Cisco
Developer Days
Automation

The bridge to possible

Use a Robot for Network Automation

(Getting more value from your NSO Test Automation)

NSO Developer Days 2022

Scott Barvick
CTO North America, Data Ductus
May 12, 2022

Data Ductus in brief

- Founded 1989 in Stockholm, Sweden
- 300+ employees
- Offices and projects in
  - EMEA
  - US
  - APAC
- More than 50+ NSO projects!
- Long history with BU and NSO community

Today’s Presentation

In the context of NSO projects…

- Motivation for automated testing
- Re-using your automated test framework to accomplish network automation tasks
- NOT RPA – robotic process automation. That can be a different session.

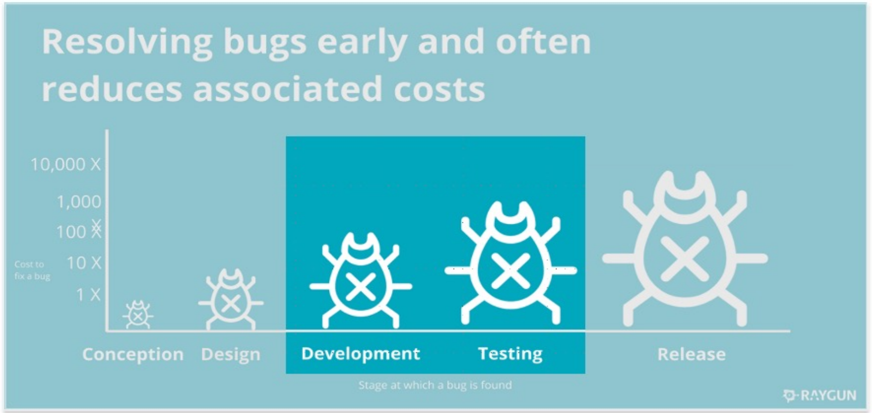
Test Automation Frameworks

Several commonly used for NSO projects:

- Robot Test Framework
- Pytest
- Junit
- Selenium
- Lux

Why Test Automation (for NSO Projects)?

NSO automation is software and …



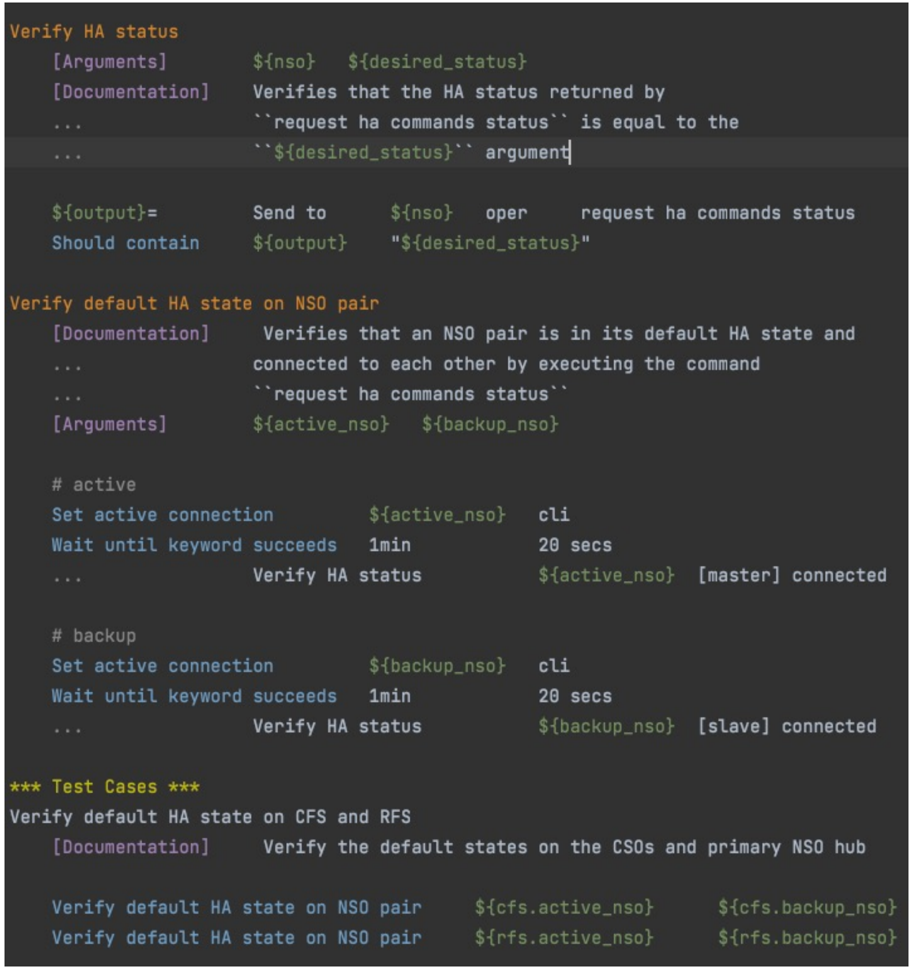
Source: https://raygun.com/blog/cost-of-software-errors/

It is also not just about bugs – changes in functionality need to be identified too

Automated Testing is Good !

