Automation and integration – using ExLibris APIs

01 June 2018

STATE LIBRARY®
NEW SOUTH WALES

World leading library:
Collect
Connect
Community

- The Systems & Applications team at the State Library of New South Wales are automating build, test and deploy processes and integrating systems using ExLibris APIs.

- Utilising agile methodologies, we have achieved continual improvement across the library systems portfolio. This presentation will detail the tools, processes and techniques we've employed to enable continual improvement.

STATE LIBRARY®
NEW SOUTH WALES

World leading library:
Collect
Connect
Community

Agile

- Approach to software development

- Requirements and solutions evolve, not defined upfront

- Cross-functional, self-organising teams

- Continual improvement process: improvement to software, process and communication

Agile values

- Individuals and Interactions over processes and tools

- Working Software over comprehensive documentation

- Customer Collaboration over contract negotiation

- Responding to Change over following a plan

STATE LIBRARY®
NEW SOUTH WALES

World leading library:
Collect
Connect
Community

Some agile principles

- Satisfy customer through early and continuous delivery

- Welcome changing requirements

- Business people and developers work together

- Face-to-face communication

- Frequently Reflect on how to become more effective

- Tune and adjust

- Working software is the primary measure of progress

STATE LIBRARY®
NEW SOUTH WALES

World leading library:
Collect
Connect
Community

Agile at SLNSW – Challenges

- Risk-averse culture

- Existing project management framework is document heavy

- Traditional waterfall development fits in more easily with

    –Hierarchical structure of organisation

    –Current processes and practices

- Changing mindset. A different way of approaching problems

CI/CD

- Continuous Integration: Merging feature branches back into main branch frequently. Requires test automation.

- Continuous Delivery: Automated release process. Single click deploy to production environment.

- Continuous Deployment: Automated deployment to production environment. No human interaction.

STATE LIBRARY®
NEW SOUTH WALES

World leading library:
Collect
Connect
Community

CI/CD - automated testing

- Unit testing: Testing units of source code – function, class, method. Fine-grained.

- System testing: Testing integrated system. Often through the user interface. Course-grained.

CI/CD - Docker

- Docker is a light weight containerisation technology

- Allows us to standardise application dependencies across environments

- Helps make dev environment setup easy

- Helps make deployment easy

STATE LIBRARY®
NEW SOUTH WALES

World leading library:
Collect
Connect
Community

CI/CD benefits

- Less bugs make it into production

- Integrating code and building release is easy

- Less time manually testing

- Deploying release is easy

- Release often and quickly

- Quicker feedback loop for end users

STATE LIBRARY®
NEW SOUTH WALES

World leading library:
Collect
Connect
Community

Agile + CI/CD success story - PanDA

- Preservation and Digital Access

- Mass preparation and ingestion of digital assets into Rosetta

- Following SCRUM framework

- Unit testing

- Uses Alma and Rosetta APIs

STATE LIBRARY®
NEW SOUTH WALES

World leading library:
Collect
Connect
Community

PanDA – Alma API

- Using REST Bib API

- Retrieving bibliographic metadata

- For Rosetta Submission Information Package

- For embedding into image XMP Dublin Core

- Using Python Requests package

STATE LIBRARY®
NEW SOUTH WALES

World leading library:
Collect
Connect
Community

PanDA – Rosetta API

- Using SOAP API

- Deposit web services to start deposit

- SIP processing web services to track ingestion

- IE Update web services to validate ingestion

STATE LIBRARY®
NEW SOUTH WALES

World leading library:
Collect
Connect
Community

PanDA – unit testing

```python
def test_004(self):
    """ Running Test: IE has non readable directories/files """
    ie_name = 'ie_004'
    ie_path = self.create_ie_file(name=ie_name, files={
        'REP': self.create_file(1, '[name].tif'),
    }, create_data=True, create_metadata=False, create_bagit=False)
    file_path = self.create_text_file('garbage_001.txt', '', ie_path)
    files.change_fs_mod(fs_path=file_path, all_read=False)
    container_runner = self.create_ie_prepare(ie_name=ie_name, ie_path=ie_path)
    container_runner.run()
    self.assert_last_ex(code='block.app.ie_prepare.ex_01', message='Exception is not raised')
```

STATE LIBRARY®
NEW SOUTH WALES

World leading library:
Collect
Connect
Community

PanDA – CI/CD

- Bamboo CI/CD server

- Build triggered from github commit

  – Pull from repo

  – Run all unit tests

  – Create artefacts

- Deploy triggered by click

  – SCP deploy artefacts to server

  – Execute bash script on server

Alma acceptance tests

- SaaS product, no access to source code

- Scheduled monthly releases

- Start by recording business users performing tasks in Alma

- Document each task step by step

- Automate acceptance tests using Python Robot Framework

File Edit View Navigate Code Refactor Run Tools VCS Window Help

RobotScripts ▸ Framework ▸ Acquisitions ▸ Manual_Purchase_Order_Importing_Bib.robot

