.1 Release Cases.....	15
5.2.1.2 Release Procedures.....	15
5.2.2 Activity Management.....	16
5.2.2.1 Activity Registry.....	16
5.2.3 Metrics and Quality.....	16
5.2.4 Communication.....	16
6 Appendix1 - List of PPS pre-defined activities.....	17

WLCG EGEE Pre Production: Service Description

1 Scope of this document

This page describes the internal organisation of the EGEE/WLCG Pre-Production Service.

In particular actors, roles, systems, interfaces, workflows and detailed tasks needed for the implementation of the use cases described in PreProductionUseCases are detailed.

In addition to that, this document develops also the second major use case of the pre-production, the distributed gLite deployment testing.

The document is addressed to ROC managers, site managers in the pre-production orbit, and members of the middleware certification and release teams (SA1/SA3). It provides guidelines for the communication among different partners involved at various title in the pre-production activity. Therefore its final version, as well as each major revision, must be subject to the approval of representatives of:

- Operation Coordination Centre (SA1)
- Certification and Release teams (SA3)
- EGEE ROC Managers (SA1)

Minor changes, namely those dealing with purely technical details, may be decided by the Pre-Production Coordination and notified to the concerned partners.

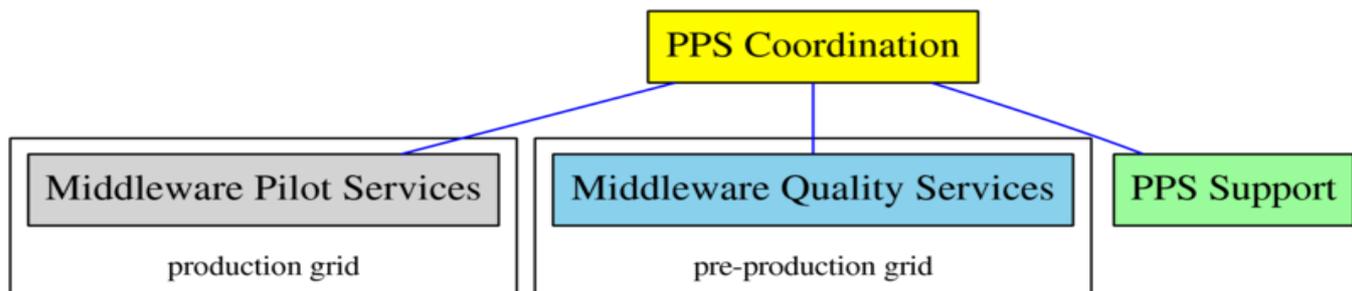
2 General description of the EGEE Pre-Production Service

The EGEE Pre-Production provides access to grid services in preview to WLCG/EGEE interested users, in order to test, evaluate and give feedback to changes and new features of the middleware.

In addition to that, the pre-production extends the middleware certification activity, helping to evaluate deployment procedures, [inter]operability and basic functionality of the software against operational scenarios reflecting real production conditions

The service is organised in two **functional areas** - *Middleware Quality Services* (MQS) and *Middleware Pilot Services* (MPS) - which are meant to pursue the aforesaid main objectives, and one **support area**, including all the services/activities needed in support to that (e.g. coordination, release management etc.)

The three service areas are staffed by the EGEE regions and respond to the PPS Coordination.



[ps]

3 Actors and roles

Resources to implement the workflows described later on in this document are from:

- EGEE/SA1: Operations Teams (OCC + EGEE Regions)
- EGEE/SA3: Integration, Testing and Release Teams (CERN + partners in the EGEE Regions)
- EGEE/JRA1: gLite Middleware Development
- VOs: represented by EIS team for HEP VOs and EGEE/NA4 for non-HEP ones
- TMB: EGEE Technical Management Board

The roles are:

- PPS Coordinator
- Regional Manager
- ITR contact: a contact person in the Integration, Testing and Release Team
- Developer: a contact person from the developers' teams
- Release Manager: a member of SA3 responsible for the content and distribution of the gLite release
- PPS Repository Manager: The maintainer of special software repositories used in PPS
- PPS *member* site: a grid site supporting the Middleware Quality Services (deployment testing, release testing, PPS monitoring infrastructure). Sites in this category normally advertise their grid services in the pre-production Information System
- PPS *partner* site: a grid site **in production** supporting the Middleware Pilot Services (support to pilots, hosting new client versions). Sites in this category may belong to the production or the pre-production Information System.

