9.2 @ UQ:
How do you test for success?

The University of Queensland

Presenters:

Alex Rohan

Russell Lobo

Agenda

- Overview

- Challenges @ UQ

- Testing - Approach and Implementation

- Types of tools @ UQ

- Automation Tools selection Criteria

- Automation @ UQ

- Load testing

- Work In Progress

- What's next?

Overview

- SI-net is The University of Queensland's Campus Solutions product

- Systems testing at UQ was conducted manually, involving maintenance of testing records stored as Word and Excel files

- When the decision was taken to upgrade SI-net to PeopleSoft 9.2 and Fluid in 2017, it was also decided to move to an automated testing approach to reduce the testing turnaround time, and also assist with system monitoring

- Different automated testing tools were considered and implemented

- During the 9.2 upgrade testing process, legacy testing documents were translated to automated scripts

- Opportunities were identified to make further use of automated testing in other scenarios
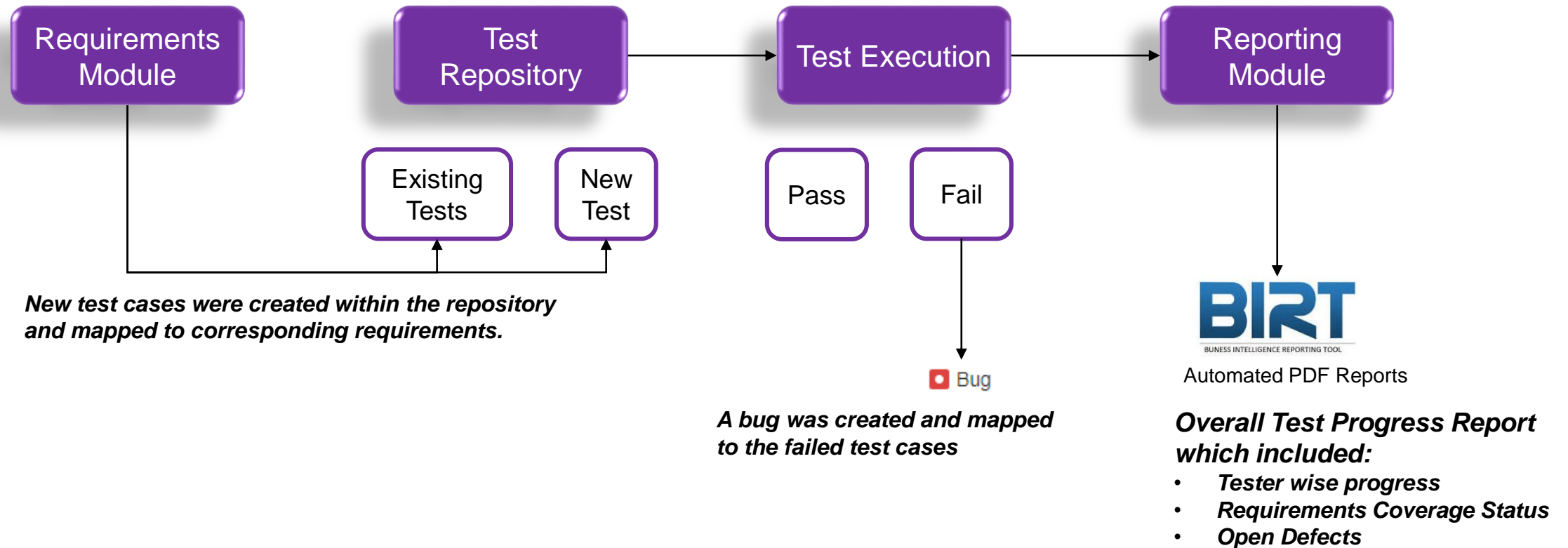
Challenges @ UQ

