

**Mobilink – OSS Expansion project**

**User Acceptance Testing**



**Innovise ESM**

**Keypoint House**

**High Street**

**Slough**

**SL1 1DY**

**Tel: +44 (0) 1753 513 902**

**Authors: Prev Datta**

**Title: Mobilink - UAT**

**Version: 1.6**

**Date: 6th August 2010**

Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Reason** | **Author** |
| 1.1 | 06/08/2010 | Created | Prev Datta |
| 1.2 | 13/08/2010 | Updated | Prev Datta |
| 1.3 | 17/08/2010 | SME’s Add Cases | Prev Datta |
| 1.6 | 20/08/2010 | SME’s Add Cases | Prev Datta |

Table of Contents

[1.1. Objectives 5](#_Toc270071181)

[1.2. Definitions 5](#_Toc270071182)

[1.3. Acceptance Testing 6](#_Toc270071183)

[1.4. Configuration Validation 6](#_Toc270071184)

[1.5. Sample Test Scenarios 8](#_Toc270071185)

[1.6. Defect Tracking Process 9](#_Toc270071186)

[1.7. Problem Priorities 9](#_Toc270071187)

[BSS Environmental Test 1.2 10](#_Toc270071188)

[BSS Environmental Test 1.3 11](#_Toc270071189)

[BSS Environmental Test 1.4 11](#_Toc270071190)

[BSS Environmental Test 1.5 12](#_Toc270071191)

[ITM Test 1.6 13](#_Toc270071192)

[ITM Test 1.7 14](#_Toc270071193)

[ITM Test 1.8 14](#_Toc270071194)

[ITM Test 1.9 15](#_Toc270071195)

[ITM Test 2.0 15](#_Toc270071196)

[ITM Test 2.1 16](#_Toc270071197)

[ITM Test 2.2 16](#_Toc270071198)

[ITM Test 2.3 17](#_Toc270071199)

[ITM Test 2.4 17](#_Toc270071200)

[Cell Down Test 3.0 18](#_Toc270071201)

[Test Event source: 18](#_Toc270071202)

[Cell down Test 3.1 19](#_Toc270071203)

[Cell down Test 3.2 20](#_Toc270071204)

[Change Management Test 4.0 21](#_Toc270071205)

[Change Management Test 4.0 22](#_Toc270071206)

[Change Management Test 4.1 22](#_Toc270071207)

[Change Management Test 4.2 23](#_Toc270071208)

[Change Management Test 4.3 24](#_Toc270071209)

[Change Management Test 4.4 24](#_Toc270071210)

[Change Management Test 4.5 25](#_Toc270071211)

[Change Management Test 4.6 25](#_Toc270071212)

[Incident Test 4.7 26](#_Toc270071213)

[Incident Test 4.8 28](#_Toc270071214)

Introduction

## Objectives

The purpose of this document is to test the functionality of the components that make up the Mobilink OSS solution encompassing IBM Tivoli Service Request Manager, OMNIbus Probes, IMPACT, Tivoli Monitoring, Network Manager and Business Service Manager. The objective of the acceptance process is to ensure the product and processes meet Mobilink’s expectations and documented requirements.

The intent of this document is to outline an acceptance process for Mobilink. It includes

* Description of the testing processes
* Describes a test case and defect tracking
* Details the acceptance criteria and test cases

## Definitions

* ***Problem/Defect –*** a system response that does not meet the predicted test result and/or design, and therefore was unexpected. Problems require a change to the system (i.e., screen, table, report, etc.) and are specific to the application (i.e., TSRM).
* ***Issue -*** is an unresolved problem that needs to be identified, prioritized and the impact determined.
* ***Tested/Validated*** - functionality that has successfully passed a minimum of one round of system testing and acceptance testing.
* ***System Test*** - test performed by the technical team to verify the code/module was built as designed.
* ***Acceptance Testing -*** test performed by the business team to ensure the built system (and in turn, design) meets the business needs.
* ***Configuration Validation –*** series of tests and checks to validate the data loads, reports, and screen configurations prior to moving into Production.
* ***End-to-end Testing*** – Acceptance testing using multiple scenarios to ensure the system completes the overall business process.

