

Reorganized Software Support

Aleš Křenek, CESNET

OMB, November 20, 2012

- TPM (TSA1.7) – 1st line support
 - handle incoming tickets in GGUS
 - resolve very trivial issues
 - reassign to 2nd line support units
- DMSU (TSA2.5) – 2nd line for middleware
 - attempt to reproduce the problem
 - request further information from the user
 - understand the source of problem (or narrow it down)
 - provide workaround and reassign to 3rd line
- several other 2nd line units
 - SAM, APEL, batch systems, ...
- 3rd line support units
 - handled by developers teams

- overlaps in TPM and DMSU work
 - organization of shifts to provide quick response
 - common tasks on ticket escalation etc.
 - involvement of the same people/institute
- two-fold 2nd line
 - DMSU vs. other 2nd line units
 - lack of communication
 - complicated ticket routing (back through TPM)
- true 1st line provided elsewhere
 - most end-users problems handled by NGIs and VOs
 - tickets going to GGUS (TPM) already filtered
- unbalanced effort allocation
 - overestimated adoption of ARC and UNICORE in EGI

- timeline
 - first proposed and discussed in March 2012 (EGI CF Munich)
 - elaborated in EGI doc. #1104 “Revision of TPM and DMSU activities”
 - finalized in MS511 (EGI doc. #1134)
 - approved by project review
 - implementation started in September 2012
- merge TPM and DMSU in a single team
- include support for “non-middleware” software
- optimize effort distribution among partners

- unified 1st line support
 - quick response to incoming tickets, close trivial ones
 - distinguish operational and software issues and reassign
 - contact specific supporters at high priority tickets
 - uses TPM SU, handled by INFN (several people)
- 2nd line experts independently
 - handle tickets as asked by 1st line, high priorities in particular
 - communicate with user, reproduce, analyse, etc.
 - legacy DMSU SU, all partners (except of KIT)
- 2nd line “hands on” meetings
 - over jabber, Tue and Thu 2pm, 1–3 hours
 - 3–5 experts in most frequent components, others invited on demand
 - revisit all open tickets (more complicated remain here)
 - discussions, new ideas, further tests, ...

- feedback to operations
 - issues with potential broader impact
 - summarized at wiki, reported at Monday operations meetings
 - tracking high priority tickets with 3rd line
- ticket follow-up and oversight
 - remind non-responding user (done at “hands on meetings”)
 - remind non-responding supporter (high priorities in particular)
 - check ETAs
 - generic task, done by KIT

- “top priority”
 - disaster, paralyzes large part of infrastructure
 - immediate reaction by sw. support
 - limited analyses, assess the priority only
 - typically 4 hour reaction by 3rd line (SLA)
 - specify ETA, as soon as possible, emergency release
- “very-urgent”
 - significant impact, no workaround known or feasible
 - 1–2 days reaction and analysis by sw. support
 - 2 days reaction by 3rd line
 - ETA up to 45 days, next scheduled release

- “urgent”
 - visible impact, only some patterns of work, workaround available
 - 2–5 days reaction by sw. support
 - 5 days reaction by 3rd line
 - no ETA assigned, “to be fixed in reasonable time”
- “less urgent”
 - easy workaround, marginal impact
 - 2 weeks analysis by sw. support
 - 15 days reaction by 3rd line
 - no ETA assigned, fixed on best effort

- ETA tracking – high priorities
 - 2nd line suggests ETA, 3rd line confirms, negotiate eventually
 - reminders on approaching deadline, eventual renegotiation
 - “ETA violations” performance metric
- unsolved low priority tickets
 - ever-increasing backlog when TP capacity is exceeded
 - “Fedora approach” to close expired tickets
 - older than 6 (or more?) months, release of affected component in-between
 - on UMD releases produce summary of expired tickets
 - close as “unsolved” after 2 weeks
 - process still to be deployed, some support in GGUS required

- changes in software support proposed and approved
 - better utilization of available effort
 - implementation of foreseen ticket follow-up processes
- most of changes already implemented
- 2 months experience, smooth work
- further details at https://wiki.egi.eu/wiki/EGI_DMSU
 - (still to be updated)