➢ Manual system testing was the norm – issues with

- resourcing

- consistency

- cost

➢ Undertaking an upgrade involved gathering together the testing artefacts

➢ Heavy reliance on the administrative compliance of previous testing team

➢ Environment issues were usually only identified when reported by users

➢ Heavy dependence on availability of particular SMEs to undertake testing

➢ Testing was undertaken by BAU team, so a balance needed to be struck between testing and BAU responsibilities
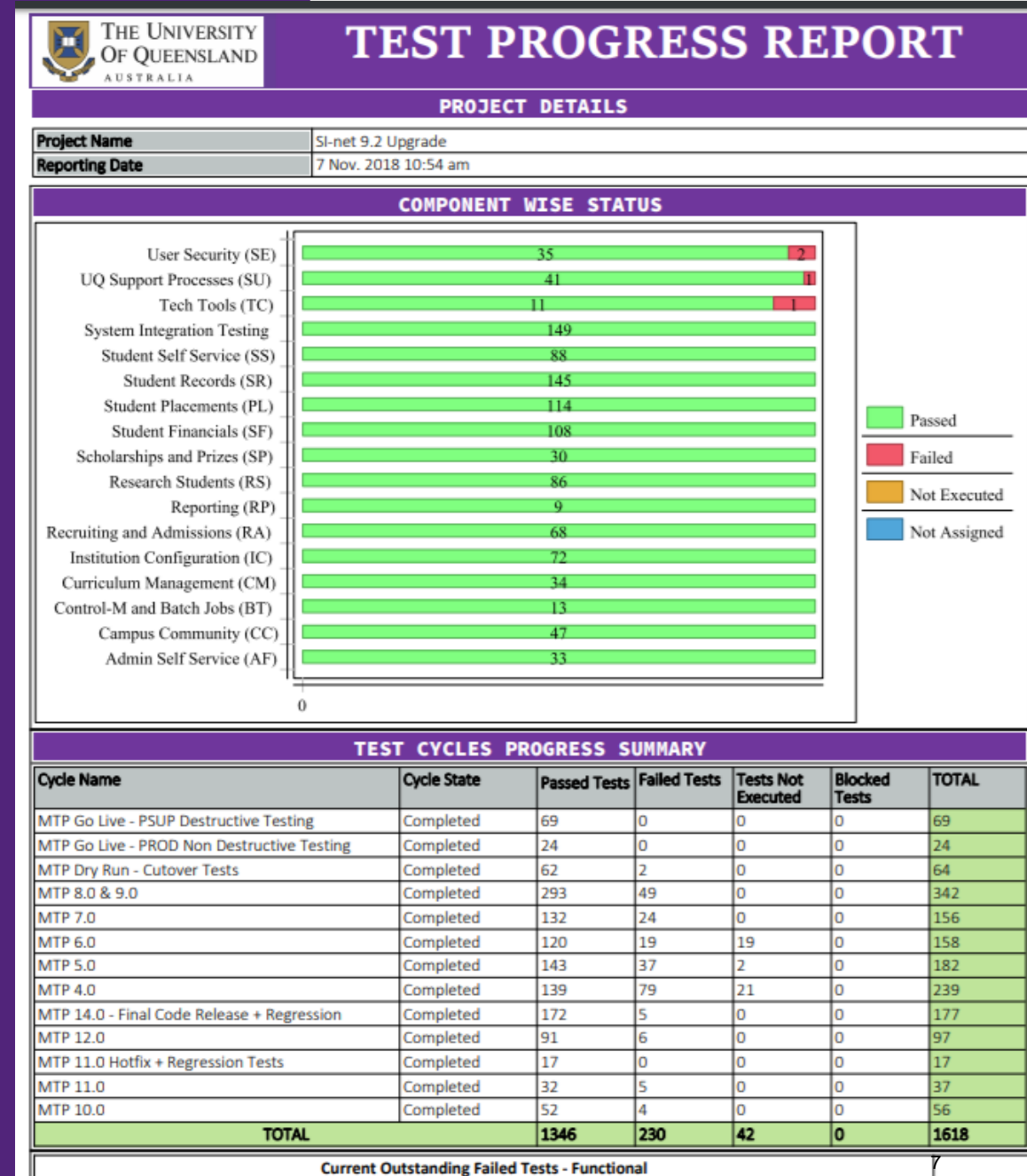
Testing Approach

❑ Clearly defined Test Management and Defect Management Processes

❑ Established Test Lab Setup

❑ Established Tools Portfolio

❑ Created Test Automation Framework

❑ Targeted critical scenarios for automation

❑ Defined UAT Framework

❑ Enabled Reporting and Traceability

Test Management Process

```
Requirements Module  →  Test Repository  →  Test Execution  →  Reporting Module
```

Requirements Module

Test Repository

Test Execution

Reporting Module

Existing Tests

New Test

Pass

Fail

New test cases were created within the repository and mapped to corresponding requirements.