Acceptance Process 1

## Acceptance Testing

The Mobilink OSS project is essentially a package implementation (i.e. TSRM/IMPACT) configured to meet Mobilink’s business needs. As such the testing process used will be somewhat simpler in nature than that used in typical development projects. Testing will primarily focus on verifying that the IBM solution meets the business requirements, i.e. acceptance testing. To verify this, the users will follow a number of scenarios which covers how the daily business will be conducted using the business processes for event and operational areas. These scenarios will then be executed using the base line configuration of TSRM and all defects/problems reported and tracked. To be effective, the testing must be structured in that all scenarios must be documented in detail so that they are repeatable, the execution results are tracked, and all defects/problems are recorded and reviewed.

Once scenario testing is completed, End-to-End (EtoE) tests will be run. The EtoE tests are essentially combining the various specific scenarios together so that a complete day in the life is simulated.

## Configuration Validation

The complete configuration (screens, value lists, tables, reports) needs to be verified prior to signing off to move to staging (i.e., code freeze) and again before moving to production. To ensure this review is thorough, each major component of the configuration will be assigned to Mobilink to review. They will test the following:

**Tables**

* Sample approx. 10% of data (i.e. 15-25 for 200 record table) and include the first and last record
* Compare the samples to the source data.
* Ensure the data is the same as the source, accurate, and complete

**Value Lists**

* Ensure all values in the list are accurate and complete

**Screens**

* Ensure the screen is useable and functions as expected, specifically:
* All the field lengths are correct and display the complete value
* Tab order is correct
* All the fields are available

**Work Flows**

* Ensure your test email is a recipient of the Work Flow.
* Follow the steps described in the Test Case
* Check to see if you have received the notification via email and the contents is correct.

**Reports**

* Ensure all the reports are readable and accurate, specifically:
* Data is correct
* All rows and columns are aligned on the report

Test Case Documentation and Tracking

The test cases are the basis of the testing exercise and should be devised to verify the application implementation & configuration supports the business processes. Tests will be designed with a predetermined objective (i.e., acceptance criteria) of testing the functionality of a specific component of the process. The design of the test cases must reflect a scientific method (i.e. tests must have an objective, prediction of outcome, controlled environment, and repeatable results).

*Example 1:*

***Acceptance Criteria***

*Mobilink require manual creation of Work Orders in some cases for vendor or internal operations.*

***Predicted Result***

*The User is able to manually create Work Orders from the WO module and assign it to a team or group of people and associate related Incident Records to it.*

***Test Description***

*1. Log on*

*2. Go to the WO module and select create NEW*

*3. Fill in the Work Order, Name, Description etc*

*4. Go to Related Records TAB and Select NEW under the Related Tickets section*

*5. Select the arrow up and browse for the Ticket you wish to associate.*

*6. Select the Ticket and SAVE*

## Sample Test Scenarios

|  |  |  |
| --- | --- | --- |
| **Test Case ID:** AW-01 | **Test Type:** Customer Service Desk, Functional Test | |
| **Test Name:** Standard work request. | | |
| **Description:** Create a standard work request. | | |
| **Data Setup:** Cost centers, GL, Active Org and Site, User needs to be assigned a default site. | | |
| **Steps to Execute:**   1. Enter request for work via web, the request should include; location, description, contact, date/time, cost center and a customer requested priority. 2. Once the above information has been correctly entered click “OK” to submit the work request to the Customer Service Desk 3. Log in to Maximo with the generic Work Controller account and select the above entered work request from the queue 4. Complete filling out the work request with asset, location code, responsibility/assignment, work type and GL account information. 5. Assign a status of **Unapproved** to the work request and route the work request to the correct work group. | | |
| **Expected Outcome:**  Work Order created and routed to the correct work group. | | **Actual Outcome:** |
| **Test Outcome: PASS - FAIL** | | **Test Date:** |
| **Tester:** Prev Datta  **Tester Sign Off:** | | **Test Lead:** Prev Datta  **Test Lead Sign Off:** |

## Defect Tracking Process

All problems/defects need to be recorded and reviewed prior to making any configuration changes to ensure that the fix will not have a negative impact and that the test result was correct. The following process will be used to review all defects.

|  |
| --- |
|  |

## Problem Priorities

The person opening the issue should assign a priority. At anytime throughout the process, the original submitter and whomever the issue is assigned to may change the priority with mutual agreement.