Project
- RobotScripts C:\Jameer\RobotScripts
  - Framework
    - Acquisitions
      - Results
      - AcqusitionsLocators.robot
      - AcqusitionsVariables.robot
      - Legal_Deposits_Claims_Submission_Creation.robot
      - Legal_Deposits_Vendor_And_Order_Management.robot
      - Manual_Invoice_Creation.robot
      - Manual_Purchase_Order_Importing_Bib.robot
      - Manual_Purchase_Order_With_New_Bib.robot
      - Non_Mono_Order.robot
      - Subscription_Series_Order_And_Receiving.robot
      - Vendor_Details_Review_On_Order.robot
      - Vendor_Management.robot
    - CommonKeywords.robot
    - CommonLocators.robot
    - CommonVariables.robot
    - CustomLibrary.py
    - PythonModules.py
  - .gitattributes
- External Libraries
- Scratches and Consoles

```robot
create_po_line_and_type_using_template_for_imported_record
    [Tags]    reg acq
    [Setup]    Run Keywords    launch_alma_session    choose_library_location

    #Search existing PO order, if found then delete it
    select_required_titles_for_search    arg2=${chose_first_titles_order_line}
    enter_value_into_element          ${top_menu_nav_search_text}    ${title_exist_in_alma_new}
    enter_view_expect_element    ${top_menu_nav_search_button}    ${PO_lines_tiitle_page_after_search}
    ${status}=    Run Keyword And Return Status    Page Should Contain    No records were found.
    Run Keyword If    '${status}' == 'False'    delete_added_purchase_order_line    arg1=${title_exist_in_alma_new}

    #Search existing bib record, if found then delete it
    search_earlier_created_record    arg1=${title_exist_in_alma_new}
    ${status}=    Run Keyword And Return Status    Page Should Contain    No records were found.
    Run Keyword If    '${status}' == 'False'    delete_created_bib_record

    visit_sub_menu_in_main_menu
    select_required_frame    ${frame_MD_editor}
    enter_view_expect_element    ${md_editor_search_catalog_profile_text}    ${md_editor_chose_profile_library_australia}
    enter_view_expect_element    ${md_editor_chose_profile_library_australia}    ${locator_catalog_profile_search_isbn_text}

    enter_value_into_element    ${locator_catalog_profile_search_isbn_text}    ${isbn_exist_in_alma_new}
    enter_view_expect_element    ${locator_catalog_profile_search_button}    ${locator_succesfull_search_refine_search_label}

    run keyword and continue on failure    Page Should Contain    ${isbn_exist_in_alma_new}
    run keyword and continue on failure    Page Should Contain    ${title_exist_in_alma_new}

    #Search and import existing record
    select_required_frame    ${frame_MD_editor}
    #Sleep    1s
    enter_view_expect_element    ${search_then_import_existing_record}    ${md_editor_element_bibliographic}
    click_then_wait_for_duration    arg1=${bib_record_PO_cart_button}
    ${status}=    Run Keyword And Return Status    Page Should Contain Element    ${delete_record_button_no}
    Run Keyword If    '${status}' == 'True'    Click Element    ${delete_record_button_no}
    Sleep    2s
    wait_for_req_element    ${PO_line_owner_type_label}
    run keyword and continue on failure    Page Should Contain    ${po_line_owner_type_view_tittle}

    #Chose POL type and template
    select_purchase_order_line_type
    wait_for_req_element    ${PO_line_owner_assign_inventory_manually}
    click_then_wait_for_duration    ${PO_line_owner_assign_inventory_manually}
    wait_for_req_element    ${create_PO_line_create_button}

    enter_view_expect_element    ${create_PO_line_create_button}    ${confirm_po_line_confirm_button}
    element_matches_required_text    ${confirm_po_line_confirmation_label}    ${po_line_owner_type_creation_confirmation_label}
    run keyword and continue on failure    page should contain    ${po_line_owner_type_creation_confirmation_message}
    enter_view_expect_element    ${confirm_po_line_confirm_button}    ${pucrhase_order_line_details_page}
    wait_for_req_element    ${created_PO_line_tittle}
    run keyword and continue on failure    page should contain    ${title_exist_in_alma_new}
    element_matches_required_text    ${pol_notification_bar}    ${pol_view_pol_creation_message}

    #Add library location
    Sleep    1s
```

58 chars   74:33   CRLF   UTF-8   Git: master

References

- https://en.wikipedia.org/wiki/Agile_software_development

- http://agilemanifesto.org/

- https://www.atlassian.com/continuous-delivery/ci-vs-ci-vs-cd