# QAFCO 1.4 Wave 2 Test Strategy

## Amendment History:

| Name | Version | Date | Amendment History |
|---|---|---|---|
| Tharun Matta | 2.0 | 23-05-2024 | Created |

## Reviewers:

This document must be reviewed by the following:

| Name | Signature | Title / Responsibility | Date |
|---|---|---|---|
| Meenakshi Panda Sahu |  | Manager |  |
| Sonal Saraf |  | Team Lead |  |
| Krishna Mahto |  | Peer Review |  |

## Approvals:

This document must be approved by :

| Name | Signature | Title / Responsibility | Date | Version |
|---|---|---|---|---|
| Meenakshi Panda Sahu |  | Manager |  | V2 |

# 1. Introduction

## 1.1 Objectives

This document defines the Project Test Strategy and approach of incremental testing stages required to ensure the acceptability of the delivered solution. It covers all phases and releases. The foundation of the test and acceptance processes will be based on the acceptance statements and criteria within the features.

## 1.2 Document scope.

This test strategy will cover the definition of:

- Testing approach: Type of testing required to cover the software/hardware tech stack.
- Testing tools: Testing tools needed to perform all kinds of testing.
- Release control
- Risk Analysis
- Review and approvals

# 2. Testing Overview

## 2.1 Test Lifecycle

The test lifecycle to be followed is defined in detail within this section. The lifecycle is based on the agile development lifecycle, a model that is widely used in industry.

In an agile project:

- Feature testing occurs throughout each sprint.
- During the release planning meeting, the team should capture acceptance criteria and immediately add them as logical test cases linked to the product backlog item.
- In each sprint planning meeting, a feature level test plan is created and reviewed.
- As the testing is done in each sprint, the results should be tracked along with which features have been successfully tested and which ones have defects.
- Care should be taken to distinguish defects and changes in requirements. If the team requires a change in requirement, this should be added to the product backlog and prioritized.

## 2.2 Test Approach

Types of Testing Different types of testing occur at different points within the project implementation lifecycle. Each type of testing serves a different purpose, but all build up to final user acceptance testing before the system goes live.

1. Unit Testing
2. Component testing
3. Functional testing
4. User Acceptance testing
5. Regression testing

6. Non functional testing

Detailed explanation

**Leading indicators:**

- Defect Elimination status
- Critical Analysis,
- Maintenance Strategy
- PMO
- LTRP
- Action KPI

**Data Source:** Above KPIs rely directly on the OnePM data and APIs, API validation is not planned

**Validation Requirements:**

- For the Mentioned leading indicators, Validation will be conducted from API to UI, and API data will be used without formal validation.
- Since the OnePM APIs are not available test with mock data.
- Once integrated with OnePM APIs will have to do  End to End and confirm that the functionality is working as expected

## 2.3 Test Documentation

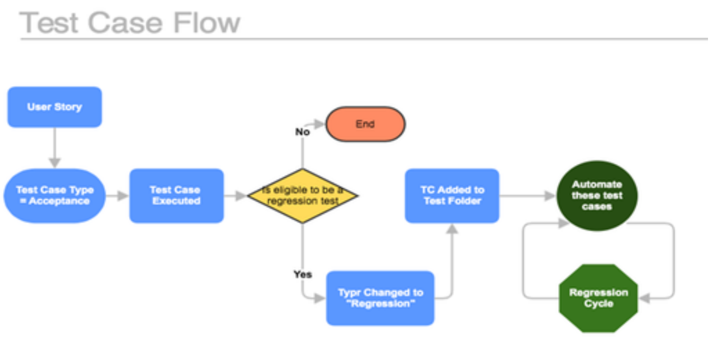
Identify all the Test Programmed documentation (along with their individual references) that will be delivered during each of the Test Phases and test cycles, and additionally highlight relevant contract delivery dates, where applicable.

As an example, this is set out as the table below:

| Document | Timetable |
|---|---|
| Test Strategy | Towards the Start of Project |
| Test Scripts | Before execution activities for given test type |
| Test Report | Draft due ***2*** days before the completion of that testing type. Baselined version due **1** day after completion of testing type. |
| Test Issue Logged | Reports available on demand |

## 2.4 Test Case

The data attributes and dependencies associated with the Test cases are as follows: Test Script Unique Identifier:


Test Case Flow
User Story
Test Case Type = Acceptance
Test Case Executed
Is eligible to be a regression test
No
End
Yes
Typr Changed to "Regression"
TC Added to Test Folder
Automate these test cases
Regression Cycle

## 2.6 Test traceability

This is important to trace test all the way up to feature level to know the feature acceptance percentage. Ultimately this signifies feature readiness for production push. Below diagrams explain the traceability from Defects to feature.

1. Each Feature is sliced down into the user stories.
2. Test cases are linked to Each User story.
3. Test cases are executed and marked with verdict as Pass and Fail.
4. Defects are created for each failed test case.

5. Test cases are written against Defects occurring from adhoc testing

## 2.7 Defect Management

Detailed Explanation

## 2.8 Test Reporting

Test reporting guidelines

## 2.9 Entry & Exit criteria for test execution

Entry & Exit criteria

Definition of Done for QA

# 3. Scoped Features

Rally Feature link:

| Feature ID | Name |
|---|---|
| F70249 | Root Cause Analysis KPIs |
| F70251 | Criticality Analysis KPIs |
| F70253 | Action KPIs |
| F70426 | FMEA, PMO KPIs |
| F70439 | LTRP KPI |
| F70962 | User Comments to describe plant performance |

# 4. Assumptions :

Factors which may affect the testing effort, and may increase the risk associated with the success of the test include:

- Completion of development of front-end processes and back-end process
- Completion of design and construction of new processes
- Completion of modifications to the local database
- Stability of all Api and database services
- Movement or implementation of the solution to the appropriate testing or production environment
- Stability of the testing or production environment
- Load Discipline
- Completion of manual testing through all applicable paths to ensure that reusable automated scripts are valid
- Fuzz & input validation testing while testing the application using Mock data
- Reports, Comment Service is tested for QAFCO NFR criteria by SPARQ team
- Report Service memory is not getting increased and not getting released if not used.

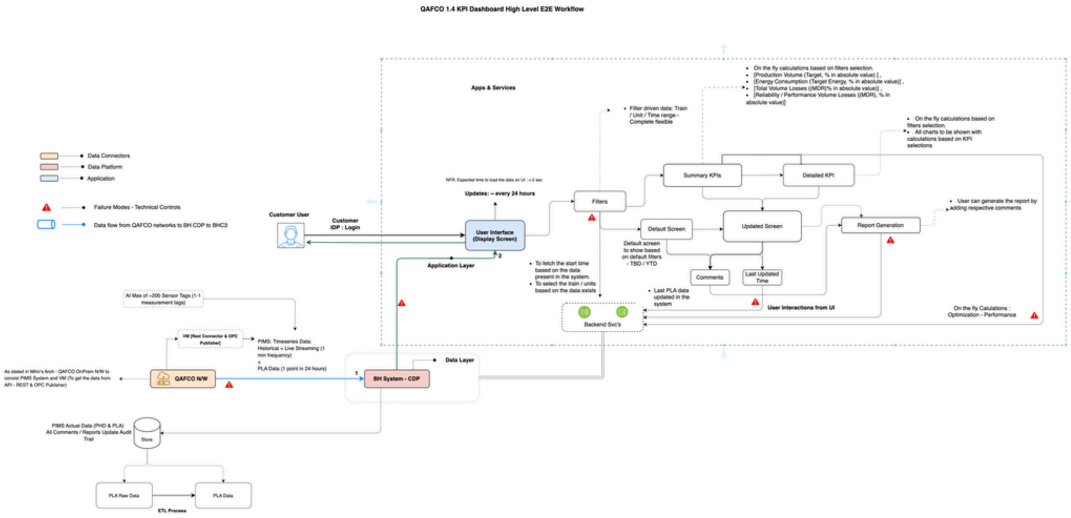
# 5. References

| Item description | Links |
|---|---|
| Figma link | https://www.figma.com/design/PwBKl09ZjWeT0BPsB13ZOr/1.4-Performance-Dashboard-R2--For-Developers?node-id=883-98062&t=X7GIkORN40ywAbXo-<br>0 Connect your Figma account |
| Rally Feature | As mentioned in **3. Scoped Features** |
| CDR | CDR 1.4 Performance Dashboard |
| NFR | **NFR Requirements:** 1.4 KPI Dashboard Nonfunctional Requirements [NFRs]<br>**FMEA Requirements:** 1.4 KPI Dashboard FMEA |

