P&A | Operational Acceptance Testing

DL	@ Vivek Upreti @ chris.wraith @ anubhav khanna (Unlicensed) @ indra.sharma @ Andrew James (Unlicensed) @ Neeraj Verma @ Nikhil Dass @ vivek.singh1 @ Cathrine McGrann @ Poulin @ Rasamy Sirisomphone (Unlicensed) @ Anand Patel (Unlicensed) @ sruthy.pramod @ rishabh.govindraj @ shivani.gupta9 @ Mohit Mair @ Saurabh Sameer Saim

- Detailed OAT Scenarios - Vault
- Detailed OAT Scenarios - Microservices
- Detailed OAT Scenarios - PostgreSQL
- Detailed OAT Scenarios - Hashicorp
- Detailed OAT Scenarios - Kafka
- Detailed OAT Scenarios - DataFlow
- Detailed OAT Scenarios - TM Workflows
- Detailed OAT Scenarios - JBPM Workflows
- Detailed OAT Scenarios - PubSub
- Detailed OAT Scenarios - Spanner (SOR, SOE, SOI)
- NFR Implementation | Vault
- OAT Steps - Documentation template
- OAT / DR Test Reference
- Detailed OAT Scenarios - Multi Region
- Multi Region - BLD02 Environment Region 2 test report
- Detailed OAT Scenarios - DataFlow Multi Region

==========

OPERATIONAL ACCEPTANCE TEST

SCOPE, APPROACH & SCENARIOS

1. Operational Documentation

In-Scope: ✅ ❌

Operational Documentation		Vault	Microservices	PostgreSQL	Hashicorp	TM Workflows	JBPM Workflows	Spanner	PubSub	DataFlow
Definition / Objective	All Documents that are necessary to operate the system. (e.g. architecture overviews, design documentation, operational docs) are identified, and they are checked for completeness, availability and accessibility to the relevant people. Transition, Standard and Crisis mode are addressed.	✅	✅	✅	✅	✅	✅	✅	✅	✅
Input	All prerequisites for implementation. Few examples are: • » IBR (Infrastructure Blueprints) • » Implementation Plan • » Installation Manual and Application Production Manual • » Architecture Overview • » Business Continuity/Disaster recovery plans									

Test Approach & Scenario	The task of the OAT tester is to ensure that these documents are available finalized and in sufficient quality to enable smooth operation. The documents must be handed over and accepted by the teams handling the production support, helpdesk, disaster recovery, and business continuity. The handover must be documented. Operation teams must be included as early as possible to obtain their valuable input about the documentation required for operating the system. This way, transparency about the completeness of documents is achieved early on in the development process, and there will be sufficient time left for counter measures if documents are missing or lacking in quality.									
Delivery Model	To be encorporated at workstream level									
Risk of Not Executing	If the operational documentation review is omitted or performed too late, testing will start without the final documentation available which reduces the effectiveness of the tests. As a result, an increased number of issues may be raised, causing delays in the timeline. For operation teams, not having the correct documentation can affect their ability to maintain, support and recover the systems.									

2. Rehearsal testing

In-Scope: ✅ ❌

Rehearsal testing		Vault	Microservices	Postgre SQL	Hashicorp	TM Workflows	JBPM Workflows	Spanner	PubSub	DataFlow
Definition / Objective	Unlike User Acceptance testing / E2E which integrates business functions. OAT integrates all functions and stakeholders of a production system. A rehearsal or staging environment testing is necessary for bigger changes in production environments, especially when they concern many stakeholders. Objective is to avoid the risks of failures in the process chain and longer system downtimes shall be minimized or avoided.	✅	✅	✅	✅	✅	✅	✅	✅	✅
Input	All prerequisites to implementation. Few examples are: • » Fully tested software with the test release recommendation implementation plan • » Installation Manual and Application Production Manual • » Business Continuity / Disaster recovery plans.									
Test Approach & Scenario	• » The main fundamentals in this phase are to ascertain that the test cases of functional change when executed ensure business continuity. • » 2 sets are needed, Roll-forward and Roll-back scenarios. • » The test team in close conglomeration with operations has to execute the relevant test cases that adhere to various factors based on the project being implemented. • » In case of Business release, new functionality introduced is the main focus. • » In case of configuration release, all major modules including technical behavior of the system needs to be scrutinized.									
Delivery Model	To be encorporated at workstream level									
Risk of Not Executing	Implementation itself is at risk Business continuity at risk IT adherence of SLA is at risk Major leakage of incidents is also possible.									

