DLS Change Management for OPS/MVS

Automating the Automation

Freddy Sonnemans
Managing Director
DLS Systems bvba
Belgium
DLS Change Management for OPS/MVS

- **Personalia:**
  - 1972 – 1976: Continental Bank (Dev Assembler/PM)
  - 1976 – 1990: ABN Bank Belgium (IT Mgr/Sysprog/PM)
  - 1997 - : DLS Systems bvba

  » CA Partner on Automation & System Mgt products
DLS Change Management for OPS/MVS

- **Unicenter CA-OPS/MVS Event Management and Automation**
  - CA’s Automation tool for zOS
  - Based on Rexx Rules/Rexx Procedures
  - Rules Stored in RuleSets (= PDS)
  - Rexx Procedures stored in PDS
  - Rules executing synchronously (in requesting AS)
  - Procedures executing in TSO-like servers
## DLS Change Management for OPS/MVS

<table>
<thead>
<tr>
<th>CMfOPS</th>
<th>Reason for Development?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Experience/Frustration in the field</td>
</tr>
<tr>
<td></td>
<td>Problems:</td>
</tr>
<tr>
<td></td>
<td>- Do more with less (people)</td>
</tr>
<tr>
<td></td>
<td>- Growing number of Ipars</td>
</tr>
<tr>
<td></td>
<td>- Manageability of Ipars</td>
</tr>
</tbody>
</table>
DLS Change Management for OPS/MVS

- **Reason for Development?**

- **Results:**
  - An extremely powerful tool becomes dangerous
  - Tools that should help become a burden
  - Automation is abandoned
DLS Change Management for OPS/MVS

- **CMfOPS**

- **Choice of language for project:**
  - Rexx vs Assembler
  - Parent product: primarily Rexx
  - TSO/ISPF Environment
  - OBVIOUS CHOICE
DLS Change Management for OPS/MVS

- **Concepts:**
  - Strict separation of development and production environments
    - TESTRULE vs RULES libraries
    - TESTSUBF vs SUBF libraries
    - TESTREXX vs REXX libraries
  - Enforce standards
    - Only rules should be in Rulesets
    - Subroutine/Functions should be in separate libraries
    - Rexx procedures should be in REXX libraries
  - All rules/procedures should work everywhere
DLS Change Management for OPS/MVS

- **Concepts:**
  - Assured delivery to all lpars in scope
    - Delivery is independent of availability of lpar (IPL, Shutdown, Lost connection)
    - Delay (Freeze/Unfreeze) mechanism for production lpars
  - Control activation of rules across lpars
DLS Change Management for OPS/MVS

- **Concepts:**

- Deployment of Rules:
  - Disable Rule
  - Backup existing rule before overwrite
  - Replace rule by new version
  - Enable rule
  - Set AutoEnable
DLS Change Management for OPS/MVS

- **Concepts:**

- Deployment of Rexx procedures:
  - Backup existing procedure before overwrite
  - Compile the procedure (if set)
DLS Change Management for OPS/MVS

- **Concepts**:

- **Deployment of Subroutine/Function**:
  - Backup existing subroutine before overwrite
  - Refresh(Disable/Enable) ALL rules containing this subroutine
  - Recompile ALL procedures containing this subroutine
DLS Change Management for OPS/MVS

- **Concepts:**

- Extensive logging/queries:
  - Detailed logging of all functions executed on every lpar
  - Query/Set for all delayed deployments
  - Integrity checking between central pool of libraries and remote OPS libraries
DLS Change Management for OPS/MVS

Goals: All OPS/MVS rules/procedures should be applicable everywhere

- All rules/procedures have now one version that:
  - Applies to all existing environments
    - PROD(EB,EG,EN)
    - CONT(EB,EG,EN)
    - CLON(EB,EG,EN)
    - KSYS
    - DEVL
    - SYST
    - TEST
  - Will be activated dynamically to new lpars, according to their type
DLS Change Management for OPS/MVS

- QUESTIONS ???????

- Demo