GGUS for CIC operations

Piotr Nyczyk

Outline

- **Ticketing system for Grid operations**
- **Savannah to GGUS transition**
- **Critical functionality**
- **Use cases/scenarios:**
- **Possible future requirements:**
  - Knowledge DB specific requirements
  - Links between ticketing system and knowledge DB

Ticketing system for Grid Operations

- **Different model than User Support**
- **Currently Savannah - task manager**
- **Used by CICs and ROCs**
- **The processes are still evolving:**
  - Precise requirements are not so well established yet
  - But! Some of them are currently identified
- **General guidelines:**
  - Operations and User Support are two separate systems/models
  - Internal usage of the system for operations: CIC team submits tickets and manages them (+ ROCs)
  - Ticketing system as much manageable by CIC Team as possible
  - Integration with monitoring tools – application level access to the information on the tickets

Savannah to GGUS transition

- **Very limited time scale of *egee* Project:**
  - We must start as soon as possible even with limited functionality
  - Intensive interaction between CIC Team and GGUS team during the design and development (iterations)
  - But! At the same time even basic system should be operational
- **How to start? - Basic ticketing system that satisfies critical requirements:**
  - Extensions which are currently implemented in Savannah
  - Some reports and stats + application level access
- **Evolution to full blown system parallel to grid operations**
- **New ideas in Operations/changes in processes -> new requirements for the system**

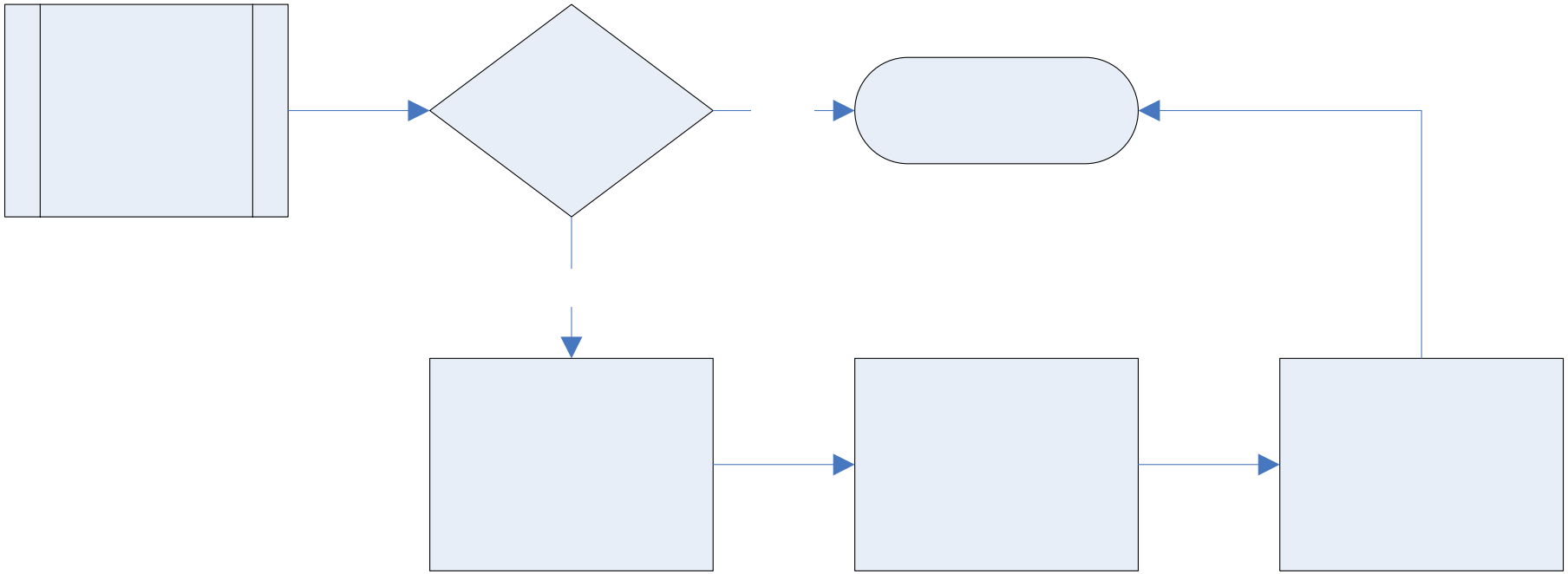
Critical requirements

- **CIC must be able to submit the tickets and assign them to specific team/group in one step**
- **Additional attributes in tickets:**
  - Category - site name
  - Priority
  - Group item - problem type
  - Action taken - last escalation level
  - Should be finished on - deadline
  - Person contacted
  - Response
- **CIC must be able to customize easily categories, problem types and escalation levels**
- **Application level queries:**
  - All tickets for given site name (ticket ID/reference, esc. level, deadline)
  - History of given ticket