| | |
|---|---|
| Swagger | |

# 6. Functional Validation

## a. End to End Flow


QAFCO 1.4 KPI Dashboard High Level E2E Workflow
Data Connectors
Data Platform
Application
Failure Modes - Technical Controls
Data flow from QAFCO networks to BH CDP to BHC3
Apps & Services
Customer User
Customer
IDP : Login
NFR: Expected time to load the data on UI : < 2 sec
Updates: ~ every 24 hours
User Interface
(Display Screen)
Application Layer
2
Filters
Filter driven data: Train / Unit / Time range - Complete flexible
On the fly calculations based on filters selection.
[Production Volume (Target, % in absolute value) ] ,
[Energy Consumption (Target Energy, % in absolute value)] ,
[Total Volume Losses ((MDR)% in absolute value)] ,
[Reliability / Performance Volume Losses ((MDR), % in absolute value)]
On the fly calculations based on filters selection.
All charts to be shown with calculations based on KPI selections
Summary KPIs
Detailed KPI
Default Screen
Updated Screen
Report Generation
User can generate the report by adding respective comments
Default screen to show based on default filters - TBD / YTD
Comments
Last Updated Time
To fetch the start time based on the data present in the system.
To select the train / units based on the data exists
Last PLA data updated in the system
User Interactions from UI
On the fly Calulations : Optimization - Performance
Backend Svc's
At Max of ~200 Sensor Tags (1-1 measurement tags)
VM (Rest Connector & OPC Publisher)
PIMS: Timeseries Data: Historical + Live Streaming (1 min frequency)
PLA Data (1 point in 24 hours)
As stated in Mihir's Arch - QAFCO OnPrem N/W to consist PIMS System and VM (To get the data from API - REST & OPC Publisher)
QAFCO N/W
1
BH System - CDP
Data Layer
PIMS Actual Data (PHD & PLA)
All Comments / Reports Update Audit Trail
Store
PLA Raw Data
PLA Data
ETL Process

## b.  In QA Scope

| | | |
|---|---|---|
| **Feature** | F70249-RCA | |
| **Description** | Root cause Analysis KPI | |
| **Depth of Testing** | **Filters**<br>• Verify Filters Trains, Products, Select dates adheres to the Defect Elimination status KPI<br>**KPI**<br>• Verify the  title of the KPI should be Defect Elimination status<br>• Verify the completed RCA percentage should be displayed as per the selected Trains, Products, Selected dates<br>◦ As a example if the selected Trains is All Trains and Product is Ammonia and Select dates 01/01/2024 - 30/03/2024<br>◦ The Completed RCA percentage should be displayed among the total number of the RCA's for the selected date range which is given in example<br>◦ Calculation of the Completed RCA between the start and end is 20 and the In progress and Not started RCAs are 100 then the percentage will be (20/100*100) = 20%<br>• Verify text displays in the KPI for number of completed RCA among the total RCAs(from the above example 20 out of 100 RCAs )<br>**Defect Elimination status metrics**<br>• Verify Clicking on the Defect Elimination status kpi, Defect Elimination status metrics should be displayed<br>• Verify The charts displayed in Defect Elimination status metrics should be as following<br>◦ **RCA distribution by status**<br>◦ **RCA distribution by criticality**<br>◦ **RCA execution status by criticality** | |

**RCA distribution by status**

- Verify the title of the chart as RCA distribution by status
- Verify the list of status displayed are as per the onePM API response.
- Verify count of completed status the completed RCAs will be displayed along with the text for the selected trains, product, start and end date
- Verify count of completed status the In progress RCAs will be displayed along with the text for the selected trains, product, start and end date
- Verify count of Not started status the completed RCAs will be displayed along with the text for the selected trains, product, start and end date
- Verify the color indication will be displayed for each status
- Verify pie chart displays on the left side of the status
- Verify the pie chart is distributed as the color indications of status
- Verify in pie chart for completed status the color indication should be light blue
- Verify in pie chart for In Progress status the color indication should be Yellow
- Verify in pie chart for Not started status the color indication should be white
- Verify hovering on the different colors on the pie chart the tool tip should be displayed
- Verify the tooltip should be with following details
    - Suppose for completed status there are 20 out of total RCAS then the details on the tooltip should be displayed as 20/100 Completed and the percentage 20%
- Verify total number of RCAs should be displayed inside the pie chart
- Verify that user should be able to hover on the Completed, In progress, Not started status
- Verify that clicking on the any of the status RCA Distribution by status Details as right navigation bar should be displayed
- Verify based on status the RCA should be displayed
- Verify all the three tabs(Completed, IN progress, Not started) should be enabled
- Verify that Selected status tab will be selected in RCA Distribution by status Details
    - Suppose In progress status is clicked only the the In progress tab details should be shown
- Verify that user should be navigated between three tabs(Completed, IN progress, Not started)
- Verify that RCA details in each tab(Completed, IN progress, Not started) of the RCA distribution by status details should be displayed as following
    - RCA Name as header of the card
    - RCA Name
    - RCA status
    - Equipment Name
    - Equipment lost volume
    - Count of Actions
- Verify that RCA Distribution by status Details right navigation bar is scrollable
- Verify that RCA Distribution by status Details right navigation bar  should be closed clicking on the  cross bar or by clicking on outside the right navigation

**RCA distribution by criticality**

- Verify the title of the chart as “RCA distribution by criticality”
- Verify the type if criticality displays
    - High Critical RCA
    - Medium Critical RCA
    - Low Critical RCA
- Verify count of High Critical RCA will be displayed along with the text as per the selected trains, products, select dates
- Verify count of Medium Critical RCA will be displayed along with the text as per the selected trains, products, select dates
- Verify count of Low Critical RCA will be displayed along with the text as per the selected trains, products, select dates
- Verify the color indication will be displayed for each criticality
    - High Critical RCA color indication in light pink

    - Medium Critical RCA color indication in Yellow
    - Low Critical RCA color indication in Blue
- Verify pie chart displays on the left side of the criticalities
- Verify the pie chart is distributed as per the color indications of criticality
- Verify in pie chart for High critical  color indication should be light pink
- Verify in pie chart for Medium critical color indication should be Yellow
- Verify in pie chart for Low critical the color indication should be Blue
- Verify hovering on the different colors on the pie chart the tool tip should be displayed
- Verify the tooltip should be with following details
    - Suppose for High Critical RCA there are 37 out of total RCAs 39 then the details on the tooltip should be displayed as 37/39 High Critical RCA Completed and the percentage 94.87 High Critical RCA
- Verify total number of Critical RCAs should be displayed inside the pie chart
- Verify that user should be able to hover on the High, Medium, Low Critical text
- Verify that clicking on the any of the Criticality RCA Distribution by status Details as right navigation bar should be displayed
- Verify the title of the right navigation bar clicking on the criticalities
    - Clicking on High Critical the right navigation title should  be “High Critical Details“
    - Clicking on Medium Critical the right navigation title should  be “Medium Critical Details“
    - Clicking on Low Critical the right navigation title should be “Low Critical Details“
- Verify all the three tabs(Completed, IN progress, Not started) should be enabled
- Verify the default tabs shown on the right navigation clicking on the criticality
    - Clicking on High Critical the default tab details shown for High Critical details should be Completed
    - Clicking on Medium Critical the default tab details shown for High Critical details should be Completed
    - Clicking on Low Critical the default tab details shown for High Critical details should be Completed
- Verify that user should be navigated between three tabs(Completed, IN progress, Not started)
- Verify that RCA details in each tab(Completed, IN progress, Not started) of the RCA distribution by status details should be displayed as following
    - RCA Name as header of the card
    - RCA Name
    - RCA status
    - Equipment Name
    - Equipment lost volume
    - Count of Actions
- Verify that RCA Distribution by status Details right navigation bar is scrollable
- Verify that RCA Distribution by status Details right navigation bar  should be closed clicking on the  cross bar or by clicking on outside the right navigation