4 Functional tasks and workflows

In this section the functional tasks for sites, regions and coordination bodies are described, as well as the workflows in the context of which they are run.

Together with the task description, a basic estimation of the needed effort is given. The estimate provided is based on the past-two-year experiences both from PPS and from the *experimental production services* activities.

Apart from the planning purposes, the basic "value" of a task will be used in PPS also in order to measure the work performed by the various contributors upon completion of the task, as described in the paragraph Activity Management.

The units used are (FTE = Full Time Equivalent):

- **PH** = 1FTE x hour = 1 person hour
- **PD** = 1FTE x day (8 hours)
- **PW** = 1FTE x 5 working days
- **PM** = 1FTE x 20 working days

The conversion table is (of course):

- PH = 1 person hour
- PD = 8PH
- PW = 5PD = 40PH
- PM = 4PW = 20PD = 160PH

4.1 Middleware Pilot Services

The mission of the Middleware Pilot Services is, in general, *to offer previews of new middleware functionalities to interested users*.

More particularly, the middleware previews can be further distinguished into two major classes:

- Previews of client tools (changes/additions affecting Worker Nodes, User Interfaces, VOBOXes)
- Previews of grid services (a.k.a. *pilot* or *experimental* services)

Updates and new releases affecting the two classes are treated in two completely different ways as it is detailed in the next sections.

New versions of both client tools and middleware services are made available and selectable to users from the VOs in the production environment (using the production information system and accessing production resources). While for the clients this happens in a semi-automated way, instances of grid services are created only on-demand, meaning that use cases, scope and goals of each pilot have to be agreed in advance between the interested VOs and the PPS.

In other words, there is no more a permanent instance of every service made available to users by default as it was the case in the previous implementation of the PPS.

A lightweight involvement of the regions is required in support to the process of distributing new clients, basically limited to make a certain number of sites accessible by the automated distribution tools

A more significant involvement may be required to the regions in support to pilots of services, namely in order to

- set-up the pilot service(s)
- manage of the pilot (interface with VOs during the exploitation, wrap-up results)
- wrap-up feedback for middleware release team

It is however to be noticed that the effort needed to run pilot services is not be allocated by the regions on a permanent way but rather shifted, upon demand, from the production tasks.

In order to simplify the process of finding, upon demand, a site available to run a pilot, a pre-registration to certain set of tasks is required.

The partner sites are requested to express their potential interest in the pilot activities in advance and to specify the services they are interested to support. The commitments are recorded in the Activity Registry and the sites are invited to join based on the preferences they expressed and the actual need of the pilot infrastructure.

4.1.1 Fully backward compatible client update

The detailed use case can be read in the Pre-Production Use Cases document and is included below.

Show Use Case Hide Use Case

Definition: By *backward compatible client update* we mean specifically those clients updates where no variables (e.g. YAIM variables) are to be changed and **no new variables** are added to configuration (*).

A predominant fraction of client updates falls in this category, the notable characteristics of which are:

- New clients are compatible with old servers
- Updates often related to bug fixes time-to-production is more important for VOs
- Empty set of configuration instructions in release notes extended test of release notes in PPS not needed

The new clients are distributed, possibly **before the certification is complete**, to collaborating sites **in production**, and made available for the VOs to test. As a precondition to be distributed, the software must have passed a first round of basic certification tests although the full certification process may be not completed yet. The patches in certification that have reached this level of "maturity" are identified and flagged (*) by SA3.

The installation of new clients at the sites does not affect the existing production instances, so that, the site is still fully functional for the production work.