Use cases/scenarios

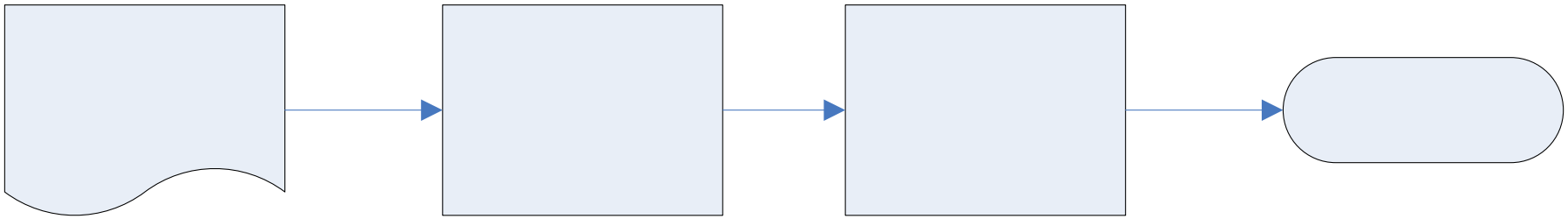
- Problem detection/diagnosis/report
- Problem resolution – communication chain site admin – ROC – CIC (severity escalation)
- Problem escalation
- Task closure – quarantine/full closure
- Stats/reports generation (metrics)

Scenario 1 – problem detection



- Checking if ticket is assigned is half-manual step (need for understanding if this is really the same problem)
- Diagnosis should result in reference to knowledge DB entry or an initial entry for a new type of problem
- Ticket submission requires information about the site to assign priority and deadline (number of CPUs)

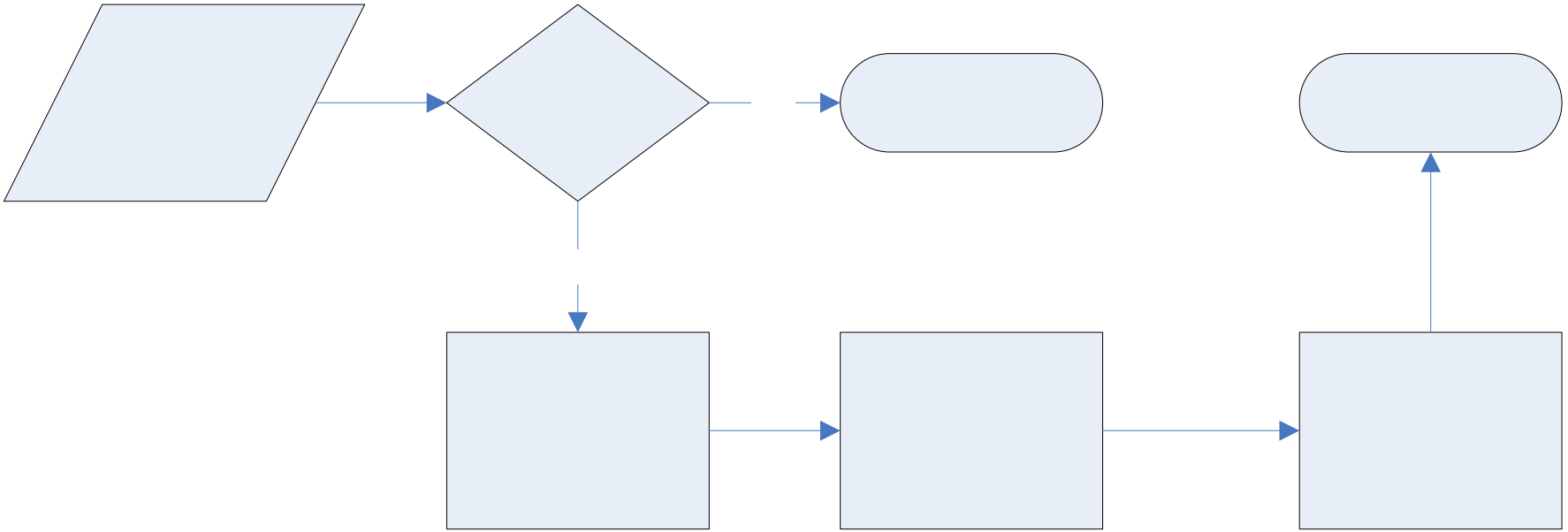
Scenario 2 - communication



- **Mail from site admin/ROC can also involve external actions like: test job resubmission, changes in the monitoring tools, etc. Some of them can be out of scope of the ticketing system for operations.**