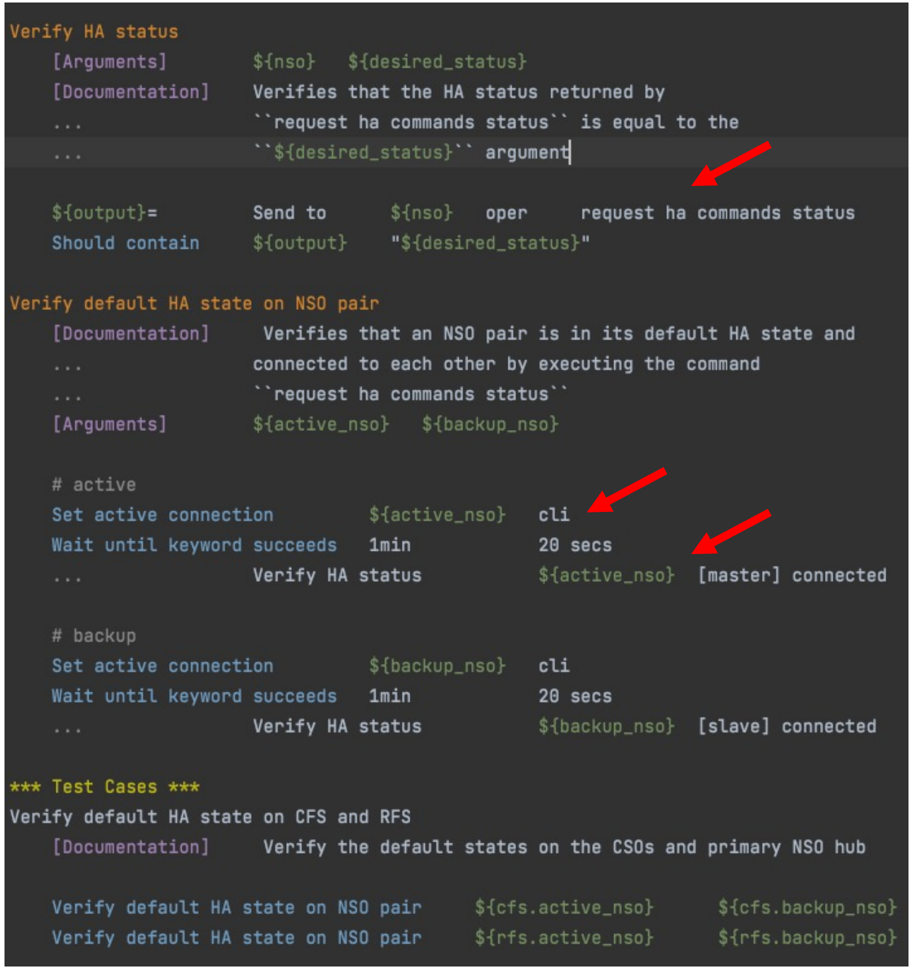
Now what about using it for for more than just testing?