These "preview" installations inherit the same local settings used by the production clients (e.g. environment settings in profile.d).

There is a dedicated tag in the production information system (using the attribute

`GlueHostApplicationSoftwareRunTimeEnvironment`) to announce that a site supports these non-certified releases.

The distribution, installation and publishing of independent versions of clients is handled by a centralised mechanism.

The "live" map of the distribution of new versions over the various sites is available in a web page in the pre-production website. Users from the VOs are able to select the desired version using a particular requirement in the jdl. This feature is available by default only to users submitting **through the gLite WMS**. Jobs submitted bypassing the WMS will use instead the standard "production" version of the clients.

The deployment of a new client version in pre-production starts when the *release manager* (from SA3) decides that a patch in certification (Status="In certification") can be "moved" to this deployment area. As a consequence of this decision, the following actions are triggered and carried-out **in parallel with the standard certification**

1. Creation of the tarball
2. Distribution to the sites (ideally using a SAM job)
3. Local testing at the sites (ideally using a SAM job)
4. Publication of the tag in the information system ideally using the same SAM job and through the *lcg-ManageVOTag* command)
5. Update of the "release bulletin" documenting the versions available at the various sites (this can be done automatically on the PPS web site based on the information extracted from Savannah)
6. General notification to potentially interested subjects (e.g. broadcasts to VOs and ROCs).
7. Personal notification to specifically interested subjects (e.g. the originator(s) of the bug(s)/request(s) fixed by a patch released) with the invitation to verify the provided solution.

Immediately after the deployment of a new client version, a dedicated public channel is made available to users to provide feedback. The feedback provided is taken into account within the parallel certification process and eventually it is propagated and summarised in the release notes. In particular the release notes mention explicitly the case in which **no feedback** was provided.

After the parallel certification and preview phases are completed, if the release notes are confirmed not to contain any special configuration information, and limited to the platforms on which an installation test was done in certification, the release is deployed in production with no further deployment testing.

Workflows and tasks to be detailed

4.1.2 Non-backward compatible client update

The detailed use case can be read in the Pre-Production Use Cases document and is included below.

Show Use Case Hide Use Case

Definition: client updates where interventions on the environment or extra configuration is needed.

The relevant characteristics of this category of updates are:

- Updates often related to new features time-to-production must comply to VO schedules but not dramatic from the service point of view
- Configuration instructions may be needed in release notes pre-deployment test of release notes is needed
- New clients are still compatible with old servers. The case of incompatibility is dealt with together with the case of non backward compatible server update

From the VO perspective the way this use case works is exactly the same as the previous one, with some considerations needed.

A local configuration actions are in general needed, the new clients cannot be simply "pushed" to the sites by the deployment team as in the previous case. So a longer elapsed time for deployment has to be expected.

In fact the operations to be performed after the decision to deploy the client in preview is taken are:

1. *Creation of the tarball*
2. Distribution of the client to a number of selected production sites (PP "Silver" partners) (ideally using a SAM job)

3. Local configuration of the clients at the sites
 4. *Local testing at the sites (ideally using a SAM job)*
 5. *Publication of the tag in the information system (ideally using the same SAM job and through the `_lcg-ManageVOTag` command)_*
 6. *Update of the "release bulletin" documenting the versions available at the various sites (this can be done automatically on the PPS web site based on the information extracted from Savannah)*
 7. *General notification to potentially interested subjects (e.g. broadcasts to VOs and ROCs)*
 8. *Personal notification to specifically interested subjects (e.g. the originator(s) of the bug(s)/request(s) fixed by a patch released) with the invitation to verify the provided solution*
-

The pilot is meant to allow the functionality testing, whereas the deployment test in the PPS infrastructure, based on YAIM and release notes is focused on several deployment scenarios (OS, architectures). The two activities run in parallel.

Workflows and tasks to be detailed

4.1.3 Backward compatible server update

The detailed use case can be read in the Pre-Production Use Cases document and is included below.