**RCA execution status by criticality**

- Verify title of the chart is “RCA execution status by criticality”
- Verify each criticality shows with the individual pie chart along with the status Completed, In progress, Not started
    - High critical shows with pie chart for the status  Completed, In progress, Not started
    - Medium Critical shows with pie chart for the status  Completed, In progress, Not started
    - Low Critical shows with pie chart for the status  Completed, In progress, Not started
- Verify each criticality shows the count of the three status(Completed, In progress, Not started) on the right side of the pie chart
- Verify the pie chart for each criticality is divided with different color indications for each status
    - Completed status color indication in light blue
    - Not started color indication in white

| | | |
|---|---|---|
| | ◦ In Progress color indication in Yellow<br>• Verify hovering on the pie chart the tooltip should be displayed<br>• Verify total number of RCA as per status should be displayed in pie chart for each criticality<br>• Verify clicking on the Status on of the Criticality the right navigation bar will be displayed<br>• Verify the title of the right navigation bar should be as following<br>◦ If each status from status list clicked in High critical then the title should be High Critical Details<br>◦ If each status from status list clicked in Medium critical then the title should be High Critical Details<br>◦ If each status from status list clicked in Low critical then the title should be High Critical Details<br>• Verify based on status the RCA should be displayed<br>• Verify all the three tabs(Completed, IN progress, Not started) should be enabled<br>• Verify that Selected status tab will be selected in RCA Distribution by status Details<br>◦ Suppose In progress status is clicked only the the In progress tab details should be shown<br>• Verify that user should be navigated between three tabs(Completed, IN progress, Not started)<br>• Verify that RCA details in each tab(Completed, IN progress, Not started) of the RCA distribution by status details should be displayed as following<br>◦ RCA Name as header of the card<br>◦ RCA Name<br>◦ RCA status<br>◦ Equipment Name<br>◦ Equipment lost volume<br>◦ Count of Actions<br>• Verify that RCA Distribution by status Details right navigation bar is scrollable<br>• Verify that RCA Distribution by status Details right navigation bar should be closed clicking on the cross bar or by clicking on outside the right navigation | |
| **Out of Scope** | • OnePM API validation<br>• OnePM data validation | |
| **Failure Modes** | • Verify UI behavior when API is giving 500 Internal server error.<br>• Verify UI behavior when API is giving 404 not found error. | |
| **Setup** | Test with actual data from DB and dashboard service. In case of unavailability of actual data, test the feature with mock JSON data (in QA). | |
| **Build Release Approach** | Following points will be considered while releasing build to consumer:<br>1. Regression will be performed as part of hardening on the final build<br>2. Post QA-Sign off build will be ready for release<br>3. Build will be deployed on pre-prod (for testing purpose hence not on prod environment)<br>4. Sanity will be performed on the deployed build by QA on pre-prod with Actual data<br>5. PO & TPM team team to carry testing along with platform team<br>6. Post Sign-off from QA, PO & TPM team build will be deployed on Prod environment of consumer | |
| **Testing Approach** | • Manual Validation to be done for UI changes and its functionality according to below criteria.<br>◦ This feature validation will be conducted from API to UI, and API data will be used without formal validation.<br>◦ Since the OnePM APIs are not available testing will be done with mock data.<br>◦ Once integrated with OnePM APIs will have to do End to End and confirm that the functionality is working as expected<br>• Automation will be covered for manually tested stories using IntelliJ IDEA 2022.3.3 (Community Edition) | |

| | | |
|---|---|---|
| | • Making sure to cover 80% of automation coverage before each release | |

| | |
|---|---|
| **Feature** | F70251- Criticality Analysis KPI |
| **Description** | Equip with insights into the overall risk profile of the equipment's, derived from the criticality analysis. This will allow them to monitor the effectiveness of maintenance strategies based on criticality analysis and ultimately demonstrate a decrease in overall risk over time. |
| **Depth of Testing** | **Filters**<br>• Verify that criticality analysis adheres with the Trains and the Product filters<br>• Verify that criticality analysis doesn’t adheres with the select dates<br>• Verify that clicking on the criticality analysis KPI select date will be grayed out<br>• Verify that hovering on the grayed out select dates tooltip should be displayed<br>• Verify that tooltip contains the Message “Click on the other KPI to get select dates enabled“<br>**KPI**<br>• Verify that clicking on the right arrow to navigate between the KPIs the Critical analysis KPI should be displayed<br>• Verify that title of the KPI should be “Critical Analysis“<br>• Verify that only the safety critical analysis should be displayed in the Criticality analysis KPI<br>◦ Suppose for the selected train and the product there are 20 safety critical then those numbers should be displayed in the KPI<br>• Verify that “Safety critical equipment flagged“ text displays at the bottom of the Number in the KPI<br>• Verify that clicking on the critical analysis KPI the Critical analysis metrics displays<br>• Verify the critical analysis metrics contains the following charts<br>◦ Criticality Distribution<br>◦ Probability of Failure Distribution<br>◦ Consequence Distribution<br>◦ Asset Heatmap<br>◦ Cumulative Risk Curve<br>**Criticality Distribution:**<br>• Verify the title of the chart is “Criticality Distribution”<br>• Verify the chart should be displayed for Number of Assets vs the Criticality Distribution categories<br>• Verify that in Y-axis the Number of assets should be displayed<br>• Verify that in x-axis Criticality Distribution categories should be displayed and those should be displayed as per categories fetched from 1pm API<br>◦ As an example we can consider criticality distribution categories as (Low, Medium, High, Sever)<br>• Verify each criticality distribution category represents in horizontal bar to denote the number of assets with respective to the Y-axis<br>• Verify that hovering on the bars tooltip should be displayed<br>• Verify the details in the tooltip should be as following<br>◦ Suppose the number of assets are 22314 for Low Criticality distribution category then the hovering on the bar the details should be shown as 22314-Low<br>**Probability of Failure Distribution**<br>• Verify the title of the chart is “Probability of Failure Distribution”<br>• Verify that chart should be displayed for Number of Assets & assets in % vs Probability of Failure Distribution category<br>• Verify that in Y-axis the Number of assets on the left side and the assets in % on the right side<br>• Verify that in x-axis Probability of Failure Distribution categories should be displayed and those should be displayed as per categories fetched from 1pm API<br>◦ As an example we can consider Probability of Failure Distribution categories as (Rare, Likely, Unlikely, Moderate, Almost certain) |

- Verify each Probability of Failure Distribution category represents in horizontal bar to denote the number of assets with respective to the Y-axis
- Verify the each horizontal bar represents the Number of assets and assets in %
  - As an example for the select trains and products the total number of probability of Failure distribution assets are 100 among 100 the rare distribution category holds 20 then assets in % calculated as 20/100*100 = 20%
- Verify that hovering on the bars tooltip should be displayed
- Verify that details in the tooltip should be as following
  - Suppose the number of assets are 20 for Rare Probability of Failure distribution category then the hovering on the bar the details should be shown as 20(20%)-Rare