🔲 Bug

A bug was created and mapped to the failed test cases

BIRT
BUNESS INTELLIGENCE REPORTING TOOL

Automated PDF Reports

Overall Test Progress Report which included:
- *Tester wise progress*
- *Requirements Coverage Status*
- *Open Defects*

Reporting

- Custom reports were created using a BI Reporting Tool

- Automated email report was sent to the wider team to track overall testing progress

- Progress reports helped plan testing activities for upcoming sprints

- Reports helped in obtaining signoffs and were a key deliverable from a test governance perspective

THE UNIVERSITY OF QUEENSLAND AUSTRALIA — **TEST PROGRESS REPORT**

PROJECT DETAILS

Project Name	SI-net 9.2 Upgrade
Reporting Date	7 Nov. 2018 10:54 am

COMPONENT WISE STATUS

Component	Passed
User Security (SE)	35 (2 Failed)
UQ Support Processes (SU)	41 (1)
Tech Tools (TC)	11 (1)
System Integration Testing	149
Student Self Service (SS)	88
Student Records (SR)	145
Student Placements (PL)	114
Student Financials (SF)	108
Scholarships and Prizes (SP)	30
Research Students (RS)	86
Reporting (RP)	9
Recruiting and Admissions (RA)	68
Institution Configuration (IC)	72
Curriculum Management (CM)	34
Control-M and Batch Jobs (BT)	13
Campus Community (CC)	47
Admin Self Service (AF)	33

Legend: Passed (green), Failed (red), Not Executed (orange), Not Assigned (blue)

TEST CYCLES PROGRESS SUMMARY

Cycle Name	Cycle State	Passed Tests	Failed Tests	Tests Not Executed	Blocked Tests	TOTAL
MTP Go Live - PSUP Destructive Testing	Completed	69	0	0	0	69
MTP Go Live - PROD Non Destructive Testing	Completed	24	0	0	0	24
MTP Dry Run - Cutover Tests	Completed	62	2	0	0	64
MTP 8.0 & 9.0	Completed	293	49	0	0	342
MTP 7.0	Completed	132	24	0	0	156
MTP 6.0	Completed	120	19	19	0	158
MTP 5.0	Completed	143	37	2	0	182
MTP 4.0	Completed	139	79	21	0	239
MTP 14.0 - Final Code Release + Regression	Completed	172	5	0	0	177
MTP 12.0	Completed	91	6	0	0	97
MTP 11.0 Hotfix + Regression Tests	Completed	17	0	0	0	17
MTP 11.0	Completed	32	5	0	0	37
MTP 10.0	Completed	52	4	0	0	56
TOTAL		**1346**	**230**	**42**	**0**	**1618**

