SA3

Oliver Keeble
CERN

SA3's responsibilities

- **"The goal of the SA3 activity is to manage and coordinate the process of building deployable and documented middleware distributions, called gLite, starting with the integration of middleware packages and components from a variety of sources".**
- **From build of release candidates to release**
  - No development (well, not much)
  - No rollout (well, not much)
- **Implementation**
  - We operate a process which acts on **Patches**, which are self-consistent changes to the middleware stack

SA3 in Numbers

Manpower: EGEE III, 17 institutes, 33 FTE

Short Name	EGEE-II	EGEE-III
CERN	432	396
PSNC	36	24
TCD	19	36
IMPERIAL	24	0
INFN	60	96
UKBH	12	0
UCY	34	12
GRNET	24	30
CSIC	12	8
PIC	24	12
CESGA	12	12
CESNET	0	24
FOM	0	24
UH.HIP	0	12
JINR	0	10
PNPI RAS	0	10
SINP MSU	0	10
STFC	0	36
ASGC	0	40
FZJ	36	0
Total	**725**	**792**

EGEE III 17 partners 33 FTE

8 new partners

Significant resources co-located with JRA1 effort (including CERN DM)

Approx 1FTE per partner

SA3 work plan has to reflect this

Communication overheads must be minimised... but work must be properly reported

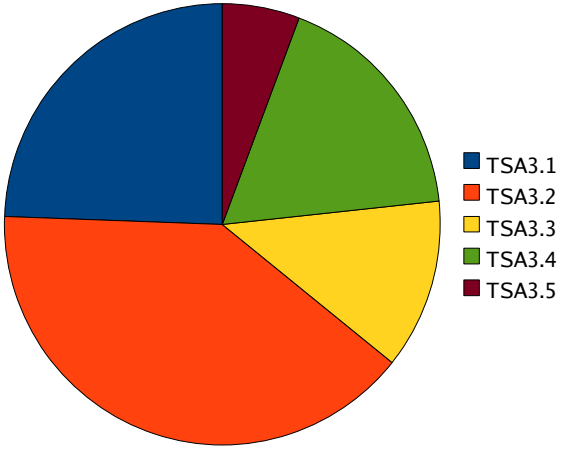
Execution Plan Highlights

- Clusters of competence (== patch preparation)
- Patch certification by partners
- Multiplatform and batch system support
- Interoperability
- Build management
- Release delivery
- Infosys, service discovery

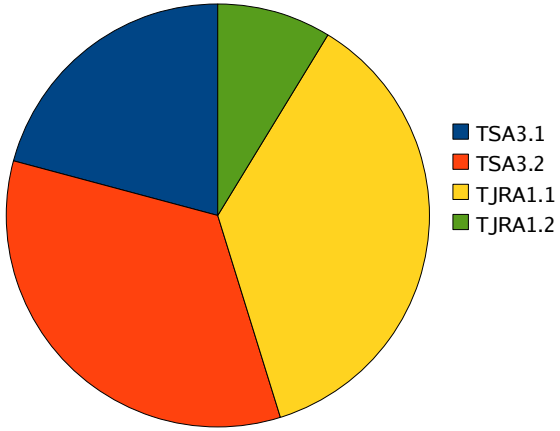
All with the transition to a sustainable infrastructure in mind

Tasks

- **TSA3.1: Integration, configuration and packaging (186PM)**
- **TSA3.2: Testing and certification (319PM)**
- **TSA3.3: Support, analysis, debugging, problem resolution (100PM)**
- **TSA3.4: Interoperability & Platform support (141PM)**
- **TSA3.5: Activity Management (46PM)**