Show Use Case Hide Use Case

Definition

With *backward-compatible (BC) server update* we mean updates that are compatible with the existing clients and don't introduce new functionalities. No changes in database schema. Backward compatible updates can in general be rolled back with not relevant information loss. A further distinction in this category is done between *minor* and *major* service updates

Minor: The following conditions have to be *all* true in order to consider an update as "minor"

- no new configuration parameters anywhere (neither in YAIM nor into component-specific configuration files)
- less than 2 "major" plus 5 "normal" bug fixes (according to the severity assigned by the EMT).
- the changes introduces correspond to not more than 2 man-days of programming (this assertion has to be validated by the release manager via a specific attribute in Savannah)
- not significant operational changes are introduced for the service administrators

Major: when any of the above conditions is false.

General policies

Pilot services (aka experimental services) are eventually set-up and run in production upon agreement of the concerned parts (VO, development, certification and operation teams). The purpose of the pilot is to speed-up the process of delivering to production a fully certified and functional service. In this view it is recommendable to set up and upgrade the pilot services using only certified software. Exceptions to this rule may be decided by the concerned parts for justified opportunity reasons. The use of non-certified software in the pilots has however to be justified, documented and recorded in order to safe-guard the reproducibility of the working environment and the integrity of the future releases.

In both major and minor cases, pilot services are set-up and run in production by a number of selected partner sites identified as *PP "Gold" partners*. VOs get explicitly involved in the activity only in case of major updates. In case of minor updates the pilot activity is kept internal to the service infrastructure .

Pilot services for **minor updates** will be operated in production for 1 week. No artificial or focused "solicitation" of the service will be created. Only standard production activity + monitoring. In this case, running the pilot will be exactly like running the production service. The only extra commitment requested to the supporting production site is the awareness that the service is still experimental, so a prompt reaction is requested in case of problems in order to roll-back.

For **major updates** a preliminary negotiation between VOs, deployment teams and sites is necessary in order to agree on terms and conditions of the pilot activity. The negotiation is chaired and followed-up by the preproduction coordination team. The different phases of this negotiations consist into:

- identify suitable candidate sites among the PP Gold partners (or volunteers)
- provide information about the new features to the VOs and letting them express their interest into participating to the pilot activity. This is done through several channels, e.g. announcement during WLCG/EGEE Operations meeting; broadcast to VO Managers; direct communication to Experiment Integration and Support team (EIS).
- restrict (eventually) the rose of candidates/options and call a meeting to kick-off the deployment activity, During this meeting an agreement has to be reached about the timeline of the site to set-up the service (e.g. 1 week) and the VO to give feedback (e.g. 2 weeks). Eventually the VOs may ask to be able to identify and select the sites providing the pilot services via jdl.

Once the pilot is started reminders for feedback are regularly sent to the VOs by the preproduction coordination.

The feedback provided by the VOs is taken into account, followed-up and eventually summarised and propagated the release notes by the release team.

In case no VOs commit to be active on the pilot service, or no feedback is received within the agreed timeline, the pilot service is evaluated with the same success criteria in use for the minor updates, and the decision to go to production is made internally by the operations team. In that case the release notes mention explicitly that **no feedback from users** was received during the pilot service.

As part of the preproduction service, deployment test of the update over significant deployment scenarios are run in parallel and separately from the pilot activity.

In case of updates to the services needed to fix critical bugs in production or security vulnerabilities the aforesaid policies may be overruled by joint decision of the EMT and PPS coordination.

The basic workflow described in the use case above is developed here with explicit mention of the connected tasks. A summary table with the relevant subtasks estimated effort and task rating is given at the end.

Steps and the numbers written in green are relevant only for *minor* updates.

Steps and the numbers written in red are relevant only for *major* updates.

Steps written in black are common to the two cases.

The initiator or owner of the action is indicated in square brackets at the beginning.