Current Outstanding Failed Tests - Functional

7

Tool Types @ UQ

Test Management Tools

- Centralized test repository
- Maintain requirements and defects traceability with test cases
- Test Execution
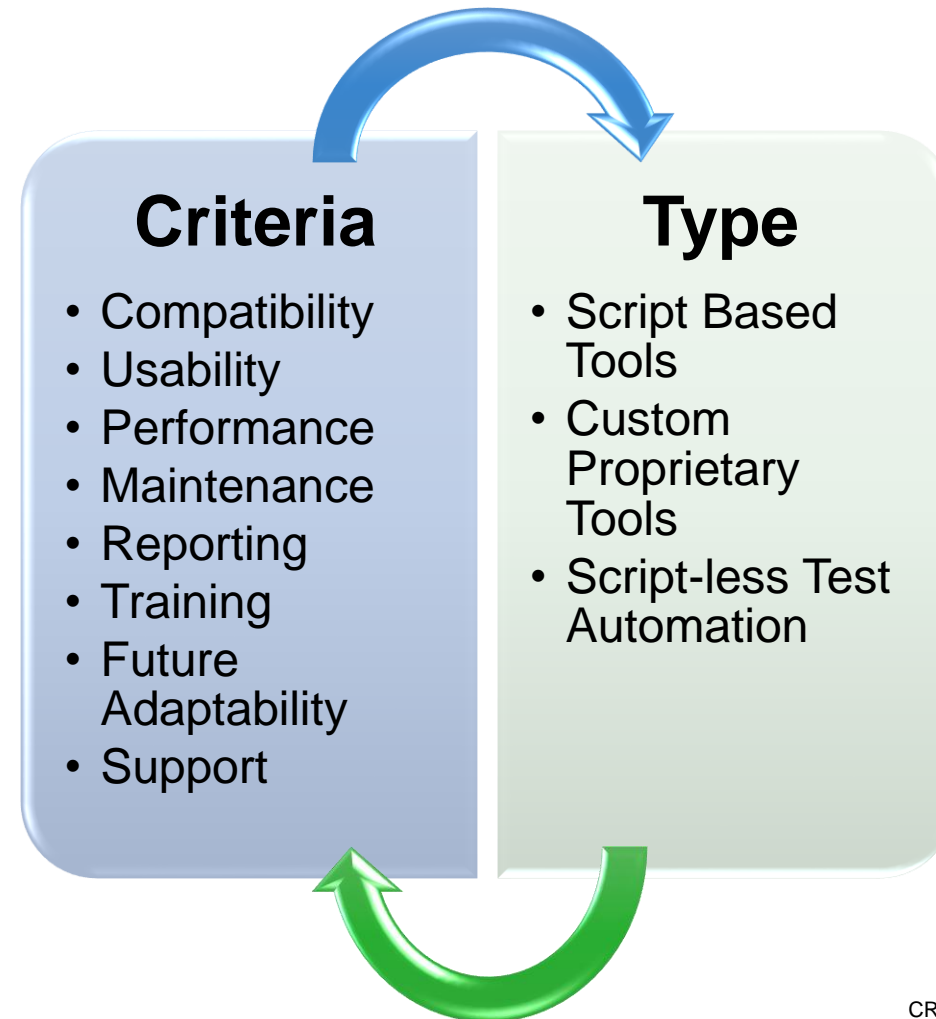- Enable reporting and generate Test Metrics