Start with an Example: Verify HA status



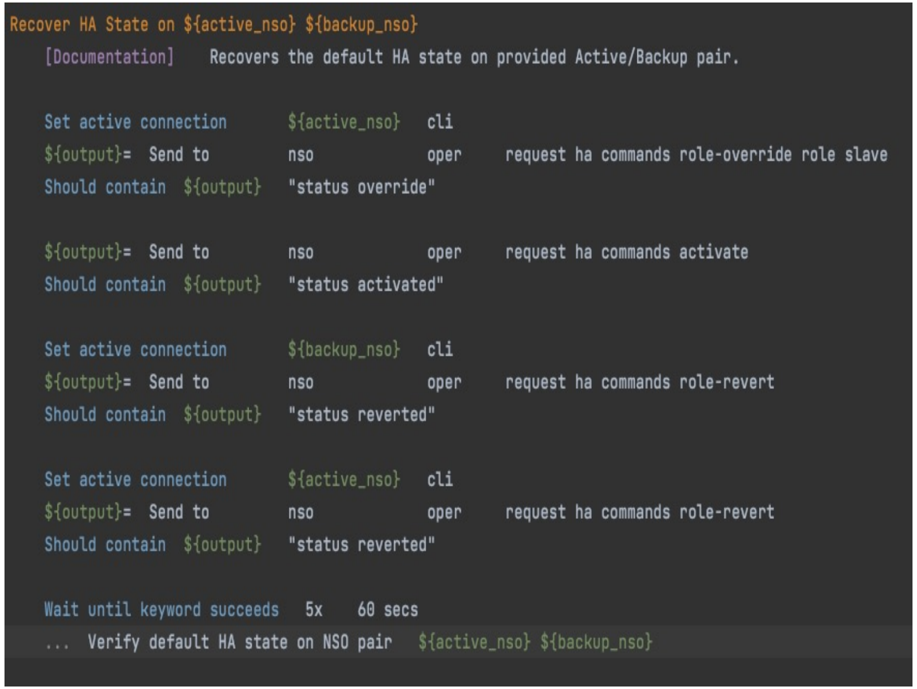
```
Verify HA status
    [Arguments]         ${nso}    ${desired_status}
    [Documentation]     Verifies that the HA status returned by
    ...                 ``request ha commands status`` is equal to the
    ...                 ``${desired_status}`` argument

    ${output}=          Send to      ${nso}    oper     request ha commands status
    Should contain      ${output}    "${desired_status}"

Verify default HA state on NSO pair
    [Documentation]      Verifies that an NSO pair is in its default HA state and
    ...                 connected to each other by executing the command
    ...                 ``request ha commands status``
    [Arguments]         ${active_nso}    ${backup_nso}

    # active
    Set active connection          ${active_nso}    cli
    Wait until keyword succeeds    1min             20 secs
    ...                 Verify HA status            ${active_nso}    [master] connected

    # backup
    Set active connection          ${backup_nso}    cli
    Wait until keyword succeeds    1min             20 secs
    ...                 Verify HA status            ${backup_nso}    [slave] connected

*** Test Cases ***
Verify default HA state on CFS and RFS
    [Documentation]      Verify the default states on the CSOs and primary NSO hub

    Verify default HA state on NSO pair     ${cfs.active_nso}       ${cfs.backup_nso}
    Verify default HA state on NSO pair     ${rfs.active_nso}       ${rfs.backup_nso}
```

Key Infrastructure



```
Verify HA status
    [Arguments]         ${nso}    ${desired_status}
    [Documentation]     Verifies that the HA status returned by
    ...                 ``request ha commands status`` is equal to the
    ...                 ``${desired_status}`` argument

    ${output}=          Send to      ${nso}    oper     request ha commands status
    Should contain      ${output}    "${desired_status}"

Verify default HA state on NSO pair
    [Documentation]      Verifies that an NSO pair is in its default HA state and
    ...                 connected to each other by executing the command
    ...                 ``request ha commands status``
    [Arguments]         ${active_nso}   ${backup_nso}

    # active
    Set active connection           ${active_nso}   cli
    Wait until keyword succeeds     1min            20 secs
    ...                 Verify HA status            ${active_nso}   [master] connected

    # backup
    Set active connection           ${backup_nso}   cli
    Wait until keyword succeeds     1min            20 secs
    ...                 Verify HA status            ${backup_nso}   [slave] connected

*** Test Cases ***
Verify default HA state on CFS and RFS
    [Documentation]      Verify the default states on the CSOs and primary NSO hub

    Verify default HA state on NSO pair     ${cfs.active_nso}       ${cfs.backup_nso}
    Verify default HA state on NSO pair     ${rfs.active_nso}       ${rfs.backup_nso}
```