1. [ITR and/or Developer (via EMT), VO]: Forward the request for a new pilot to the PPS coordinator
 - ◆ EMT: this is normally the case for minor updates
 - ◆ VO: if a pilot service is requested by a VO it is normally dealing with a major update of the functionality (or perceived as such by the user). The requested is forwarded either contacting pps-support@cernNOSPAMPLEASE.ch or during one of the regular operations meetings (WLCG/EGEE Operations meeting, WLCG Service Coordination Meeting)
2. [PPS Coordinator]: pre-screen available resources
 - ◆ identify suitable candidates sites among the sites (Sites registered in the Activity Registry +

- volunteers)
- ◆ provide information about the new features to the VOs and let them express their interest into participating to the pilot activity. This is done through several channels, e.g. announcement during WLCG/EGEE Operations meeting; broadcast to VO Managers; direct communication to Experiment Integration and Support team (EIS).
- ◆ restrict the rose of candidates/options and select the sites to run the pilot
- ◆ verify that adequate documentation is available for installation, configuration
- 3. [PPS Coordinator]: Contact the sites and give instruction to start the pilot
 - ◆ give pointers to documentation
 - ◆ agree on timelines
- 4. [PPS Coordinator]: Organise a pilot kick-off meeting with the sites, SA3, the developers (if needed), the VOs
- 5. [Site(s), ITR, PPS Coordinator, VO, Developers] Participate to the pilot kick-off meeting
 - ◆ The goal of the meeting is to reach an agreement about the timeline for the site to set-up the service (e.g. 1 week) and for the VO to give feedback (e.g. 2 weeks). Particular requirements from the VOs are also expressed in this meeting.
- 6. [PPS Repository Manager] Set-up a mirror repository. This stem may be not needed in some cases, but it is likely that, if the service is going to be published in the production information system, the repository used for the installation will be mirrored in order to decouple the production environment from changes possibly happening in the original one provided by the developers.
- 7. [Site(s)] Set-up the pilot service . It is likely that, if the service is going to be published in the production information system, the repository used for the installation will be mirrored from the original one provided by the developers. In that cas alos the set-up of the mirror repository has to be considered
- 8. [Site(s), ITR and/or Developers] : Run the pilot service
 - ◆ The site manager is generally meant to act as a production **service manager**, the only special commitment with the PPS activity being the prompt reaction in case of problems or in the event of a roll-back
 - ◆ ITR and Developers may be called to help the site as **service experts** especially in support of new and undocumented features. Sometimes, especially if the site is "close" to developers, the role may end-up to be covered by the site administrator itself.
 - ◆ Changes in the system however should only be applied by the site manager, who is finally responsible for the feedback given to the release from the operational point of view
 - ◆ Any subsequent change/issue in the system following the first set-up should be also notified in copy to the PPS Coordinator (through the pps-support mailing list)
 - ◆ The default duration of the pilot (after completed installation) is fixed to 1 week for minor updates and 2 weeks for major updates. Exceptions or different requirements can be discussed individually during the preliminary phases. The PPS Coordinator is in charge of checking periodically and eventually to send reminders.
- 9. [Site] : Produce post-mortem pilot report (set-up description, non-functional issues e.g. resource consumption, stability): 4PH
- 10. [Site, ITR, Release Manager, PPS Coordinator, VO, Developers]: a "post mortem" or "wrap-up" meeting is done upon success or expiration of the timeline previously agreed. In this meeting an assessment is done and a decision is made about the follow-up (including decision to prolongate the testing time). Eventually general guidelines for the deployment can be drafted. In case of minor updates this discussion in brought to the EMT

Tasks and Effort

Site	Produce post-mortem pilot report (set-up description, non-functional issues e.g. resource consumption, stability)	0.5	4	-
<i>Who</i>	<i>What</i>		<i>Credits</i>	<i>Notes</i>