The priorities are as follows:

* ***P1 - Critical -*** System wide failure. No one can complete any test cases/business processes.  Everything must be dropped until fixed.  Cannot go live with this defect.  Example:  Maximo database crash and no one can logon
* ***P2 - Major*** - Major failure that stops that test case/business process from proceeding and there are no work around, but other work can continue. Bumps all scheduled work until resolved. Cannot go live with this defect.  Example:  SQL error when trying to create a work order on WOTRACK
* ***P3- Minor*** –. Defect that has work around and does not prevent completion of the test case/business process.  Resolution can be scheduled.  Can go live with the defect but will require that a work around also be implemented.  Example: GL string cannot be picked from the value list but the correct one can be typed in.

***E - Enhancements -*** Functionality that was not in the original scope.  May or may not be required to go live.

Test Cases

## BSS Environmental Test 1.2

Please ensure the following pre-requisites are met before starting the following test cases.

* + All Event need to be
    - Domain = ‘BSS’
    - EventType = ‘EnvironmentalAlarm’
  + Event 1 - Temperature Alarm Node A
    - Node = ‘HYD498A\_\_CtyzenPlz\_2’
    - AddText = ‘BTS-EXTERNAL [9]External Temperature [89]’
  + Event 2 - Temperature Alarm Node B
    - Node = ‘HYD410A\_\_SiteArea\_2
    - AddText = ‘BTS-EXTERNAL [9]External Temperature [89]’
  + Event 3 - Voltage Alarm Node A
    - Node = ‘HYD498A\_\_CtyzenPlz\_2’
    - AddText = ‘BSC-EXTERNAL [50] Low Voltage [7]’
  + Event 4 - Voltage Alarm Node B
    - Node = ‘HYD410A\_\_SiteArea\_2
    - AddText = ‘BSC-EXTERNAL [50] Low Voltage [7]’

| Test Number: 1.2.1 | Revision: 1.0 | | Author: Chris Janes | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** BSS Environmental | | | **Standard/Customisation:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | BSS Environmental - Basic Functionality: | | | |
| Test Purpose | To test Basic Functionality of the BSS Environmental event Handling | | | |
| Procedure | 1. Insert the test event (Event 1) 2. Check the event appears in WebTop 3. Wait for Synthetic event ( x Min) 4. Check Synthetic event is generated 5. Check Event Journal of synthetic event of entry about the original event 6. Check SyntheticServerSerial and Synthetic ServerName fields are populated in the original event 7. Wait for 1 minute check TTNumber is populated in the synthetic event 8. Check Incident Record in TSRM 9. Wait for a further minute and check TTNumber is populated in the original event 10. Clear the original event 11. Check that the synthetic event is set to clear 12. Check that the Incident Record is set to resolved | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## BSS Environmental Test 1.3

| Test Number: 1.3.1 | Revision: 1.0 | | Author: Chris Janes | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** BSS Environmental | | | **Standard/Customisation:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | BSS Environmental - Multiple Nodes of the same type: | | | |
| Test Purpose | To test Multiple Nodes of the same type within BSS Environmental event handling | | | |
| Procedure | 1. Insert the test event (Event 1) 2. Check that synthetic event and Incident record are created 3. Insert second test event (Event 3) 4. Check that it’s SyntheticServerSerial, Synthetic ServerName and TTNumber are populated 5. Check journal of synthetic event has details of both source events 6. Clear Event 1 7. Check that the synthetic event is NOT cleared 8. Clear Event 3 9. Check that the synthetic event is set to clear 10. Check that the Incident Record is set to resolved | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## BSS Environmental Test 1.4

| Test Number: 1.4.1 | Revision: 1.0 | | Author: Chris Janes | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** BSS Environmental | | | **Standard/Customisation:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | BSS Environmental - Single Node of the multiple type: | | | |
| Test Purpose | To test Single Node of the multiple type within BSS Environmental event handling | | | |
| Procedure | 1. Insert the test event (Event 1) 2. Check that synthetic event and Incident record are created 3. Insert second test event (Event 2) 4. Check that an additionalsynthetic event and Incident record are created 5. Clear Event 1 6. Check that the synthetic event associate to Event 1 is not set to clear 7. Check that the Incident Record associate to Event 1 is not set to resolved 8. Clear Event 2 9. Check that the synthetic event associate to Event 2 is set to clear 10. Check that the Incident Record associate to Event 2 is set to resolved | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## BSS Environmental Test 1.5