**Consequence Distribution**

- Verify the title of the chart is “Consequence Distribution”
- Verify that chart should be displayed for Number of Assets & assets in % vs Consequence Distribution category
- Verify that in Y-axis the Number of assets on the left side and the assets in % on the right side
- Verify that in x-axis consequence distribution categories should be displayed and those should be displayed as per categories fetched from 1pm API
  - As an example we can consider Consequence Distribution categories as (Rare, Likely, Unlikely, Moderate, Almost certain)
- Verify each Consequence Distribution category represents in horizontal bar to denote the number of assets with respective to the Y-axis
- Verify the each horizontal bar represents the Number of assets and assets in %
  - As an example for the select trains and products the total number of Consequence Distribution assets are 100 among 100 the rare distribution category holds 20 then assets in % calculated as 20/100*100 = 20%
- Verify that hovering on the bars tooltip should be displayed
- Verify that details in the tooltip should be as following
  - Suppose the number of assets are 20 for Rare consequence distribution category then the hovering on the bar the details should be shown as 20(20%)-Rare

**Asset Heatmap**

- Verify that title of the chart is “Asset Heatmap“
- Verify the chart will be displayed in matrix type for the consequence type and likelihood type
- Verify that consequence type should be displayed as a header of the matrix
- Verify that for each consequence type the likelihood type value should be displayed
- Verify that consequence types are fetched from 1pm API and those should be displayed as header
  - As a example the consequence types are as following (Very low, minor, medium, major catastrophic)
- Verify that likelihood types are fetched from 1pm API
  - As a example the consequence types are as following (Almost certain, Likely, Unlikely, Rare, Moderate)
- Verify that legends display with the color indication at the bottom of the chart
  - Low legend with green color
  - Medium legend with light blue color
  - High legend with orange color
  - Sever legend with red color
- Verify that each consequence type has all likelihood type value
  - As a example for Medium consequence type  all the likelihood type (Almost certain, Likely, Unlikely, Rare, Moderate) values should be displayed in grid and each grid should represent the color of the medium legend which is light blue

**Cumulative Risk Curve**

- Verify the title of the chart should be “Cumulative Risk Curve“
- Verify dropdown displays at the top right corner of the chart
- Verify dropdown values should be as following

| | |
|---|---|
| | ◦ All Trains<br>◦ Year<br>• Verify that the y-axis of the graph should display the millions<br>• Verify that the x-axis of the graph should display the number of assets<br>• Verify that when All Trains dropdown option selected all trains cumulative curve and also the individual trains curve should be displayed<br>• Verify that when All Trains dropdown selected legends should be displayed as All Train, Train1, Train2, Train3, Train4, Tain5, Train6<br>• Verify that each legend indicates in different color codes when All trains dropdown option selected<br>◦ All Trains legend with blue<br>◦ Train1 with light blue<br>◦ Train2 with purple<br>◦ Train3 with shade yellow<br>◦ Train4 with light pink<br>◦ Train5 with  light yellow<br>◦ Train6 with green<br>• Verify that cumulative curve displays in years when the dropdown option Years selected<br>• Verify that for the selected trains and products how many years data fetched from 1pm API those many years cumulative curves should be displayed<br>• Verify that years displayed as legends at the bottom of the chart |
| **Out of Scope** | • OnePM API validation<br>• OnePM data validation |
| **Failure Modes** | • Verify UI behavior when API is giving 500 Internal server error.<br>• Verify UI behavior when API is giving 404 not found error. |
| **Setup** | Test with actual data from timeseries service and PO service. In case of unavailability of actual data, test the feature with mock JSON data (in QA). |
| **Approach** | QA and Preprod env<br>• Manual Validation to be done for UI changes and its functionality according to below criteria.<br>◦ This feature validation will be conducted from API to UI, and API data will be used without formal validation.<br>◦ Since the OnePM APIs are not available testing will be done with mock data.<br>◦ Once integrated with OnePM APIs will have to do  End to End and confirm that the functionality is working as expected<br>• Automation will be covered for manually tested stories using IntelliJ IDEA 2022.3.3 (Community Edition)<br>• Making sure to cover 80% of automation coverage before each release |
| **Automation** | |

| | |
|---|---|
| **Feature** | F70253-Actions KPI |
| **Description** | Actions originate from various sources within Cordant. Management needs to monitor these actions to ensure smooth plant operations and achieve company goals. Allow executive leadership within the customer company to see the status and flow of action processing so they can see if the business management is in good health or not, and if not where the problems lie. |
| **Depth of Testing** | **Filters**<br>• Verify that Action KPI is applicable with the Filters Trains, Product, Select dates<br>**KPI**<br>• Verify Action KPI titled with “Actions”<br>• Verify Only the completed Actions will be displayed in the KPI<br>**Example:**<br>Suppose start and end dates given as following 01/01/2024 to 31/03/2024 |

- If Action is started before start date(01/01/2024) and got completed before the end data(31/03/2023) then the Action will be considered as Completed
- If Action is started before start date(01/01/2024) and got completed after the end data(31/03/2023) then the Action will be considered as not completed
- If Action is started after start date(01/01/2024) and got completed within the end data(31/03/2023) then the Action will be considered as completed
- If Action is started after start date(01/01/2024) and got completed after the end data(31/03/2023) then the Action will be considered as not completed
- Verify clicking on the Action KPI the Action metrics will be displayed
  - **Actions by Actions Priority**
  - **Actions by Actions Status**

**Actions by Actions Priority**

- Verify the title of the chart is Actions by Actions Priority
- Verify the priority in Actions by Actions Priority
  - High
  - Medium
  - Low
- Verify the count of High Actions should be displayed for the selected Trains, Product, Select dates
- Verify the count of Medium Actions should be displayed for the selected Trains, Product, Select dates
- Verify the count of Low Actions should be displayed for the selected Trains, Product, Select dates
- Verify each Action(High, Medium, Low) is denoted with different color indication
  - High with Light pink
  - Medium with Yellow
  - Low with thick blue
- Verify the pie chart should be displayed at the left side if the Actions
- Verify the pie chart distributed among the Actions with different color indication
  - High Action in pie chart the color indication should be in pink
  - Medium Action in pie chart the color indication should be in Yellow
  - Low Action in pie chart the color indication should be in blue
- Verify hovering on the colors in pie chart the tooltip will be displayed
- Verify the details shown in the tooltip
  - Suppose if the High Actions are 7 and the total number of Actions are 27 then in tooltip the details should be shown as High Actions 7(25.93%)