		<i>effort (PD)</i>		
PPS Coordinator	Pre-screen of resources for pilot start-up	0.75	6	-
PPS Coordinator	Instruct the sites to start the pilot	0.25	2	-
PPS Coordinator	Organise a pilot kick-off meeting	0.25	2	-
PPS Coordinator, site(s), ITR, VO	Participate to kick-off meeting (preparation, attendance, follow-up)	0.25	2	
PPS Repository Manager	Set-up a mirror repository	0.5	4	-
Site	Set-up the pilot service	1.5	12	-
Site	Provide 1 week of support as service manager	1	8	averaged data from CREAM and WMS experimental services
Site, ITR or Developers	Provide 1 week of support as service expert	2	16	averaged data from CREAM and WMS experimental services

[PPS Coordinator, Site, ITR, VO] Participation to wrap-up meeting, including preparation and follow-up|0.25|2| - |

NOTE: A post-mortem review of two long-lasting pilots (1 year and 100 days) has shown that the overall coordination can safely be estimated as ~0.4 PDs/week

Based on the table above an estimation of the integrated effort to be spent by SA1 to run a 3-week pilot in PPS is

Total SA1 effort (for a 3-week pilot):

- PPS Coordination: 1.25PD
- Site: 5.5PD
- *Service expert support: 6PD (to be counted if provided by the site or SA1 personnel)*

4.1.4 Non-backward compatible server update

The detailed use case can be read in the Pre-Production Use Cases document and is included below.

Show Use Case Hide Use Case

Definition

With *backward-compatible (BC) server update* we mean updates that are compatible with the existing clients and don't introduce new functionalities. No changes in database schema. Backward compatible updates can in general be rolled back with not relevant information loss. A further distinction in this category is done between *minor* and *major* service updates

Minor: The following conditions have to be *all* true in order to consider an update as "minor"

- no new configuration parameters anywhere (neither in YAIM nor into component-specific configuration files)
- less than 2 "major" plus 5 "normal" bug fixes (according to the severity assigned by the EMT).

- the changes introduced correspond to not more than 2 man-days of programming (this assertion has to be validated by the release manager via a specific attribute in Savannah)
- not significant operational changes are introduced for the service administrators

Major: when any of the above conditions is false.

General policies

Pilot services (aka experimental services) are eventually set-up and run in production upon agreement of the concerned parts (VO, development, certification and operation teams). The purpose of the pilot is to speed-up the process of delivering to production a fully certified and functional service. In this view it is recommendable to set up and upgrade the pilot services using only certified software. Exceptions to this rule may be decided by the concerned parts for justified opportunity reasons. The use of non-certified software in the pilots has however to be justified, documented and recorded in order to safe-guard the reproducibility of the working environment and the integrity of the future releases.

In both major and minor cases, pilot services are set-up and run in production by a number of selected partner sites identified as *PP "Gold" partners*. VOs get explicitly involved in the activity only in case of major updates. In case of minor updates the pilot activity is kept internal to the service infrastructure .

Pilot services for **minor updates** will be operated in production for 1 week. No artificial or focused "solicitation" of the service will be created. Only standard production activity + monitoring. In this case, running the pilot will be exactly like running the production service. The only extra commitment requested to the supporting production site is the awareness that the service is still experimental, so a prompt reaction is requested in case of problems in order to roll-back.

For **major updates** a preliminary negotiation between VOs, deployment teams and sites is necessary in order to agree on terms and conditions of the pilot activity. The negotiation is chaired and followed-up by the preproduction coordination team. The different phases of this negotiations consist into:

- identify suitable candidate sites among the PP Gold partners (or volunteers)
- provide information about the new features to the VOs and letting them express their interest into participating to the pilot activity. This is done through several channels, e.g. announcement during WLCG/EGEE Operations meeting; broadcast to VO Managers; direct communication to Experiment Integration and Support team (EIS).
- restrict (eventually) the rose of candidates/options and call a meeting to kick-off the deployment activity, During this meeting an agreement has to be reached about the timeline of the site to set-up the service (e.g. 1 week) and the VO to give feedback (e.g. 2 weeks). Eventually the VOs may ask to be able to identify and select the sites providing the pilot services via jdl.

Once the pilot is started reminders for feedback are regularly sent to the VOs by the preproduction coordination.