| Test Number: 1.5.1 | Revision: 1.0 | | Author: Chris Janes | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** BSS Environmental | | | **Standard/Customisation:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | BSS Environmental - Multiple Node of the multiple type: | | | |
| Test Purpose | To Test Multiple Nodes of the multiple type within BSS Environmental event handling | | | |
| Procedure | 1. Insert the test event (Event 1) 2. Check that synthetic event and Incident record are created 3. Insert second test event (Event 2) 4. Check that an additional synthetic event and Incident record are created 5. Insert third test event (Event 3) 6. Check that it’s SyntheticServerSerial, Synthetic ServerName and TTNumber are populated 7. Check journal of synthetic event associated with Events 1 and 2 has details of both source events 8. Insert forth test event (Event 4) 9. Check that it’s SyntheticServerSerial, Synthetic ServerName and TTNumber are populated 10. Check journal of synthetic event associated with Events 3 and 4 has details of both source events 11. Clear Event 1 12. Check that the synthetic event associate to Events 1 and 3 is not set to clear 13. Check that the Incident Record associate to Events 1 and 3 is not set to resolved 14. Clear Event 2 15. Check that the synthetic event associate to Events 2 and 4 is not set to clear 16. Check that the Incident Record associate to Events 2 and 4 is not set to resolved 17. Clear Event 4 18. Check that the synthetic event associate to Events 2 and 4 is set to clear 19. Check that the Incident Record associate to Events 2 and 4 is set to resolved 20. Clear Event 3 21. Check that the synthetic event associate to Events 1 and 3 is set to clear 22. Check that the Incident Record associate to Events 1 and 3 is set to resolved | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## ITM Test 1.6

| Test Number: 1.6.1 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM Functionality | | | |
| Test Purpose | Require Windows server monitoring | | | |
| Procedure | From the ITM console, expand the windows computer systems and select a windows ITM agent.  Browse through the collected attributes for the server  Compare the attribute values with those manually collected from the server | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## ITM Test 1.7

| Test Number: 1.7.1 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM Functionality | | | |
| Test Purpose | Enable Poll on servers for availability | | | |
| Procedure | Simulate a server failure by simultaneously killing all ITM processes on a monitored server.  View the ITM event list to ensure that the relevant event is present  View the Omnibus event list to ensure that the relevant event is present | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## ITM Test 1.8

| Test Number: 1.8.1 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM Functionality | | | |
| Test Purpose | Enable Windows Service Monitoring | | | |
| Procedure | Simulate a windows service failure by manually stopping a service marked as automatic start  View the ITM event list to ensure that the relevant event is present  View the Omnibus event list to ensure that the relevant event is present | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## ITM Test 1.9

| Test Number: 1.9.1 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM Functionality | | | |
| Test Purpose | Enable Windows Service Monitoring | | | |
| Procedure | Simulate a CPU problem by lowering the monitored threshold in CPU situations to a lower value  Simulate a memory problem by lowering the monitored threshold for memory situations to a lower level  Simulate an error log problem by echoing some known problem text into a monitored log file  For each simulated problem, view the ITM and Omnibus event lists to ensure that the relevant event is present | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## ITM Test 2.0

| Test Number: 2.0.1 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM Functionality | | | |
| Test Purpose | Monitor Processor Application Information | | | |
| Procedure | Simulate application processes problems by adding a new process ‘test’ to each application process monitoring situation.  For each simulated problem, view the ITM and Omnibus event lists to ensure that the relevant event is present | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## ITM Test 2.1

| Test Number: 2.1.1 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITN Functionality | | | |
| Test Purpose | IN Domain Machines have OS Solaris 9 and Redhat Linux. Require OS Solaris 9 and Redhat Linux platform monitoring | | | |
| Procedure | Run the command tacmd listSystems on the ITM server to generate a list of all monitoring agents.  Compare the agent list with Mobilinks IN domain server list and ensure that all known IN servers have agents installed | | | |
| Results |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## ITM Test 2.2

