***Sample Power BI Report***

***Design Document Fragment***

***Validation***

|  |  |
| --- | --- |
| *Date:* | *2025-04-15* |
| *Version:* | *0.5* |
| *Author:* | *Greg Philps* |

***Table of Contents***

[1. Introduction 3](#_Toc195588758)

[2. Scope of Work 3](#_Toc195588759)

[3. Workflow 3](#_Toc195588760)

[4. Issues 3](#_Toc195588761)

[5. Business Rules 3](#_Toc195588762)

[6. Data 3](#_Toc195588763)

[7. Reports 3](#_Toc195588764)

[8. Validation 3](#_Toc195588765)

[8.1. Validation Spreadsheet 3](#_Toc195588766)

[8.1.1. Setup and Teardown 4](#_Toc195588767)

[8.1.2. Status 4](#_Toc195588768)

[8.2. Development Testing 4](#_Toc195588769)

[8.2.1. Design Criteria 4](#_Toc195588770)

[8.2.2. Unit Testing 4](#_Toc195588771)

[8.2.3. Integration Testing 5](#_Toc195588772)

[8.3. Acceptance Testing 5](#_Toc195588773)

[8.3.1. Standards Testing 6](#_Toc195588774)

[8.3.2. Performance Testing 6](#_Toc195588775)

[8.3.3. Security Testing 6](#_Toc195588776)

[8.3.4. Signoff 6](#_Toc195588777)

[8.4. Production Use 6](#_Toc195588778)

[8.4.1. Smoke Testing 6](#_Toc195588779)

[8.4.2. Monitoring 7](#_Toc195588780)

[8.4.3. Issue Recreation 7](#_Toc195588781)

# Introduction

*… see sample fragment 01 …*

# Scope of Work

*… see sample fragment 01 …*

# Workflow

*… see sample fragment 02 …*

# Issues

*… see sample fragment 02 …*

# Business Rules

*… see sample fragment 02 …*

# Data

*… see sample fragment 03 …*

# Reports

*… see sample fragment 04 …*

# Validation

*The three main areas of manual validation (development testing, acceptance testing, and production testing) are described below along with the table (spreadsheet) of tests.*

## Validation Spreadsheet

*Describe the tests that will be used to design and validate the system (report), including group (ID and name), test (ID, name, notes, priority, expected results, and tolerance/allowed variance), procedure (setup [steps], procedure [steps], actual results, and teardown [steps]), date (date and time performed and by whom), and status (pass, fail, deferred).*

*The tests are itemized is a separate spreadsheet.*

<https://github.com/alexbadiu-insightsinmotion/PBI-Documentation/blob/main/Design%20Document%20-%20Sample%20Validation%20Spreadsheet%20-%20V0.5.xlsx>

### Setup and Teardown

*Describe the steps necessary to change the systems from its initial state, and, once the test is complete, to return the system to its initial state (so the system is prepared for the next test).*

*The steps should be clearly stated (easily repeatable) as they may be performed many times.*

### Status

*Record the date the test was conducted, the name of the person who conducted the test, and the status of the test. All tests that were completed must have these 3 items, while there should be explanatory notes describing the cause/situation for any tests that were deferred.*

*This information, when viewed along with the test priorities, will allow fact-based evaluation of whether a system (report) passed the validation and is ready for deployment to the production environment.*

## Development Testing

*Development testing will be ongoing throughout the development period.*

### Design Criteria

*Provide the test cases to the developers during the development period to ensure that the criteria that will be used during acceptance testing are used to develop the solution (each test case should describe a scenario and expected behaviour, including normal, alternate, and exception paths where appropriate).*

*One of the best ways to increase project velocity is to write the test cases as early as possible in the development cycle; the developers can then take the test criteria into account when designing and iterating the solution. (If the test cases are not developed until acceptance testing, it decreases the likelihood that the solution will perform as desired and meet expectations.)*

### Unit Testing

*Conduct unit testing on individual system components throughout the development period.*

### Integration Testing

*Conduct integration testing on the system as a whole throughout the development period and will verify the connections between modules (reports) and data sources. (For reports that serve multiple audiences, integration testing should be conducted for each audience [e.g., IT, administrators, users, external, etc.].)*

## Acceptance Testing

*Acceptance testing will be conducted by persons different than the developers who will ideally be those responsible (i.e., the business users) for ensuring the reports meet business acceptance criteria. This is also commonly referred to as UAT, or User Acceptance Testing.*