Test Automation Tools

- Automate test cases for application under test
- Reduce manual effort
- Reduce turnaround time
- Enables high volume based testing
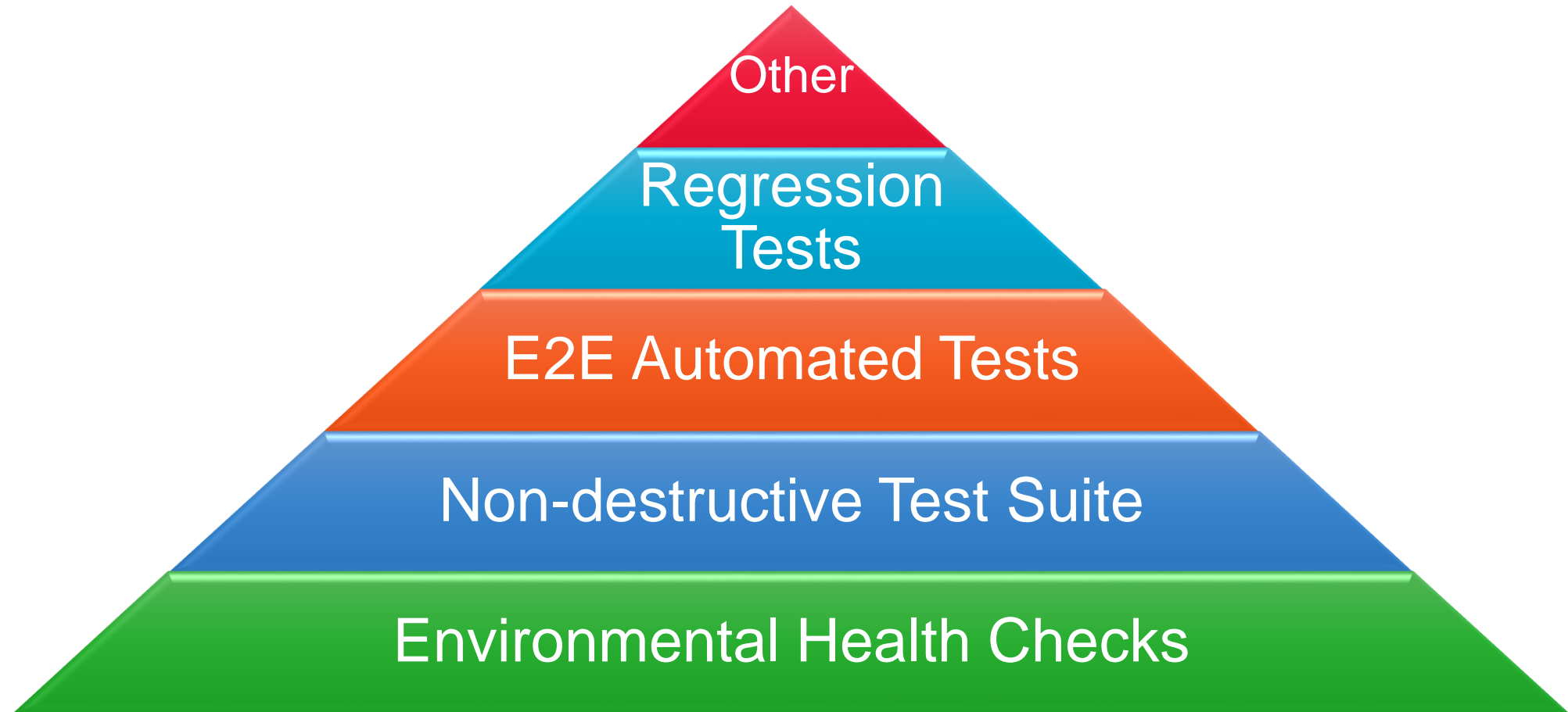
Continuous Integration Tools

- Create builds to run automated test scripts
- Automate deployment
- Enable monitoring of automated tests
- Enable trend analysis of certain functions/features/applications

Automation - Tools Selection Criteria

Criteria

- Compatibility
- Usability
- Performance
- Maintenance
- Reporting
- Training
- Future Adaptability
- Support

Type

- Script Based Tools
- Custom Proprietary Tools
- Script-less Test Automation

Automation @ UQ

A pyramid diagram with the following levels from top to bottom:
- Other
- Regression Tests
- E2E Automated Tests
- Non-destructive Test Suite
- Environmental Health Checks

Environmental Health Checks

Problem Statement

- Unstable test environments

- Broken integrations for application

- Large turnaround times for resolution

- Manual configuration management

Solution

- Identified common cause of failures along with operations team

- Established list of checkpoints – functional tests & config related

- Built automated environment health check scripts

- Embed the health check scripts as part of Continuous Integration / Monitoring Solutions run in Docker containers

- Incorporated health check scripts as part of environment refresh processes

URL alive=PASS
Login=PASS
Web Profile=PASS
Report Manager=PASS
Int Gateway=PASS
Int Nodes=PASS
Elastic Search=PASS
SSL=PASS
Logout=PASS
Process Scheduler=PASS
PSNT=🔴 PSUNX=✅
DB Last Refreshed: 12-Oct-18 06:10