**Actions by Actions Status**

- Verify the title of the chart is “Actions by Actions Status“
- Verify the Action Status should be as per onePM API response
- Verify that each Action status shows with the count and the percentage
- Verify that each Actions status indicated with the different colors
  - Not started indicated as light orange
  - In progress indicated as yellow
  - Refused indicated as light green
  - Overdue indicated as light blue
  - Closed indicated as light pink
- Verify that each status displays with the count and the percentage
  - Suppose Not started Action status are 4 out of 20 Actions then the count will be shown as 4 and percentage will be shown as 4/20*100 = 20 visually it will be shown as 4(20%)
- Verify that pie chart is divided with individual color for all the Action status
- Verify that hovering on the pie chart the tooltip will be displayed
- The tooltip should be displayed as following
  - Suppose Not started Action status are 4 out of 20 Actions then the count will be shown as 4 and percentage will be shown as 4/20*100 = 20  Not started 4(20%)

| | |
|---|---|
| | • Verify the total count of the Actions by status will be displayed inside the pie chart |
| **Out of Scope** | • OnePM API validation<br>• OnePM data validation |
| **Failure Modes** | • Verify UI behavior when API is giving 500 Internal server error.<br>• Verify UI behavior when API is giving 404 not found error.<br>• Verify UI behavior when there are no Actions for the selected Trains and Products |
| **Setup** | Test with actual data from timeseries service and PO service. In case of unavailability of actual data, test the feature with mock JSON data (in QA). |
| **Approach** | QA and Preprod env<br><br>• Manual Validation to be done for UI changes and its functionality according to below criteria.<br>◦ This feature validation will be conducted from API to UI, and API data will be used without formal validation.<br>◦ Since the OnePM APIs are not available testing will be done with mock data.<br>◦ Once integrated with OnePM APIs will have to do  End to End and confirm that the functionality is working as expected<br>• Automation will be covered for manually tested stories using IntelliJ IDEA 2022.3.3 (Community Edition)<br>• Making sure to cover 80% of automation coverage before each release |
| **Automation** | |

| **Feature** | F70426-FMEA, PMO KPIs, F70439 - LTRP KPI | |
|---|---|---|
| **Description** | • Identifying the importance of each machine based on its impact on operations.<br>• Determining the most effective maintenance schedules for each machine to prevent breakdowns.<br>• Analyzing potential failure modes of each machine and their consequences. | |
| **Depth of Testing** | • **Filters**<br>◦ Verify that Maintenance Strategy adheres with the Trains and the Product filters<br>◦ Verify that Maintenance Strategy doesn’t adheres with the select dates<br>◦ Verify that clicking on the Maintenance Strategy KPI select date will be grayed out<br>◦ Verify that hovering on the grayed out select dates tooltip should be displayed<br>◦ Verify that tooltip contains the Message “Click on the other KPI to get select dates enabled“<br><br>**KPI**<br><br>• Verify the title of the KPI should be “Maintenance Strategy”<br>• Verify the following text should be displayed in KPI  **FMEA & PMO, LTRP**<br>• Verify clicking on  the KPI Maintenance Strategy Metric should be displayed<br>◦ **Applicable Strategy vs Total Number of Assets**<br>◦ **Total Assets for Long term replacement plan**<br>◦ **Long Term Replacement Plan Actions % Complete**<br>◦ **Long Term Replacement Plan Cost Over the Years**<br>◦ **Long Term Replacement Plan Details**<br><br>**Applicable Strategy vs Total Number of Assets** | |

- Verify the title of the chart is “Applicable Strategy vs Total Number of Assets“
- Verify that Applicable Strategy vs Total Number of Assets should represent in tabular form for the all strategies
- Verify the  table should contain four columns
    - Strategy
    - Applicable Asset
    - Include Scope
    - % scope
- Verify There are four type of strategies
    - Failure Mode and effects analysis
    - Preventative Maintenance Optimization
    - Run to Failure
    - Existing Strategy
- Verify that for each type of strategy there will be Applicable Assets, number of assets included in the scope of the percentage and the % scope
- Example:
    - Suppose there are 100 Applicable assets for the Failure Mode and Effect Analysis among the 100 the Include scope is 60 and % scope is 60/100*100 = 60%
    - Suppose there are 1000 Applicable assets for the Preventative Maintenance Optimization among the 1000 the Include scope is 200 and % scope is 200/1000*100 = 20%
    - Suppose there are 5000 Applicable assets for the Run to Failure among the 5000 the Include scope is 500 and % scope is 5000/5000*100 = 100%
    - Suppose there are 5000 Applicable assets for the Existing Strategy among the 5000 the Include scope is 500 and % scope is 5000/5000*100 = 100%
- Verify that View report button should be displayed at the top of the table
- Verify that table shouldn’t be sortable

**Total Assets for Long term replacement plan**

- Verify the title of the chart should be “Total Assets for long term replacement Plan”
- Verify the count of the “Total Assets for long term replacement plan“ should be displayed as per selected Trains and Products

**Long Term Replacement Plan Actions % Complete**

- Verify the title of the tile should be Long Term Replacement Plan Actions % Complete
- Verify the details should show the percentage of the Long Term Replacement Plan Actions % Completed for the selected Trains and Products Filter
- Example: Suppose there are 1700 Long Term Replacement Plan Actions % Completed for the selected Trains and Products out of 1700 only 17 Long Term Replacement Plan Actions % Completed **(17/1700*100 = 1%) the 1% should be displayed in the tile**

**Long Term Replacement Plan Cost Over the Years**

- Verify the title of the graph is Long Term Replacement Plan Cost over the Years
- Verify the chart is interpreted as graph between Years vs Cost
- Verify the x-axis represents the Year
- Verify the Y-axis represents the Cost
- Verify the each year cost represents in the horizontal bars

| | | |
|---|---|---|
| | • Verify hovering on the horizontal bars tooltip should be displayed<br>• Verify the details in the tooltip<br>○<br>**Long Term Replacement Plan Details**<br>• Verify the title of the chart should be “Long Term Replacement Plan Details”<br>• Verify the chart represents in the tabular form<br>• The columns of the table are<br>○ Plant<br>○ Units<br>○ Equipment Name<br>○ To be Replaced(Year)<br>○ During TAR(Yes/No)<br>○ Driver for<br>○ Replacement Cost<br>○ Department<br>• Verify all the table columns should be sortable<br>• Verify the table should be paginated<br>• Verify the minimum rows displayed for page should be 10<br>• Verify switching between the pages the sorting remains the same<br>• Verify next page navigation icon should be in disable state if user is in the last page<br>• Verify previous page navigation icon should be in disable state if user in the first page | |
| **Out of Scope** | • OnePM API validation<br>• OnePM data validation | |
| **Failure Modes** | • Verify UI behavior when API is giving 500 Internal server error.<br>• Verify UI behavior when API is giving 404 not found error.<br>• Verify UI behavior when there are no Actions for the selected Trains and Products | |
| **Setup** | Test with actual data from timeseries service and PO service. In case of unavailability of actual data, test the feature with mock JSON data (in QA). | |
| **Approach** | QA and Preprod env<br>• Manual Validation to be done for UI changes and its functionality according to below criteria.<br>○ This feature validation will be conducted from API to UI, and API data will be used without formal validation.<br>○ Since the OnePM APIs are not available testing will be done with mock data.<br>○ Once integrated with OnePM APIs will have to do End to End and confirm that the functionality is working as expected<br>• Automation will be covered for manually tested stories using IntelliJ IDEA 2022.3.3 (Community Edition)<br>• Making sure to cover 80% of automation coverage before each release | |
| **Automation** | | |

| | |
|---|---|
| **Feature** | - User Comments to describe plant performance |
| **Description** | The customer could annotate charts with explanatory comments and also have an executive summary that gave an up front succinct status of the plant performance without having to dive |