| Test Number: 2.2.1 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM Functionality | | | |
| Test Purpose | VAS Domain Machines have OS OpenVMS, Linux, Solaris and HP-UX. Require OS OpenVMS, Linux, Solaris and HP-UX platform monitoring | | | |
| Procedure | Run the command tacmd listSystems on the ITM server to generate a list of all monitoring agents  Compare the agent list with Mobilinks VAS domain server list and ensure that all known VAS servers are being monitored | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## ITM Test 2.3

| Test Number: 2.3.1 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM Functionality | | | |
| Test Purpose | Log file Monitoring | | | |
| Procedure | For each monitored logfile configuration, echo a known error string into the logfile.  Ensure that an event is generated both within ITM and Omnibus | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## ITM Test 2.4

| Test Number: 2.4.1 | Revision: 1.0 | | Author: Dave Pridmore | Date: 17/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** ITM | | | **Standard/Customisation:** | |
| **Product:** ITM | | | **Requirement Catalogue reference:** | |
| Test Title: | ITM Functionality | | | |
| Test Purpose | In total there are 413 servers to monitor | | | |
| Procedure | Use the output of the tacmd listSystems command to count the number of servers being monitored by ITM. Compare this to the 413 value. | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## Cell Down Test 3.0

Please ensure the following pre-requisites are met before starting these test cases.

### Test Event source:

All cell outage events need to be

AlertGroup = ‘Cell Out of Service’

MainFlag = 1

ImpactFlag = 5 (Set by Multisite down policy)

We need cell outage events for particular site. Let’s assume, we have 5 cells with ‘Cell Out of Service’ for particular site i.e. ‘LHR860H\_FRF\_4(BSC08).

Event 1 - “LHR19822\_\_S\_KachaRahimRoad” outage alarm

Event 2 - “LHR29822\_\_S\_KachaRahimRoad” outage alarm

Event 3 - “LHR39822\_\_S\_KachaRahimRoad” outage alarm

Event 4 - “LHR49822\_\_S\_KachaRahimRoad” outage alarm

Event 5 - “LHR59822\_\_S\_KachaRahimRoad” outage alarm

Also, based on scenario, we need resolution event for these cells.

AlertGroup = ‘Cell Out of Service’

MainFlag = 1

ImpactFlag = 0

Event 6 - “LHR19822\_\_S\_KachaRahimRoad” resolution alarm

Event 7 - “LHR29822\_\_S\_KachaRahimRoad” resolution alarm

Event 8 - “LHR39822\_\_S\_KachaRahimRoad” resolution alarm

Event 9 - “LHR49822\_\_S\_KachaRahimRoad” resolution alarm

Event 10 - “LHR59822\_\_S\_KachaRahimRoad” resolution alarm

| Test Number: 3.0.1 | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Cell Down | | | **Standard/Customization:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | Cell outage alarms | | | |
| Test Purpose | When all cell outage alarming then policy should create synthetic event as “Site Down” and raise TT. Also, when all cell resolution alarms come then synthetic event should be clear and incident record set to resolved. | | | |
| Procedure | 1. Insert the test events (Event 1,Event2, Event3,Event4,Event5) 2. Check the events appears in WebTop 3. Wait for Synthetic event ( x Min) 4. Check Synthetic event is generated and it should be “Site Down” 5. Check Event Journal of synthetic event should have entries of cell outage events details. 6. Check SyntheticServerSerial and Synthetic ServerName fields are populated in the original cell outage events. 7. Also, check for any existing “Environmental Alarm” for this site, they should be cleared and their details added into journal entry of synthetic event. 8. Wait for 1 minute check TTNumber is populated in the synthetic event 9. Check Incident Record in TSRM 10. Wait for a further minute and check TTNumber is populated in the original cell outage events 11. Insert the resolution events for all cells(i.e. Event6, Event 7, Event8,Event9,Event 10) 12. Check that the synthetic event is set to clear 13. Check that the Incident Record is set to resolved | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## Cell down Test 3.1