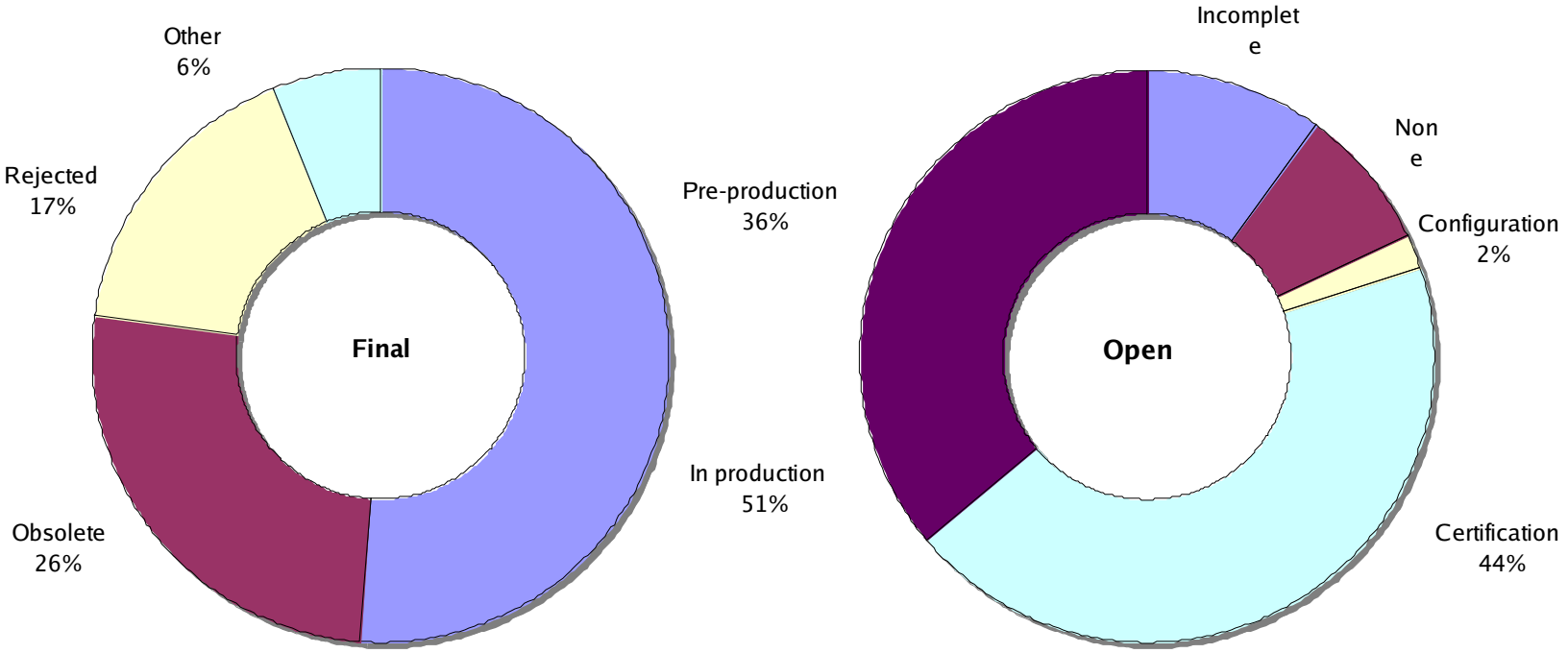
Distribution of tasks in SA3



Software change management SA3/JRA1

Patches – stats from EGEE-II

- Total: 482 Closed: 432 Open: 50



Clusters of Competence

- **Focus on pre-release testing and patch preparation**
  - We have a detailed proposal on what this role entails
    - All necessary work to produce a successful Patch
    - Should also include test writing and multiplatform work
  - This will avoid the expensive overhead of the full process
- **Will still do an independent validation phase**
  - But this will be light
  - Can be re-evaluated on the basis of experience
  - Can identify the certifier early so questions can be asked before patch submission

Partner Patch certification

- **The certification process can be parallelised**
  - We can only succeed if we take advantage of this
  - We have 9 partners involved in certification
    - Not all are full time, but this gives considerable scope
- **Clear criteria for certification have been documented**
  - This includes checklists for each component
- **Active followup and coordination will be needed**
- **Participation will all be logged and reported**
- **Will do a demo at the forthcoming all-hands**

Testing infrastructure

- **Framework**
  - Long term direction of production monitoring
- **Regression Tests**
- **Virtualisation**
- **ETICS**
  - Move testing as far upstream as possible
  - Deployment tests
  - Regression tests
  - Unit tests