| | |
|---|---|
| | into deeper charts of performance |
| **Depth of Testing** | **KPI**<br>• Verify on each KPI add note icon should be displayed<br>• Verify hovering on the add icon button on KPIs “Add Executive Summary“ tooltip should be displayed<br>• Verify “Add Executive Summary“ pop up displays clicking on the “Add Executive Summary“ icon<br>• Verify title of the pop up should be “Add Executive Summary“<br>• Verify layout of the “Add Executive Summary“ popup should be as following<br>◦ Select Trains<br>◦ Select Relevance date<br>◦ What went well with thumbs up icon<br>◦ What didn’t go well with thumbs down icon<br>◦ Save button at the bottom<br>◦ Cross button at the top right corner<br>• Verify Product and KPI should be displayed in “Add Executive Summary<br>◦ Suppose if user clicks on the “Add Executive Summary“ icon in production volume kpi then the KPI will displayed as Production Volume<br>◦ Suppose the product filter is selected as Ammonia then in the popup the product will be displayed as Product<br>• Verify select train dropdown displays in “Add Executive Summary“ pop up<br>• Verify dropdown options displays as following clicking on the select train dropdown<br>◦ ALL TRAINS, Y1, Y2, Y3, Y4, Y5, Y6<br>• Verify the by default select train dropdown option should be All Trains<br>• Verify the Relevance date should be should be selected in the format MM/DD/YYYY<br>• Verify the Relevance date should not be selected more than five years<br>• Verify the future date should not be selected as Relevance date<br>• Verify What went well text should not be greater than 500 characters<br>• Verify what want well text character should be within the 500 characters<br>• Verify What didn’t go well text should not be greater than 500 characters<br>• Verify what didn’t go well text character should be within the 500 characters<br>• Verify after filling all the mandatory details(Select Relevance Date) Save button enables<br>• Verify clicking on the save button the popup should be appeared<br>**Detailed Metric of each KPI**<br>• Verify each KPI detailed chart contains the Add Note icon<br>• Verify hovering on the add note icon tooltip should be appeared as “Add Note“<br>• Verify clicking on the Add Note the add note popup should be displayed<br>• Verify the title of the popup should be Add Note<br>• Verify the layout of the Add Note popup<br>◦ Product<br>◦ Metric<br>◦ Select Relevance Date<br>◦ Add Note text box<br>◦ Cross button at the top right<br>• Verify the product and metric should be displayed as following<br>◦ Suppose the add icon clicked on the production volume trend chart then the Metric should be displayed as Production volume Trend<br>◦ Suppose Product filter selected as Ammonia then the product will be displayed as Ammonia<br>• Verify the Relevance date should be should be selected in the format MM/DD/YYYY<br>• Verify the Relevance date should not be selected more than five years<br>• Verify the future date should not be selected as Relevance date<br>• Verify the add note character should be within 700 characters<br>• Verify add note text box shouldn’t allow more than 700 characters |

- Verify save button enable by filling the mandatory details (Select Relevance Date)
- Verify clicking on the Save button the popup should be enabled

**View Notes**

- Verify view Note icon displays between the last updated date and the Generate report popup
- Verify clicking on the view Note icon clock icon displays to state the last updated date
- Verify hovering on the clock icon tooltip should appear to state the last updated date
- Verify the layout of the view Note Panel should be as following
    - Notes as title of the panel
    - Cross button in the top right to close the panel
    - Comment should be displayed for Executive summary and the for the Notes
- Verify the layout of how Executive summary comment displays in the Notes panel
    - Suppose the Executive summary added as following
        - Executive summary added in production volume KPI
        - Trains selected as A1
        - Relevance date as 03/31/2024
        - Created by  User 1
        - Created date 04/20/2024
    - from the above given details the Executive summary Note should be displayed as following
        - User1
        - 04/20/2024
        - Train A1  Ammonia Production volume
        - Relevance date 03/31/2024
        - What went well Note with thumbs icon
        - What didn't go well with thumbs down icon
        - See more clickable text if the comment is more than 90-95 characters
- Verify that clicking on see more text complete Executive summary should be displayed and also see less clickable text should be displayed
- Verify that clicking on see less clickable text the comment should be displayed in 90-95 characters and also see more clickable text should be displayed
- Verify the layout of how Add Note comment displays in the Notes panel
    - Suppose the Executive summary added as following
        - Add Note added in the production volume per unit
        - Trains selected as A1
        - Relevance date as 03/31/2024
        - Created by  User 1
        - Created date 04/20/2024
    - from the above given details the Executive summary Note should be displayed as following
        - User1
        - 04/20/2024
        - Train A1  Ammonia Production volume
        - Relevance date 03/31/2024
        - Notes that is added
        - See more clickable text if the comment is more than 140-145 characters
- Verify that clicking on see more text complete add note comment should be displayed and also see less clickable text should be displayed
- Verify that clicking on see less clickable text the comment should be displayed in 140-145 characters and also see more clickable text should be displayed
- Verify clicking on the close button Notes panel should be closed
- Verify that after closing the Notes panel clock to state the last updates date should not be displayed
- Verify that last updated date should be displayed as usual after closing the Notes panel

**Edit/Delete Comment**

- verify the three horizontal dots displays on the comment in the view note panel
- Verify the EDIT and DELETE options should be displayed clicking on the three horizontal dots on the comment
- Verify the UI should be as following clicking on the Edit option on the Executive summary in view notes panel
    - Edit Note as header
    - Edit fields should be Select Relevance Date, What went well, What didn't go well, Trains dropdown
    - Cancel button in enable state, Save button in disable state
- Verify the UI should be as following on the Edit option on the Executive summary comment in view notes panel
    - Edit Note as header
    - Edit fields should be Select Relevance Date, Notes Trains dropdown
    - Cancel button in enable state, Save button in disable state
- Verify the Non editable fields in Executive summary comment and the add note comment in the view not panel
    - Executive Summary Note: Product and KPI title
    - Add Note: Product and Metric chart title
- Verify that user should be able choose the Trains dropdown while editing the Executive summary note and add note
- Verify that user should be able to select relevance date with in the 5 years while editing the Executive summary and add note comment
- Verify that UI should handle the error of choosing relevance date greater than 5 years while editing the Executive summary and add note comment
- Verify that error handling if number of characters in what went well and the What didn't went well exceeds more than 500 characters while editing
- Verify that error handling if number of characters in add note exceeds more than 500 characters while editing
- Verify that clicking on the save button after editing the Executive summary comment or add note comment, comment should get saved successfully and the toast message should be appeared
    - Toast message: Comment Edited successfully
- Verify that edit should cancel clicking on the cancel button after editing the Executive summary comment or add note comment
- Verify that comment should get deleted clicking on the Delete in the delete popup
- Verify the layout of the delete note popup should be as following
    - Delete Note? as header
    - are you sure you want to delete this note? as indication
    - Cancel and Delete button
    - cross button at the top of the popup
- Verify that delete note popup should be displayed clicking on the delete button in add executive summary comment and add note comment
- Verify that delete note should be canceled clicking on the Cancel button or the cross button the Delete Note popup
- Verify that clicking on the Delete in the delete popup toast message should be appeared
    - Toast message: comment deleted successfully
- Verify that deleted comment should not be displayed in the view note panel
- Verify that comment created date should get updated after editing the comment
    - Suppose the comment created at March 17, 2024 10:30 AM

      then the comment edited on April 18, 2024 12:00 PM

      then comment created date should be displayed as

      April 18, 2024 12:00 PM
- Verify the comment should be displayed as per edited after saving the edited comment
    - Suppose while creating the comment the comment is saved with the following details

      Trains A1

| | |
|---|---|
| | Relevance date: 10/03/2024<br>What went well: Everything goes well<br>What didn't go well: NA<br>And edited as following<br>Trains All Trains<br>Relevance date: 10/04/2024<br>What went well: NA<br>What didn't go well: NA<br>◦ Then the edited comment will be displayed as following<br>▪ Trains All Trains<br>Relevance date: 10/04/2024<br>What went well: NA<br>What didn't go well: NA |
| **Out of Scope** | • OnePM API validation<br>• OnePM data validation |
| **Failure Modes** | • Verify UI behavior when API is giving 500 Internal server error.<br>• Verify UI behavior when API is giving 404 not found error.<br>• Verify UI behavior when comments service is down. |
| **Setup** | Test with actual data from timeseries service and PO service. In case of unavailability of actual data, test the feature with mock JSON data (in QA). |
| **Approach** | QA and Preprod env<br>• Manual Validation to be done for UI changes and its functionality according to below criteria.<br>◦ This feature validation will be conducted from API to UI, and API data will be used without formal validation.<br>◦ Since the OnePM APIs are not available testing will be done with mock data.<br>◦ Once integrated with OnePM APIs will have to do  End to End and confirm that the functionality is working as expected<br>• Automation will be covered for manually tested stories using IntelliJ IDEA 2022.3.3 (Community Edition)<br>• Making sure to cover 80% of automation coverage before each release |
| **Automation** | |