| Test Number: 3.1.1 | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Cell Down | | | **Standard/Customization:** Custom | |
| **Product:** IMPACT | | | **Requirement Catalogue reference:** | |
| Test Title: | Partial site down | | | |
| Test Purpose | When cell of the site alarming then policy should raise synthetic event or update it based on number of cell outage alarms to “Site Partial Down” or “Site Down”. Also, based on cell resolution alarms, it should clear synthetic event or downgrade it to “Site Partial Down”. | | | |
| Procedure | 1. Insert the test event (Event 1,Event2, Event3) 2. Check the events appeared in WebTop 3. Check that synthetic event creates and should be “Site Partial Down”. 4. Its journal has entry of all three cell outage alarm details. 5. Incident record is created 6. Check SyntheticServerSerial & SyntheticServerName fields are populated in the original cell outage alarms. 7. Also, check for any existing “Environment Alarm” for this site, they should be cleared and their details should be added into journal of synthetic event. 8. Insert remaining cell outage event (Event4,Event5) 9. Check the synthetic event should update to “Site Down” and its journal has entry of newly cell outage alarms. 10. Check that SyntheticServerSerial & SyntheticServerName fields are populated in the original cell outage alarms. 11. Check journal of synthetic event has details of both source events 12. Wait for some time (x Min) 13. Insert resolution event for some cells (i.e. Event6, Event7,Event8). 14. Check that synthetic event should be downgraded to “Site Partial Down” and its journal has entry of resolution events. 15. Insert resolution event for remaining cell (i.e. Event9, Event10) 16. Check that synthetic event should be clear and its journal has entry of resolution events. 17. Check that the Incident Record is set to resolved | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## Cell down Test 3.2

| Test Number: 3.2.1 | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Cell Down | | | **Standard/Customization:** Custom | |
| **Product:** IMPACT | | | Requirement Catalogue reference | |
| Test Title: | Cell outage alarms resolve during sleep time | | | |
| Test Purpose | When cell outage alarms occur first time then based on sleep time, if alarm resolved during that period then does not raise TT. | | | |
| Procedure | 1. Insert the test event (Event 1,Event2,Event3) 2. During sleep time, insert resolution events for cell outage (Event6, Event7 & Event8). 3. Check that synthetic event should not be created as we don’t have any cell outage alarms. | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## Change Management Test 4.0

| Test Number: 4.0.1 | | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- | --- |
| **Test Category:** Change Management | | | | **Standard/Customization:** Custom | |
| **Product:** TSRM | | | | **Requirement Catalogue reference:** Custom | |
| Test Title: | **Create** - Create Change Request | | | | |
| Test Purpose |  | | | | |
| Procedure | Create a new Request for Change. This is done by creating a new Process Request, setting the Process Manager field to be 'Change', filling out the needed information and then click on the 'Submit' icon. Common important information to provide:   * Select classification (Change type). * Set priority. * Set target completion.   Set target CIs (if known). | | | | |
| **Results** |  | | | | Pass  Fail |
| **Reason for failure or comments** | | |  | | |

## Change Management Test 4.0

| Test Number: 4.0.1 | Revision: 1.0 | | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- | --- |
| **Test Category:** Change Management | | | | **Standard/Customization:** Custom | |
| **Product:** TSRM | | | | **Requirement Catalogue reference:** Custom | |
| Test Title: | | **Accept** Accept Change Manager | | | |
| Test Purpose | |  | | | |
| Procedure | | Quick review of the RFC. If all of the critical information is provided and the request is valid, add additional details needed to create an actual Change from this request and click on the 'Accept' icon. Common details that should be added during this phase:   * Assign owner to Process Request. * Accept Process Request. * Assign job plan to Change record. * Add additional approval tasks to Change, if needed. * Assign owners to tasks.   Update Change status to "In Progress" | | | |
| **Results** | |  | | | Pass  Fail |
| **Reason for failure or comments** | | |  | | |