Multiplatform support

- **'Porting' and 'multiplatform support'**
  - Crucial Difference
- **Focus on specific service/platform combinations**
  - clients
- **Next platform: Debian 4**
  - SL5
  - Other definitions of 'platform'
- **Regular builds in ETICS**
  - Have some requirements on ETICS for full support
- **Multiplatform support starts in the codebase**
  - But there are assumptions everywhere
  - We know that JRA1 has little effort to spare
- **Effect on release process (1 change, multiple effects)**
- **TCD produced a 10 step porting guide**

Porting builds

Grid-Ireland Local ETICS Build of gLite-3_1_0

Builds using ETICS version: 1.3.4-1

Worker Node Build Status													
ARCH	OS TYPE	VERSION	DISTRO	VDT	deps	GridIre	Basic	RGMA	VOMS	DM	gfal	torque	WN-head
ia32	CentOS	4	yum	3/3	30/30	2/2	12/12	41/41	14/14	17/17	20/20	3/3	105/107
	SuSE	10	apt	0/1	21/21	1/2	N/A	N/A	N/A	N/A	N/A	3/3	N/A
	CentOS	5	yum	1/1	24/24	2/2	12/12	41/41	14/14	17/17	20/20	3/3	93/107
	Debian	4	debs	1/1	28/28	1/1	12/12	41/41	14/14	16/17	16/20	3/3	94/107
	Solaris	10	pkg/tarball	1/1	23/23	2/2	12/12	33/41	0/11	7/17	7/20	3/3	N/A
x86_64	CentOS	4	yum	1/1	26/26	2/2	9/9	41/41	15/15	18/18	21/21	3/3	108/108
	SLES	9	apt	1/1	24/24	1/1	9/9	37/41	10/15	12/18	12/21	3/3	76/111
powerpc	Mac OS X	10.4	fink/tarball	1/1	23/23	1/1	12/12	33/41	0/11	0/17	11/29	3/3	55/107
	Mac OS X	10.3	fink/tarball	1/1	23/23	1/1	12/12	32/41	0/11	0/17	11/29	3/3	58/108
	AIX	5	rpm/tarball	1/1	21/21	1/1	10/11	0/6	0/4	7/17	7/20	3/3	N/A

Batch system integration

- **Have a 'batch coordination' partner (NIKHEF)**
- **Middleware now interacts with a number of batch systems**
  - Necessary to grid-enable the maximum number of resources
- **Require;**
  - blah/cream plugins
  - Information providers
  - Documentation
  - Accounting
  - Glexec issues
  - yaim updates
  - testing
- **This does NOT necessarily include configuration or documentation for the batch system itself; this is not middleware**

The build

- **Look forward to an ETICS which allows fast builds**
  - This is a prerequisite for all the rest
    - 3 hrs to checkout a subsystem does not allow fast turnaround
- **Source objects**
  - rpms, tarballs
- **Subsystems**
- **Build management**
  - Developer triggered builds
  - Our build definition is currently inconsistent
    - Some packages in the release are not there
    - Some packages there are not in the release
  - Detailed policy currently being refined in EMT