Mail from site admin/ROC or

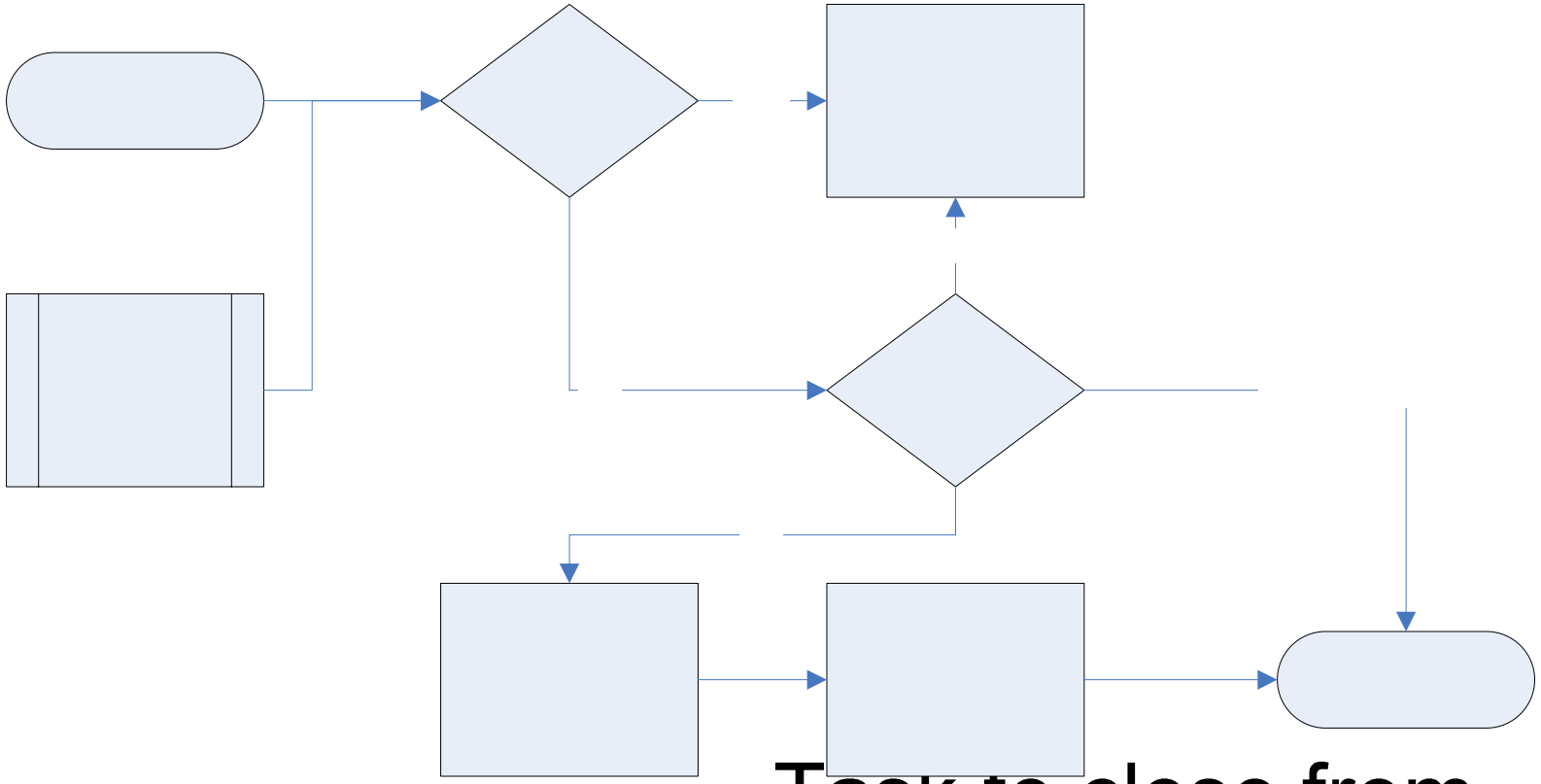
Scenario 3 – problem escalation



- **Checking if problem is still there involves manual analysis of monitoring tools**

Outdated task

Scenario 4 – task closure/quarantine



- **Trivial problem (GIIS died etc.) should be closed without quarantine – list of trivial problem not well defined yet.**

Scenario 5 – stats/reports

- **By region per week:**
  - Number of opened tickets
  - Number of closed tickets
  - Average/min/max time from open to close
  - Average/min/max escalation level just before closure
  - Average/min/max time before escalation level changes
  - Average/min/max time for response
- **Overall statistics per week (see above)**
- **Others….**