*If acceptance testing is not conducted and one instead relies on developer testing, then the impacts and risks must be accepted, such as:*

* *The DEV environment may not be reflective of the PROD environment (e.g., workstation specifications, network [geographical] performance, etc.)*
* *Developers will test their understanding of how the system (report) works, which is not necessarily how the system (report) should work*
  + *To mitigate this impact, it is imperative that the report specifications are exhaustive and handle all normal, alternate, and exception flows*
* *Developers often have elevated permissions in the DEV environment and system (report) security may not be adequately verified*
* *The DEV environment often has reduced data volume, variety, and freshness compared to the PROD environment*

*Acceptance testing should be conducted in a dedicated testing environment that:*

* *Is configured in the same fashion as the production environment.*
* *Has representative data variety, freshness, volume, and access:*
  + *To give confidence that testing will reflect use in the production environment*
  + *To leverage the business knowledge of the testers (business users) as to whether values are correct or within tolerance ranges*

*In addition, appropriate accounts need to be available for testers to utilize during testing to confirm solution security.*

### Standards Testing

*Verify the corporate standards for theming have been applied (e.g., fonts, colours, shadings, sidebar navigation, etc.). The theme was described above.*

<<< INSERT LINK TO REPORTS FRAGMENT >>>

*Conduct standards testing with a fixed configuration/environment (e.g., Windows 11 Home 64-bit, Intel I5 processor, 8 GB RAM, browser used = Microsoft Edge version 1.35.0, network normal usage level [e.g., not Between 9-11 am ET, not overnight (11pm-7am)], etc.).*

### Performance Testing

*Even though a system (report) may exhibit all required features and meet all relevant standards, it may not be used by end users (and hence not solve its’ intended purpose) if it is too slow. Performance must be at an acceptable level before a deployment decision is made, and this must be verified.*

*For performance testing to be relevant, then, as said above, the testing environment needs to be representative of the production environment in data variety, freshness, volume, and access, and should have load and network performance similar to production.*

### Security Testing

*For reports that serve multiple audiences, acceptance testing should be conducted using appropriate accounts for each target audience (e.g., IT, administrators, users, external, etc.). This testing should ensure that only appropriate records are displayed to any group (e.g., executives see all records, sales managers see only records for their territories, salespeople see only their own sales records, etc.).*

### Signoff

*Each individual acceptance test should be signed-off and dated by the person who conducted the test and assigned a status.*

*Once all tests have a signoff, an overall go/no-go decision can be made by the business owner on whether the system is ready for deployment to the production environment.*

## Production Use

### Smoke Testing

*Smoke testing should be performed after deployment to the production environment to confirm the access to the report, that is works as intended, and that it is ready for use.*

*Smoke testing refers to light or ad-hoc testing of the major features and is often used to identify red flags or hot spots for resolution before widespread use.*

*Smoke testing is a reference to the old adage, \*“where there’s smoke there’s fire”*

### Monitoring

*Testing is not “done” when a report is deployed to the production environment; rather, ongoing monitoring should regularly be conducted to confirm that the report is functioning as intended, and may include such areas as:*

* *Data Refresh (is the scheduled refresh happening as intended?)*
* *Functionality (are the inherent features functioning as intended?)*
* *Access (is the security properly applied for internal audience(s) and external users [if appropriate]?)*

*A superset or subset of these areas will apply to different situations and will be subject to organizational monitoring criteria.*

*Of note in all cases, though, is the credentials used for the monitoring: if the monitor is an Administrator and uses her/his account (or a service account) to monitor the workspace/semantic model/report/app, then elevated permissions may not be reflective of audience use. An account with appropriate credentials and permissions should especially be used if there are access or security restrictions for external users.*

### Issue Recreation

*Often manual testing will not be exhaustive, and it is also not unheard of that users will encounter issues and file bug reports. The user often is more focussed on identifying that an issue exists, rather than including a complete set of steps to recreate the issue. Also, just because an issue exists does not necessarily mean that it needs to be fixed immediately.*

*Having a user or team accurately and completely and document an issue is very helpful in being able to recreate the issue (e.g., account, workstation setup, browser used, etc.) will contribute to the severity, priority, and level-of-effort estimation being properly evaluated and prioritized. Addressing the issue can then be either scheduled or added to any known issues.*

*-- end of fragment*