## Change Management Test 4.1

| Test Number: 4.1.1 | Revision: 1.0 | | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- | --- |
| **Test Category:** Change Management | | | | **Standard/Customization:** Custom | |
| **Product:** TSRM | | | | **Requirement Catalogue reference:** Custom | |
| Test Title: | | Access - Change Owner and Change Analyst | | | |
| Test Purpose | |  | | | |
| Procedure | | Identify the impact and urgency assigned to the change. Use this information to determine how much in-depth assessment must take place for authorization. Based on this, also do the following:   * Change Owner creates new tasks, if additional assessment is needed and assigns owner to Change Analyst. * Change Analyst provides deeper technical assessment and/or business assessment. * Change Owner and/or Analyst determines if other CIs are impacted, using Impact Analysis tab. * Change owner and/or Analyst create Implementation tasks for Target and Impacted CIs.   Optionally use Move/Swap/Modify action to specify updates to CIs | | | |
| **Results** | |  | | | Pass  Fail |
| **Reason for failure or comments** | | |  | | |

## Change Management Test 4.2

| Test Number: 4.2.1 | Revision: 1.0 | | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- | --- |
| **Test Category:** Change Management | | | | **Standard/Customization:** Custom | |
| **Product:** TSRM | | | | **Requirement Catalogue reference:** Custom | |
| Test Title: | | Authorize - Change Approver | | | |
| Test Purpose | |  | | | |
| Procedure | | Ensure Change has all the information needed and then determine if the change should occur. If so, mark Approval task(s) "Complete". | | | |
| **Results** | |  | | | Pass  Fail |
| **Reason for failure or comments** | | |  | | |

## Change Management Test 4.3

| Test Number: 4.3.1 | Revision: 1.0 | | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- | --- |
| **Test Category:** Change Management | | | | **Standard/Customization:** Custom | |
| **Product:** TSRM | | | | **Requirement Catalogue reference:** Custom | |
| Test Title: | | Schedule Change Manager | | | |
| Test Purpose | |  | | | |
| Procedure | | Negotiate the schedule for when the change will be implemented. Update the list of all approved changes and their targeted dates for implementation. Communicate these schedules to stakeholders. This includes:   * Ensure Change Window(s) have been created previously for the Targeted and Impacted CIs. * Ensure Implementation Tasks have been created and target CIs have been assigned to them. * Schedule tasks using Task Scheduler. * Optionally create a Release to add the Change to a Release.   Mark Schedule tasks "Complete". | | | |
| **Results** | |  | | | Pass  Fail |
| **Reason for failure or comments** | | |  | | |

## Change Management Test 4.4

| Test Number: 4.4.1 | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Change Management | | | **Standard/Customization:** Custom | |
| **Product:** TSRM | | | **Requirement Catalogue reference:** Custom | |
| Test Title: | Implement - Change Implementer and Change Owner | | | |
| Test Purpose |  | | | |
| Procedure | * Implement all tasks in Change record, in the time period specified. Mark Implementation tasks "Complete".to add the Change to a Release.   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## Change Management Test 4.5

| Test Number: 4.5.1 | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Change Management | | | **Standard/Customization:** Custom | |
| **Product:** TSRM | | | Requirement Catalogue reference: Custom | |
| Test Title: | **Implement** - Change Implementer and Change Owner | | | |
| Test Purpose |  | | | |
| Procedure | * Implement all tasks in Change record, in the time period specified. Mark Implementation tasks "Complete".to add the Change to a Release.   Mark Schedule tasks "Complete". | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## Change Management Test 4.6

| Test Number: 4.6.1 | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Change Management | | | **Standard/Customization:** Custom | |
| **Product:** TSRM | | | **Requirement Catalogue reference:** Custom | |
| Test Title: | Post Implementation Review and Close - Change Owner | | | |
| Test Purpose |  | | | |
| Procedure | Review Change. Ensure implementation is complete without errors. Mark Change "Complete”. | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## Incident Test 4.7

