Process to describe how to release grid middleware to the production infrastructure: basic ideas

- Provided by the EGI.org MU
  - Management: They are responsible for driving the process (e.g. Release Manager?)
  - Software: All mw clients and services should follow a proper versioning schema, and should implement capabilities to report the version, so it can be known who is really deploying what. NB: A software element (e.g. FTS, WMS) is composed of one or more software components (i.e. RPMs)
  - Repository: There is a single production repository populated and maintained by EGI.ord MU
    - Product teams contribute tested & certified releases into a 'beta' repository
    - Product teams may have their own repository – but out of band of EGI.org
    - EGI.org moves component from beta to production repository once verified
- Provided by the Rollout Sites within the production infrastructure
  - Staged roll out process: few pre-selected sites involved first
  - A pre-defined list of few production sites per component, fulfilling a set of criteria, and with an SLA to regulate this task
    - Proper metrics are needed to make sure that these sites fulfill their commitment (SLA)
- Process
  - A release is announced to the rollout sites by the MU.
  - As defined by their SLAs sites are expected to report on failures of 'their' components within the SLA specified time period
    - If no issues filed within SLA period the release is 'good' for wider deployment
  - Staged roll out is not a compulsory waiting time: sites can skip the waiting time and proceed before, under their own risk
  - Staged roll out is transparent for the product teams, for them, the component is released once it is in the 'beta' repository
  - Note: This is in contrast with the present EGEE PPS, where components are 'released' onto the PPS but are not yet in production
- If a problem is found at a production site with a release:
  - Problem is reported through standard channel
  - No difference made between different stages of roll out (first sites deploying or any site, these should be transparent to mw support units)
  - EGI.org MU/OU/UCS management decides on its criticality and priority
  - If critical, it is the responsibility of the MU to engage with the relevant software provider to provide the means to fix the problem asap, keeping the production repository in a consistent state (e.g. producing asap a new version of the component that restores the previous functionality which did not have the critical problem).
  - It is not the responsibility of the production sites to be resilient to failures in this process (e.g. to expecting them to roll back, or to maintain local repositories), but responsibility of the process itself (this should be backup solutions that a site can try on its own will; the real solution on this case is to keep the repository always consistent)