## c. Not in QA Scope

Non-Functional Requirements (Security, Tenancy, Configuration/ Deployment/ Installation/ Upgrades/ Troubleshooting)

# 7. Non-Functional Validation:

## 7.1 Performance CTQ

**NFR Setup**

| Sentralregisteret | **Attributes** | **Maximum Values** | **Average or Good to have** |
|---|---|---|---|
| 1 | **Max time to load the charts (detailed) - All KPIs selected one by one. [Custom filter driven]** | **~5 sec** | **~3 sec** |
| | **If the default filters are YTD** | **~3 sec** | **~2 sec** |
| 2 | **Max time to load the summary KPI's** | **~3 sec** | **~2 sec** |
| 3 | **Max time user can go back in history of dates \| Data retention in the system - For few for filtered** | **Multiple years (~5+ years) : Manually** | |

| | | | |
|---|---|---|---|
| | **KPIs graphs** | **Classified** | |
| 4 | **API Endpoint Responses - CRUD Ops (Internal + External, if any) - Linked to Point#1** | ~3 sec | ~2 sec |
| 5 | **No of concurrent users** | **~50** | **~30 for now** |
| 6 | **Data size (PLA) : Per day / month / years** | For 12 units / 12 Trains: Ammonia and Urea:<br>PLA Data:<br>24 hours = ~1 mb (at max)<br>1 month = ~30 mb<br>1 year =  ~360 mb<br>5 years = ~1.8 mb<br>10 years = ~3.6 GB | |
| 7 | **Time taken from data to be sent from CDP to App layer** | | |
| 8 | **Application to highlight the data updated** | **Near to time - same needs to be shown on UI. with date & time** | |
| 9 | **App Page refresh - For last updated time - PLA Data** | **[By Refresh / Logout & Login], No on the fly updates on screen required for now.** | |

| | |
|---|---|
| No of concurrent users | |

**API NFR Table**

| **API Name** | **LSL** | **USL** | **10 users 100 loops** | **Error(%)** | **30 users 100 loops** | **Error(%)** | **50 users 100 loops** | **Error(%)** |
|---|---|---|---|---|---|---|---|---|
| **AssetDetails** | 2s | 3s | | | | | | |
| **EnergyConsumptionMonthWise** | 2s | 3s | | | | | | |
| **GenerateToken** | 2s | 3s | | | | | | |
| **LastUpdated** | 2s | 3s | | | | | | |
| **ProductionVolumeDaily** | 2s | 3s | | | | | | |
| **R&P_losses_distribution_per_plant_system** | 2s | 3s | | | | | | |
| **R&P_volume_losses_daily** | 2s | 3s | | | | | | |
| **R&P_volume_losses_monthwise** | 2s | 3s | | | | | | |
| **TotalVolumeMonthWise** | 2s | 3s | | | | | | |
| **volumeLossCategoryWise** | 2s | 3s | | | | | | |
| | | | | | | | | |
| **PlantsStopEventDetails** | 2s | 3s | | | | | | |
| **ProductionLossEvent** | 2s | 3s | | | | | | |

**Longevity Performance Dashboard KPIs NFR**

| Performance dashboard KPI | LSL | USL | 3 users infinite loops | Error(%) |
|---|---|---|---|---|
| **Production volume KPI and Production volume Metrics Charts** | 2s | 3s | | |
| **Energy Consumption KPI and Energy Consumption Metrics Charts** | 2s | 3s | | |
| **Volume Losses KPI and Volume Losses Metrics Charts** | 2s | 3s | | |
| **Reliability and performance volume losses and Reliability performance volume losses Metrics Charts** | 2s | 3s | | |
| **Defect Elimination KPI and Defect Elimination Metrics Charts** | 2s | 3s | | |
| **Critical Analysis KPI and Critical Analysis Metrics Charts** | 2s | 3s | | |
| **Maintenance Strategy KPI and Maintenance Strategy Metrics Charts** | 2s | 3s | | |
| **Action KPI and Action Metrics Chats** | 2s | 3s | | |

**Performance Dashboard KPIs NFR**

| Performance dashboard KPI | LSL | USL | 5 users 50 loops | Error(%) | 10 users 50 loops | Error(%) |
|---|---|---|---|---|---|---|
| **Production volume KPI and Production volume Metrics Charts** | 2s | 3s | | | | |
| **Energy Consumption KPI and Energy Consumption Metrics Charts** | 2s | 3s | | | | |
| **Volume Losses KPI and Volume Losses Metrics Charts** | 2s | 3s | | | | |
| **Reliability and performance volume losses and Reliability performance volume losses Metrics Charts** | 2s | 3s | | | | |
| **Defect Elimination KPI and Defect Elimination Metrics Charts** | 2s | 3s | | | | |
| **Critical Analysis KPI and Critical Analysis Metrics Charts** | 2s | 3s | | | | |
| **Maintenance Strategy KPI and Maintenance Strategy Metrics Charts** | 2s | 3s | | | | |
| **Action KPI and Action Metrics Chats** | 2s | 3s | | | | |

**Verification Aims: -**

- E2E Performance Team to run the system and E2E performance tests
- Component is considered as black box. Focus is on E2E workflows
- Test scope – Load Test, Scalability Test, Stress Test (Break Point), Longevity test
- Tests to focus on load, target KPIs (Response Time, Transaction Rate, CPU, Memory, Disk, Network etc.)
- Performance defects to be raised in Rally

| Label | # Samples | Average | Median | 90% Line | 95% Line | 99% Line | Min | Max | Error % | Throughput | Received KB/sec | Sent KB/sec |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  |  |  |  |  |  |

Performance will be measured by response time, throughput, reliability, and scalability

o Response time will be calculated by **Request + Process + Response Time** considering Network Latency

o Throughput will be calculated by **Number of Requests sent Per Sec**

o Reliability will be calculated by **Number of Errors with respect to Number of Requests**

o Scalability will be calculated by verifying Response, Throughput, Percentage of Errors by adding Resources to existing Setup

· Performance Test Life/Process

o Design test by specifying Number of Users, Amount of Data, and Load

o Dedicated Test Environment/Bed Setup for Performance

o Execute/Run test

o Analyse Results

o Optimize

*<repeat until we get expected performance from system>*

Tools to be used for performance testing:

· Verification using – *JMeter*

## 7.2 Reliability

*Availability of 99.72%, 1 day per quarter evaluates to 99.17% and 1 day per month evaluates to 96.71%.*

- We will have 2 instances of services running as per ops directions, which ensures availability even in case of catastrophic failure.
- We will run the instance in K8S which can auto scale as the demand needs which makes it more reliable in sudden surge of traffic.
- All messages received by cloud will have Http response codes to confirm the message is processed.
- Reliability will be calculated by Number of Errors with respect to Number of Requests
- We will execute test for24 hours/72 hours/ 1 week
- Verify execution results
- Calculate
- Probability = Number of Failing Cases/ Total Number of Cases under consideration
- %Availability = ((Total Time – Total Downtime) / Total Time) * 100
- Expected down time is 1 hour/month

## 7.3 Resiliency

Verify how the end-to-end workload performs under intermittent failure conditions and what's the recovery strategy

Fault injection: Test failures in combination and measure the recovery times

· Shut down API service

· Change Authorization

· Limit available system resources (such as RAM, Storage)