Reuse the test infrastructure for automation



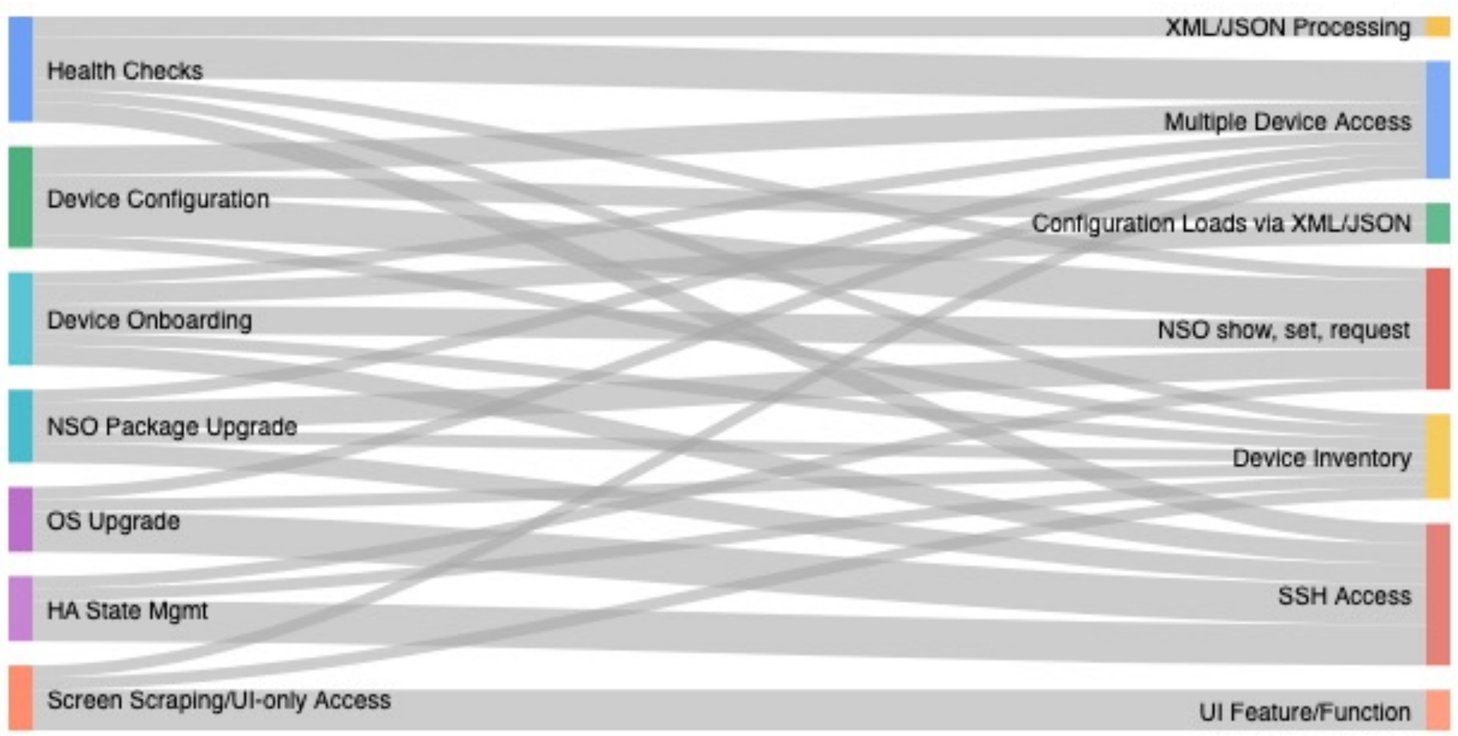
```
Recover HA State on ${active_nso} ${backup_nso}
    [Documentation]    Recovers the default HA state on provided Active/Backup pair.

    Set active connection      ${active_nso}    cli
    ${output}=  Send to         nso              oper     request ha commands role-override role slave
    Should contain  ${output}   "status override"

    ${output}=  Send to         nso              oper     request ha commands activate
    Should contain  ${output}   "status activated"

    Set active connection      ${backup_nso}    cli
    ${output}=  Send to         nso              oper     request ha commands role-revert
    Should contain  ${output}   "status reverted"

    Set active connection      ${active_nso}    cli
    ${output}=  Send to         nso              oper     request ha commands role-revert
    Should contain  ${output}   "status reverted"

    Wait until keyword succeeds   5x    60 secs
    ...  Verify default HA state on NSO pair   ${active_nso} ${backup_nso}
```

Could we do even more?

Components of a test automation implementation

- Multiple device access and connection types (e.g. IOS, IOSXR, Juniper, OpenStack, … with REST, RESTCONF, CLI, Netconf, …
- Device inventory collection and access
- Configuration loads with individual parameters or XML/JSON
- NSO access for show, config, request…
- Bash SSH access
- XML/JSON data processing
- UI feature/function encapsulation

Network Automations from test framework functions



Requirements for a Network Automation Platform

Category	COTS WF Platform	Open Source WF Platform	QA Robot Framework
Device Connectivity	✓	DIY	DIY
Configuration	✓	DIY	DIY
Ease of simple automations	✓	✓	DIY
Ease of complex automations	✓	✓	✗
REST API	✓	✓	DIY
Results DB	✓	✓	DIY

Summary

- Test automation for your NSO projects is critical for ongoing success
- The investment in that test automation can have additional benefits
    - Testbed automation natural and should be undertaken first
    - UI automation possible
    - Actual use for network maintenance is possible

The bridge to possible