3. Installation Testing

In-Scope: ✅ ❌

Installation Testing	Vault	Microservices	Postgre SQL	Hashicorp	TM Workflows	JBPM Workflows	Spanner	PubSub	DataFlow

Definition/ Objective	This test activity ensures that the application installs and uninstalls correctly on all intended platforms and configurations. Objective is to ensure correctness, completeness and successful integration into system management functionality for the following : Installation, De-installation, Fallback, Upgrade, Patch.	✅	✅	✅	✅	✅	✅	✅	✅	✅
Input	Target systems with different configurations (e.g. created from a base image), as well as the specification of changes to the operating system (e. g. registry entries), and a list of additional packages that need to be installed.									
Test Approach & Scenario	The following aspects must be checked to ensure correct installation and de-installation: • » The specified registry entries and the number of files available need to be verified after the installation is finished. • » The application must use disk space as specified in the documentation in order to avoid problems with insufficient space on the hard disk. • » Ifapplicable,the installation over old(er) version(s) must be tested as well the installer must correctly detect and remove or update old resources. • » The occurrence of installation breaks has to be tested in each installer step, as well as breaks due to other system events (e.g. network failure, insufficient resources). The application and the operating system must be in a defined and consistent state after every installation break possible. • » Shared resources may have to be installed or updated during installation, and while these processes are performed, conflicts with other applications must be avoided. In terms of uninstallation, the system should be cleaned from shared resources that are not used anymore. This will result in higher performance and increased security for the whole system.									
Delivery Model	In colloboration with Platform and Workstream engineers									
Risk of Not Executing	May result in conflicts with other applications Compromise or even broken systems Low user acceptance even-though the software itself has been tested successfully and works as designed. Number of manual effort, cost and conflicts in larger systems									

4. Platform Upgrade Testing

In-Scope: ✅ ❌

Platform Upgrade Testing		Vault	Microservices	PostgreSQL	Hashicorp	TM Workflows	JBPM Workflows	Spanner	PubSub	DataFlow
Definition / Objective	This type of testing comprises test activities that ensure successful exchange or upgrade of central components like run time environments, database systems or standard software versions. Objective is to obtain proof of correct functionality, sufficient performance or fulfillment or other quality requirements.	✅	✅	✅	✅	✅	✅	✅	✅	✅
Input	IBR (Infrastructure Blue prints) Functionality in chain that needs regression testing									

Test Approach & Scenario	2 possible approaches for introducing central components: • » The first approach would be to set up central components as productive within the development system, i.e. central components would move parallel to the application software along the test stages towards a release date according to a common release plan. Testing would start implicitly with developer tests. • » The second approach would be to test changing a central component in a production-like maintenance landscape. In this case, a dedicated regression test would be performed parallel to production. Central components would be released for both operation and development. • » Based on the approach the steps would be : 1. Deriving relevant applications from impact analysis 2. Selecting regression tests on the basis of risk assessment 3. Performing regression tests (including job processing) a. Parallel to development b. In a dedicated maintenance environment 4. Deciding on acceptance or rejection									
Delivery Model	By executing cross-workstream integration tests on old and new platform									
Risk of Not Executing	Incompatible software platform System downtimes Non-functional issues affecting business continuity Missing fallbacks Data defects Very crucial for Multi-vendor environment and cloud computing									

5. SLA / OLA Monitoring Testing

In-Scope: ✅ ❌

SLA / OLA Monitoring Testing		Vault	Microservices	Postgre SQL	Hashicorp	TM Workflows	JBPM Workflows	Spanner	PubSub	DataFlow
Definition / Objective	This test type examines the implemented monitoring functionality in order to measure the service and operation level. Ob- jective is to derive if the monitoring functionality is complete, correct and operable in order to derive the right service and operation level.	✅	✅	✅	✅	✅	✅	✅	✅	✅
Input	Business Continuity Checks KPI parameters Thresholds and warning requirements									
Test Approach & Scenario	Select relevant SLA/OLA Deriving Monitoring scenarios to estimate service levels Integration scenarios into test scenarios of other test types Execute tests and calculating service levels from monitoring									
Delivery Model	Workstream and programme level logging and monitoring									