#2952   7 minutes ago
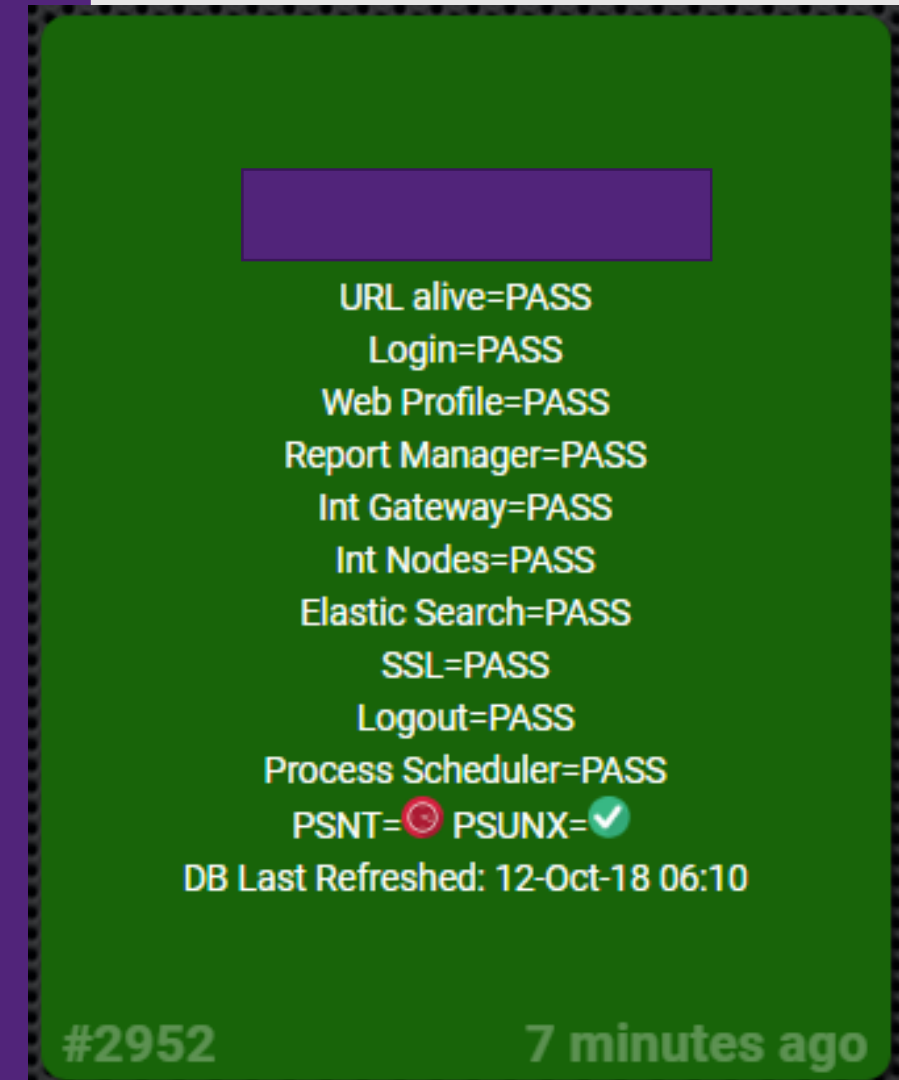
Functional Tests

Problem Statement

- ERP patching strategy impacted due to large test effort – introduces security risks

- Test data creation laborious process

- No quick testing suite available for validating functional modules across the application

Solution

- Established list of Non-Destructive tests. Scenario coverage includes all functional areas of the application

- Non-destructive tests run against prod and non-prod environment

- Tests do not modify underlying DB transactional data – can be run against prod

- Architected test scenario design with SME's. Created E2E flow scenarios that cover all business critical functions