· Redeploy a VM or API Service

· Ingest fault data and verify how system behaves

- Rate limiting, throttling, and access control policies

· Check API response time, latency, TTFB/TTLB in various scenarios (in isolation and under load)

Peak load: Verify how the application behaves under real-world conditions

· Test for peak load and anticipated increase in peak load

· Increase number of concurrent users accessing the application

## 7.4 Metrics

**Availability metrics**

- Mean time between failures (MTBF) is the how long a component can reasonably expect to last between outages

MTBF= Total time of correct operation in a Period / Number of Failures

(PT) Total Production Time: TBD (1 Month)

(DT) Total Down Time: TBD

(F) Number of Failures count: TBD(1 month)

(UT) Uptime: PT – DT = TBD

MTBF: UT/F = TBD(1 month)

- Mean Time to Recover (MTTR) is the average time it takes to restore a component after a failure

total hours of downtime caused by system failures/number of failures

(DT) Total down time: TBD

(F) Number of failures count: 1TBD

MTTR: DT/F = TBD

**Recovery metrics**

- Recovery Time Objective (RTO) is the maximum acceptable time an application is unavailable after an incident i.e., how much data is one potentially prepared and willing to lose, worse case
- Recovery Point Objective (RPO) is the maximum duration of data loss that's acceptable during a disaster i.e., if/when the 'bad thing' happens, how much time does it take to be back up and running again

## 7.5 Security Requirements

All the apps will be accessible only via Appshell. Login/Authentication will be taken care by Appshell

- Users are authenticated by Appshell before entering the application. Only authenticated users are allowed to pass through
- Appshell provides a valid authentication token when apps need to communicate with the dependent services
- All data will be transferred over HTTPS, thus providing security during transport
- Set data is confidential/ highly confidential/ open. Specific to data - User Role.

  · Check that the API is designed according to correct security principles: deny-by-default, fail securely, least privilege principle, reject all illegal inputs, etc.

  · Positive: ensure API responds to correct authorization via all agreed auth methods – Bearer token, cookies, digest, etc. – as defined in spec

  · Negative: ensure API refuses all unauthorized calls

  · Role Permissions: ensure that specific endpoints are exposed to user based on role. API should refuse calls to endpoints which are not permitted for user's role

  · Data leaks: ensure that internal data representations that are desired to stay internal do not leak outside to the public API in the response payloads

  · Rate limiting, throttling, and access control policies

  · Check API response time, latency, TTFB/TTLB in various scenarios (in isolation and under load)

  . Vulnerability Scanning, License Review and Compliance Test

## 7.6 Logging and Monitoring: Tools need to be used under discussion

a. Debugging:

1. Application-level debugging: At browser Console

b. Logging: On Kubernetes/Kubernetes/another framework

1. Log levels: info, debug, warning, error
2. Log persistence for failover scenarios.

3. Logs to enable Debugging.
4. Error traceability with stack trace. (Logs traceability)

c. Monitoring:

1. Transaction (request) traceability: N/A
2. Cluster monitoring: N/A
3. Service pod level monitoring: Using K8 dashboard/dashboard/another framework

# 8. Test Setup

## 8.1 Test Environment setup

### 8.1.1 Setup requirements

- OS: Windows
- Devices: Laptop, Monitors
- Browsers: Chrome, Safari, and Edge with latest versions
- Resolution:

o Min Resolution expected is 1024x768 (Tablet)

o Laptop 1440 (Laptop & Above)

o Max is 1920 (desktop & Large Monitors)

### 8.1.2 Setup layout

The PD application is cloud deployment. Two connectors are deployed on-premise.

- Databases -> DataBricks
- CDP
- One PM APIs>> For End to End/Regression Test
- Comment Service
- Backend/Mock Services -> JSON Object in case of unavailability of actual data >> At story/Feature level Testing
- UI -> Summary Dashboard.
- Env -> QA, Dev, Pre-prod, Prod

## 8.2 Test data Strategy

Actual data which will appear from PIMS/ test the feature

add data with mock JSON data (in QA) via database.

One PM APIs will be tested with Mock Data:

| KPIs | Currently Tested | Max number of status/categories | Length of title of category (Spilling over to 2 or 3 lines?? After How many characters we need elipses?) |
|---|---|---|---|
| Defect Elimination Status Metrics | | | |
| RCA Distribution by Status | 3 | test with 10 | 30 chars |
| | | Number of RCAs more than 1000 per status?? | test with 1000 in individual statuses (4 digits) |
| RCA Distribution Per Criticality | 3 | No action | No Action |
| RCA Execution Status Per Criticality | Dependent on " RCA Distribution Per Criticality" number | No action | No Action |

| | | | |
|---|---|---|---|
| Criticality Analysis Metrics | | | |
| Criticality Distribution | 3 | 5 | 30 chars |
| Probability of Failure Distribution | 5 | 6 | 30 chars |
| Consequence Distribution | 5 | 6 | 30 chars |
| Asset Heatmap | dependent on " probability and consequence" number | 6 by 6 | 30 chars |
| Cumulative Risk Curve - By All Trains | 6 | No action | No Action |
| Cumulative Risk Curve - By Year | Removed | No action | No Action |
| Actions Metrics | | | |
| Actions by Action Priority | 3 | test with 10 | 50 chars |
| Actions by Action Status | 6 | test with 10 | 50 chars |
| Maintenance Strategy Metrics | | | |
| Application strategy vs Total Number of Assets | 4 (static) | 6 | 50 chars |
| Total Asset for Long Term Replacement Plan | | 6 | 50 chars |
| Long Term Replacement Plan Actions % Complete | | No action | |
| Long Term Replacement Plan Cost | 10 years (static) | Data missing for one or more years | |
| Long Term Replacement Plan Details | 10 on a page (without pagination) | 10 | character limit for "Drive for" column 50chars |

## 8.3 Test Environment Strategy

The following chart shows which types of testing are conducted in each environment. Whenever the changes are migrated from one environment to another, it is advisable to revalidate testing conducted in the prior environment in the new environment.

| | **Development Environment** | **QA Environment** | **Pre-Prod Environment** | **Production Environment** |
|---|---|---|---|---|
| **Unit Testing** | ✔ | ✔ | | |
| **Functional Testing** | | ✔ | | |
| **Regression Testing** | | ✔ | ✔ | |
| **Performance Testing** | | | ✔ | |
| **User Acceptance Testing** | | | ✔ | |
| **Production Validation** | | | | ✔ |
| **Sanity Testing** | | | ✔ | ✔ |

## 9.Automation Testing and Coverage

### 9.1 Automation coverage

- Regression test case automation coverage:
  - Target 70%
  - Max 100%
- Sanity test case automation coverage – 100%.
- End to end test scenarios automation (Can be part of Regression): 70 – 100%
- Performance and Longevity Testing - 100%

### 9.2 Automation tools and framework

- UI automation tool – Selenium and Cucumber based.
  Unified Framework.
- API automation tool – HTTP Client and Cucumber based.
  Unified Framework.
- Performance testing – JMeter based.
  Unified Framework plugin. (In-Progress): standard strategy for perf testing: vm or Jenkins with network latency, Etc.

### 9.3 Automation in CI

1. Jenkins's pipeline Job to be setup for Regression and Sanity test suite.
2. Sanity scheduled to run on Code deployment and on Demand.
3. Regression suite to run nightly.
4. Jenkins to run JMeter scripts.

## 10. Staffing and Training Needs

1. QA Staffing resource need = 2
2. Training:

a. QAFCO Strategy Knowledge

b. Knowledge of Cyborg framework.

c. Understanding of Rest Services.

d. Knowledge of Cordant user requirements.

## 11. Milestones and Schedule

## 12. Risks and Contingencies

## 13. Appendix

### 13.1 Definitions and Terms