6. Backup & Restore Testing

In-Scope: ✅ ❌

Backup & Restore Testing	Vault	Microservices	Postgre SQL	Hashicorp	TM Workflows	JBPM Workflows	Spanner	PubSub	DataFlow

Definition / Objective	Backup and restore testing focuses on the quality of the implemented backup and restore strategy. In an expanded test execution, the test objective of a backup includes all the resources, ranging from hardware to software and documentation, people and processes.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Input	Working test environment and operation process									
Test Approach & Scenario	Backup and restore testing can be executed in a use-case scenario based on well-defined test data and test environments. In general, a test will comprise the following steps: • » Quantifying or setting up the initial well-defined testing artefacts • » Backing up existing testing artefacts • » Deleting the original artefacts • » Restoring artefacts from backup • » Comparing original artefacts with restored ones. • » If applicable, performing a roll-forward and checking again.									
Delivery Model	To be achieved within workstream with help of Thought Machines									
OAT Test Environment	If backup and restore functionality is available, testing can in principle be executed parallel to early functional testing. However, since the tests will involve planned downtimes or phases of exclusive usage of environments. Moreover, this activity will require the following: • » Representative test data • » Established backup infrastructure • » Established restore infrastructure.									
Risk of Not Executing	Risk of losing data in a restore situation impeding ability to perform disaster recovery Business continuity interruption SLA/OLA adherence hampered									

7. Failover Testing / Recovery Testing

In-Scope: ✓ ✗

Failover Testing / Recovery Testing		Vault	Microservices	PostgreSQL	Hashicorp	TM Workflows	JBPM Workflows	Spanner	PubSub	Data Flow
Definition / Objective	The objective of failover testing can be subdivided into two categories: • » The degree of the quality of fault recognition (technical measures have to be implemented to detect the failure event e.g. DataFlow pipeline issue) • » The efficiency and effectiveness of the automatic failover reaction in terms of reaction time and data loss.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Input	IBR (Infrastructure Blue prints) Installation Manual and Application Production Manual Architecture Overview/ Failover plans									
Test Approach & Scenario	The test case specification has to describe the measures taken to trigger the failure event. It is not necessary to execute events exactly as they happen in the real world since it can be sufficient to simulate them with technical equipment. For instance: • » Failure: Lost Network Connection • » Failure: Vault API/PostgreSQL Connection • » Failure: File system or cluster failure.									
Delivery Model	To be achieved in colloboration with platform and workstream engineer									

DATA-SPECIFIC

RESILIENCE

Scenario	Expected behavior and SLA
Datastore (BQ or Spanner) is not reachable	
Pub/Sub topic is not available	
Pub/Sub queue is not reachable	
Dataflow pipeline is not deployed/available	
It must be possible to run batch alongside real time transaction processing, while still maintaining service targets (for batch duration and real time performance).	
When any service becomes unavailable, this must be handled with appropriate user-friendly responses given to the user.	

==========**Glossary**==========

1. **Availability**: Implementation of any new features into the live production environment should not adversely affect the integrity of the current production services
2. **Availability**: Each component can be shutdown and start successfully within the agreed time scale.
3. **Availability:** Back-out of a failed change from the production environment will be successful and will not adversely affect existing services.
4. **Availability**: The system should automatically adjust itself to availability of system resources.
5. **Alerts & Monitoring**: All critical alerts must go to the ServiceNow and reference the correct resolution document.
6. **Alerts & Monitoring**: Alerts are in place and issued if agreed thresholds are exceeded
7. **Alerts & Monitoring**: Threshold Monitoring Alerts are in place and issued if agreed thresholds are exceeded. For e.g., disk utilization, CPU, memory, etc.
8. **Recovery**: Any recovery documentation produced or altered, including Service Diagrams, is valid. This should be handed over to the relevant support areas.
9. **Recovery**: Any component is affected by the failure, should show recommended order of restart, time to complete, etc.
10. **Recovery**: If failover is invoked, failback can be performed successfully, and recovery to the original state is achievable.
11. **Recovery**: If several components have been affected by a failure, there should be a proven plan showing the recommended order of restart, time to complete, etc.
12. **Recovery**: The system/component can be shut down and restarted cleanly, without service disruption, or within an agreed window of scheduled downtime.