| Test Number: 4.7.1 | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Incident Management | | | **Standard/Customization:** Custom | |
| **Product:** TSRM | | | **Requirement Catalogue reference:** Custom | |
| Test Title: | Create Incident through to Close Incident | | | |
| Test Purpose |  | | | |
| Procedure | * Login to Maximo using the Incident Agent account (UID=”Test1”, password=“Test1”) * Select ‘Go To’🡪’Service Desk’🡪’Incidents’ * Press the ‘New Incident’-button * Select ‘Reported by’ detail-button * Press ‘Select Value’ * Select ‘Hussain” * Specify ‘Affected Person’ “Hussain” * Specify ‘Summary’ “Cell Out of Service’?” * Press the ‘Classification’ detail-button * Select “Classify” * Select ‘Classification’ “MSC🡪BTS🡪Cell” * Press the ‘Urgency’ magnify-button * Select value “Low” * Press the ‘Impact’ magnify-button * Select value “Low” * The ‘Priority’ will automatically be calculated and set to ‘4’ (Low) * Press the ‘Call Code’ magnify-button * Select ‘Call Code’ “Phone” * Press the ‘Save’-button to save the incident * Switch to ‘Solution Details’-tab * Press the ‘Solution’ field detail button * Press ‘Select Value’ * Select the most likely solution * Communicate the ‘Resolution’ to the customer * Press the ‘Cause Code’ binocular-button * Select ‘Cause Code’ “Caused by ?????” * Press the ‘Close Code’ magnify-button * Select ‘Close Code’ “Complete The Incident was resolved normally” * Press ‘Select Action’ “Change Status” * Select ‘New Status’ “Closed” * Specify ‘Memo’ “Communicated solution, agreed closing” and press the ‘Ok’-button * The ‘Status’ of the Incident changes to “CLOSED” and the Incident will become read only | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |

## Incident Test 4.8

| Test Number: 4.7.1 | Revision: 1.0 | | Author: Nitesh Patel | Date: 18/08/2010 |
| --- | --- | --- | --- | --- |
| **Test Category:** Problem Management | | | **Standard/Customization:** Custom | |
| **Product:** TSRM | | | **Requirement Catalogue reference:** Custom | |
| Test Title: |  | | | |
| Test Purpose |  | | | |
| Procedure | * Select menu option ‘Go to’🡪’Service Desk’🡪’Problems’ * Press the ‘New Problem’-button * Specify ‘Reported by’ “DBA” * Press the ‘Phase’ magnify-button * Select the ‘Identify and Record Problem’ phase * Specify ‘Summary’ “Disk space improperly monitored”, ‘Details’ “Got a message from a Database Administrator who informed me that the disk space of a certain database server was very low. It seems that disk space is not correctly monitored (at least not on this server and perhaps on other servers as well?” and select ‘Asset’ “DBSVR001 * Select ‘Classification’ “MSC/BTS/CELL”, ‘Urgency’ “2 High”, ‘Impact’ “4 Local” * Select Action “Select Assignee” * Select “PMA001 Problem Analyst” * Login to Maximo using the Problem Analyst account (UID=””, password=“”) * Select the Problem as created and assigned to you in the previous steps * Select Action ‘Change Status’, set the ‘New Status’ to “WORKING”, fill the ‘Memo’-field and press the ‘Ok’-button * Press the ‘Phase’ magnify-button * Select the ‘Classify Problem’ phase * Press the ‘Save’-button * When needed Re-classify the Problem by pressing the ‘Classification’ detail-button and selecting another classification. Because of the fact that this is the first time the Problem Analyst sees the problem it is very unlikely that the Analyst changes the Classification. It is more likely that the Urgency and Impact are changed, but for now we leave them as is. * Press the ‘Phase’ magnify-button * Select the ‘Investigate and Diagnose Problem’ phase * Switch to the ‘Log’-tab, switch to the ‘Work Log’-sub-tab and press the ‘New Row’-button * The problem Analyst diagnoses the Problem; he finds out that indeed on that server disk space is not monitored effectively and that it is also the case on other database servers. * Specify ‘Type’ “Update’, specify ‘Summary’ “Checked disk monitoring”, specify ‘Details’ “Checked disk monitoring and discovered on that server disk space is not monitored effectively. This is also the case on other database servers” And press the ‘Save’-button * The problem is diagnosed and can be marked as ‘Known Error’. * Check the ‘Is Known Error?’ checkbox and specify the ‘Known Error Created’-field * Switch to the “problem’-tab and press the ‘Phase’ magnify-button * Select the ‘Identify and Record Error’ phase * Press the ‘Save’-button | | | |
| **Results** |  | | | Pass  Fail |
| **Reason for failure or comments** | |  | | |