- Automated all E2E scenarios thereby enabling quick regression testing turnaround times
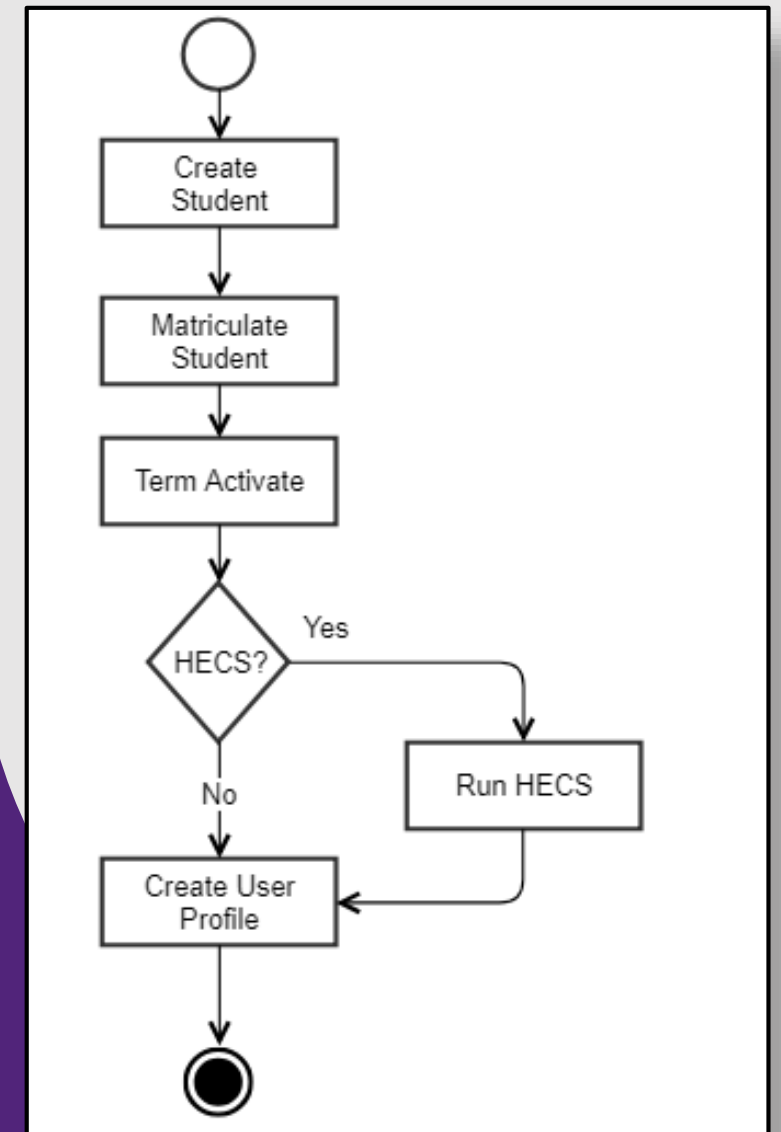
Performance Testing

Problem Statement

- Version upgrade and changes in infrastructure could have impact on performance

- General user feedback prior to version upgrade related to performance of certain functions

Solution

- Established reusable load tests using a load testing tool. Scenario coverage included the most used functions of the application (e.g. Course Enrolment, View Grades)

- Enabled a production like environment for execution of load test

- Conducted load, soak and capacity tests to check the performance of the system under various load patterns