The feedback provided by the VOs is taken into account, followed-up and eventually summarised and propagated the release notes by the release team.

In case no VOs commit to be active on the pilot service, or no feedback is received within the agreed timeline, the pilot service is evaluated with the same success criteria in use for the minor updates, and the decision to go to production is made internally by the operations team. In that case the release notes mention explicitly that **no feedback from users** was received during the pilot service.

As part of the preproduction service, deployment test of the update over significant deployment scenarios are run in parallel and separately from the pilot activity.

In case of updates to the services needed to fix critical bugs in production or security vulnerabilities the aforesaid policies may be overruled by joint decision of the EMT and PPS coordination.

4.2 Middleware Quality Services

4.2.1 Overview

Mission: “To test the middleware deployment tools (packaging, documentation) against scenarios relevant for production”

- Workload distributed among 'PPS' sites (not registered as production)
- Services published in pre-production IS
- Testing interaction with different platforms, batch and storage systems
- Contributing to interoperability testing
- Providing additional info and advice for deployment in production
- Dedicated monitoring infrastructure for validation
- "service-oriented" testing: several deployment test managers sharing the tools
- NOT production-like service --> pre-deployment runs on-demand with releases

task for regions

- set-up and run test services in different deployment scenarios
- set requirements for deployment scenarios coming from the sites and local VOs
- evolutionary maintenance of monitoring infrastructure for validation
- evolutionary maintenance of automated distribution tools
- deployment testing management (run tests and wrap-up feedback, per-service)

coordinations tasks

- Deployment testing coordination (tools and procedures) ? delegated
- Set requirements for deployment scenarios coming from middleware development/certification and "global" VOs
- Release management (tools and procedures) ? delegated
- Interface to EMT

MPS interface with VOs, regions, sites and middleware providers for the definition and kick-off of new pilots

4.2.2 Pre-deployment test

The deployment testing is a lightweight test of the gLite packaging done within the PPS infrastructure. The test is focused on the installation and configuration methods and may include some bit of functionality testing, depending on the service. The test is performed by a specialised administrator as soon as a package gets into PPS. The typical workflow of the deployment test is:

- A new middleware update is released to PPS.
- The test coordinator sends out to a dedicated mailing list test report templates for each affected service, asking the test responsible to perform the test
- The service managers responsible for the test perform the upgrade using the documentation provided for PPS and reports using the template.
- The test coordinator collates the individual reports in a single summary one.
- The full report made publicly available by publication on the PPS website

This test usually does not take to the site more than one hour to be completed (including the reports) and it has to be repeated each time a software update affecting a service is released to PPS.

In order to take part to this test the site should belong to the PPS infrastructure (registered in the GOCDB as a PPS site, publishing in the PPS BDII). This is due to the fact that the tools used to verify the correctness of the installation are pointed to the PPS information system and not the production one.

Tasks and Effort

PPS Site Admin	Pre-deployment test of one service + report	0.375	3	Possibly including minor workarounds
<i>Who</i>	<i>What</i>	<i>effort (PD)</i>	<i>Credits</i>	<i>Notes</i>
SAM Client Administrator	Clone a SAM Sensor	1	8	
SAM Portal Administrator	Create a display for a new sensor	0.5	4	-
SAM Client Administrator	Provide 1 week of support as SAM-client service manager	0.5	4	Possibly including minor re-configurations
SAM Client Administrator	Provide 1 week of support as SAM-server service manager	0.5	4	Possibly including minor re-configurations

4.2.3 Release Testing

The *Release Testing* is a complementary step of the process of releasing middleware Updates to the production system . Before the middleware updates are released to the public, they are delivered to a number of selected production sites. Sites involved in release testing may be part of pools performing similar task at regional level. The sites apply the update to one or more of their grid services and provide feedback to the release managers. Details on the integration of the release testing with the release procedure are available in PPSReleaseProcedures#PPS_Production.