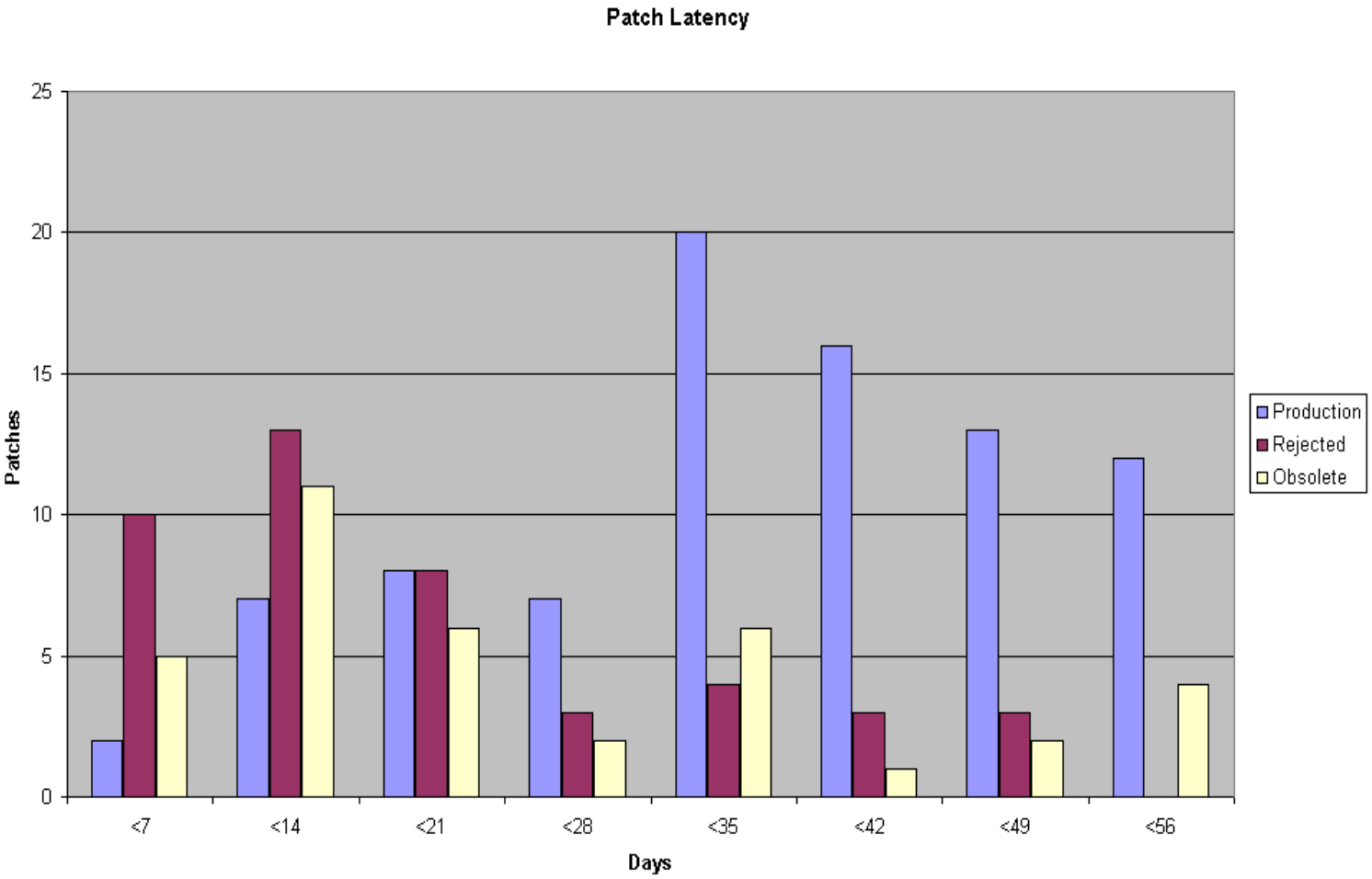
QA, task tracking, reviews

- **Will continue with the reviews we did in EGEE-II**
  - One per partner during the project
- **Improve task tracking**
  - Will negotiate end dates for all tasks
    - Try to create objective criteria for task completion
    - Patches will be evaluated before work begins
  - Want to make the hard work visible
  - Need to be able to identify problem areas

Greasing the wheels

- **Is certification slow?**
  - The question is really whether the inevitable delay imposed is worth the value added
- **The day you can remotely install a tier-2 is the day certification can be completely automated**
  - For now manual steps are necessary
  - On delivery, many patches fail
- **Clusters of competence will help a lot**
- **Partner patch certification**
- **Process automation, savannah etc**
  - Programmatic interface to savannah
    - reject/clone instead of recycle
- **PPS (under review) and release scheduling**
- **Flow control - more efficient batching of fixes**

Patch Latency



The gLite distribution

- **Why do we have an integrated distribution at all?**
- **New approach to the clients?**
  - The requirements are pretty different for the clients
  - We can distribute them like experiment software
  - All porting efforts are concentrated here
  - Are currently made available as tarballs too
- **Externals**
  - DAG, jpackage
- **'Useful stuff'**
  - hierarchical release
  - RESPECT
- **Direct release of services**
  - dCache

Summary

- **Ultimate aim is to grid-enable the maximum number of resources through releases of reliable, portable middleware**
- **All this with the transition to a sustainable infrastructure in mind**
  - What we set up in the next 2 years must be optimised for sustainable operation on low resources