- Used metrics to analyse areas of improvement

Scenario 1: Add Course -> Signon -> Drop Course

Legend: Baseline Test, Progression Test
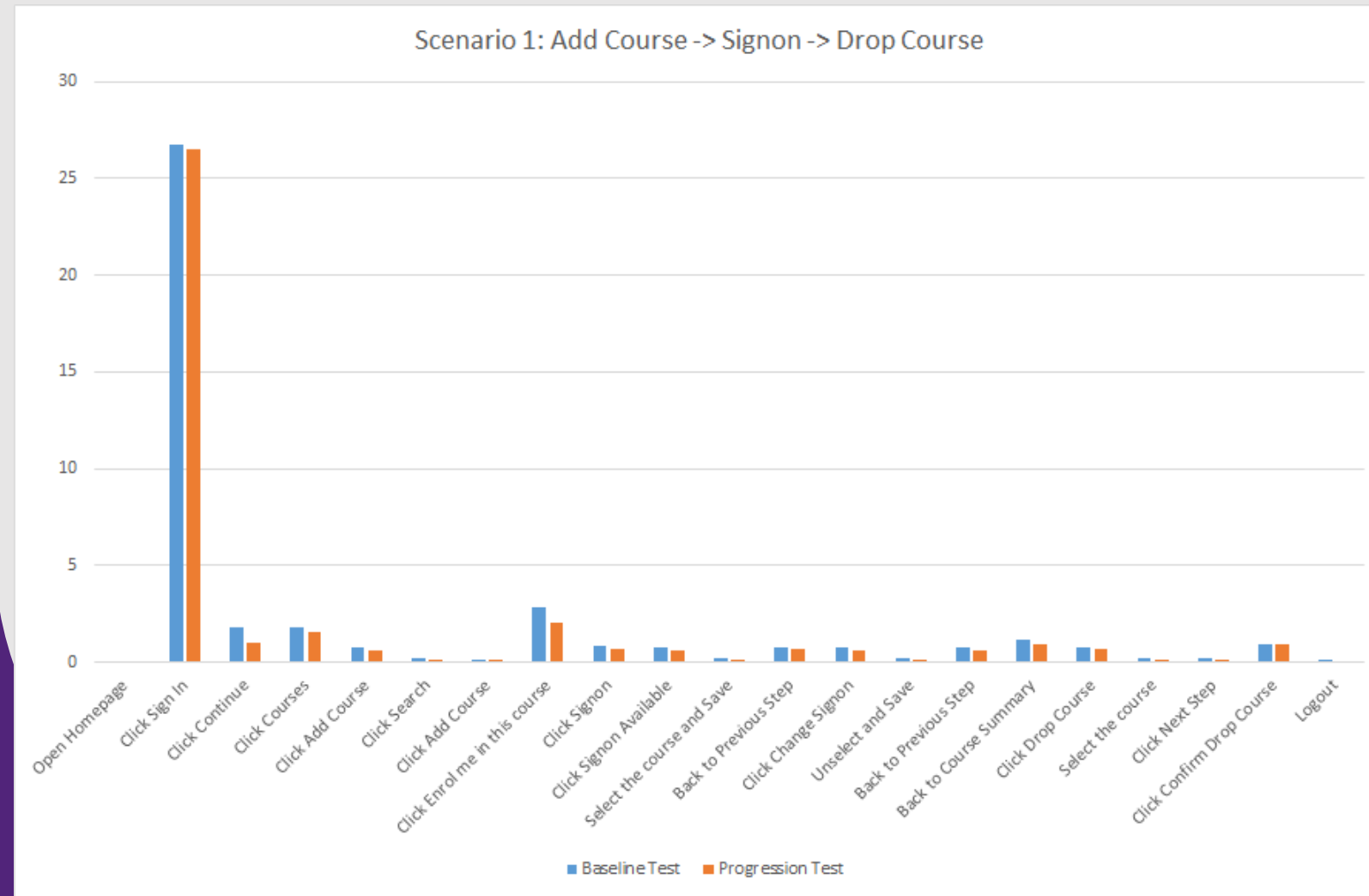
Benefits

✓ Improved the efficiency of testing

✓ Enable cross-skilling through Centralized Test Repository

✓ Lesser turnaround time specially during shakedown tests and fixing environment configuration
  e.g. 8-minute non-destructive suite vs. 30-minute manual non-destructive shakedown testing

✓ Improved quality of testing

✓ Testing was more streamlined with introduction of processes

✓ Reporting Module helped the project management in terms of planning and obtaining sign-off from various stakeholders

✓ Performance tests ensured the performance of the most frequently used features were unaffected by the version upgrade

What's In Progress?

Incorporating Test Management Process for BAU

- Introduction of standard testing processes with use of a test management tool to orchestrate BAU testing activities

- It would ensure BAU Testing is more streamlined

- Progressively building test assets for future use

Risk-based testing

- Carry out risk analysis during the test scoping phase

- Features that are used most likely and has the most impact on failure is tested most thoroughly whereas least likely used features with minor impact can be de-scoped

- Enables efficient use of resources and helps stick to deadlines and budget

What's next?

Expanding Automated tests to assist BAU

- Test cases for automation will be picked depending on certain selection criteria.

- Incorporate Business Process Automation to ensure E2E scenarios can be automated and thus reduce testing time

- Helps expanding the regression suite that can be run after an environment is available after a migration

Make Functional Test Suites scalable to conduct Non Functional Tests

- Management of multiple test suites could be challenging sometimes.

- Efficient use of test resources.

- Helps in establishing an optimized tools portfolio.

Transition towards DevOps

- Define automation frameworks to effective create test scripts from reusable functions

- Enable environment management, test management and test data management

Questions?

Thank you

Russell Lobo | Test Lead
Information Technology Services
r.lobo@uq.edu.au
07 3365 7160

Alex Rohan | Application Technical Analyst
Enterprise Support Systems, ITS
a.rohan@uq.edu.au
07 3365 7106

facebook.com/uniofqld

instagram.com/uniofqld

linkedin.com/in/russell-lobo-58967167

facebook.com/uniofqld

instagram.com/uniofqld

linkedin.com/in/alexrohan