Tasks and Effort

<i>Who</i>	<i>What</i>	<i>effort (PD)</i>	<i>Credits</i>	<i>Notes</i>
Production Site	Apply a middleware update for release testing	0.125	1	-

5 Non-functional tasks and workflows

5.1 Coordination

5.2 Support Area

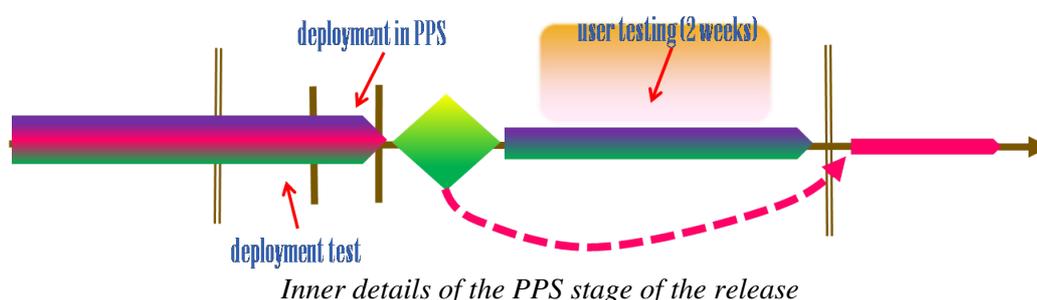
5.2.1 Release Management

5.2.1.1 Release Cases

Rule: Ideally each new update to the gLite middleware is supposed to be released in PPS first, to stay there for a given amount of time, available to the users; then to be delivered into production.

The amount of time the software spends in PPS is not strictly defined

- If no activity is scheduled or in progress on a service the service is moved to production with the next release
- If user testing is ongoing and producing useful results the stage in PPS can be stretched
- The components of one PPS update are treated independently, so delays in processing one component don't affect the others (see picture below)



Exception(s): In occasion of **exceptional** events, the standard flow can be overruled. We have categorized the "special" releases as follow:

- *"Urgent" releases:* (E.g. fixes for a disclosed vulnerability). These patches are released at the same time both in production and PPS (with priority to the production service)
- *"High-priority" releases:* (E.g. fixes for a broken service). These patches are released directly in production, upon the notion that, if a service is already broken, things cannot get worse. High-priority releases are generally issued following-up operational problems. The PPS is synchronised soon after.

5.2.1.2 Release Procedures

The following release workflows are implemented within the pre-production service. The detailed steps of each workflows are available in the PPS Release Procedures document

Certification PPS (Standard Releases):

The *PPS Release Team* gives a "green light" to the *gLite Release Team* for the update of the repository. The *gLite Release Team* updates the repository and sends out release notes. The release is processed by PPS (deployment test, internal distribution), A new "green light" is given for the next update.

Details

Certification Production (Urgent and Fast-Tracked Releases):

The decision to make a fast-tracked release normally follows a very serious issue affecting the production system (e.g. sever security incident or critical break of functionality). The deployment method of a fast-tracked release is generally undefined, but in general it requires some formal steps in the standard release procedure to be skipped. The Operations Coordination Centre (occ-grid-support@cern.ch) and the gLite Release team (gd-release-team@cern.ch) together are ultimately responsible for the approval of the update as well as of the deployment methods.

The *gLite Release Team* initiates the release to production and notifies the PPS in order to synchronise the two environments and eventually for support with release testing. Details

Certification Production (Urgent and Fast-Tracked Releases):

The *PPS Release Team* takes actions in order to synchronise PPS environment after a fast-tracked release

Details

PPS Production:

The *PPS Release Team* periodically recommends to the *gLite Release Team* a list of components which are rated mature for production. The list constitutes the starting point for the next release to production.

Details

5.2.2 Activity Management

5.2.2.1 Activity Registry

5.2.3 Metrics and Quality

5.2.4 Communication

6 Appendix1 - List of PPS pre-defined activities

The list of pre-defined activities is maintained in a central database.

It can be checked through the PPS Website [↗](#)

This topic: LCG > PreProductionServiceDescription

Topic revision: r30 - 2013-08-24 - TWikiGuest



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback