Test Plan Document

## Supplier app

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test type** | | | | |
|  | **Unit/unitary** | **Performance** | **Volume** | **Regression** | **User acceptance** |
| **Test type requirement** | Required | Optional and recommended | Optional but not recommended | Required | Required |
| **System environment** | Development | Quality | --- | Quality | Quality |
| **Team** | Project team | Project team or Both | --- | Project team | Both |
| **Rationale** | \*Note 1 | \*Note 2 | \*Note 3 | \*Note 4 | \*Note 5 |

**\*Note 1**: After building/configuring, you must check if the work produces the expected results. Unit testing takes place in the development system. Only if it is successful the new function is deployed in a test environment (e.g., Quality). Testing is not performed in training or production systems.

**\*Note 2**: The app has been developed to reduce the time spent on each buying event; hence, performance is very critical. During user acceptance testing, performance verification is implicit, but a performance test can involve simulations with multiple users concurrently accessing the system. Performance testing can involve both teams or just one and can even be achieved with automation tools. The relevant part is creating a scenario where multiple requests will be produced simultaneously.

\***Note 3**: The application description states that no large amounts of data are involved.

\***Note 4**: This app extends the functionality of the existing application. The output from this app will be an input to existing systems and thus a close integration with existing systems. Therefore, regression testing is required.

\***Note 5**: After unit testing, the new programs and configurations are applied in a test environment where end users do acceptance testing. This can also be done with the participation of project team members. This is because the test environment can have more data and hardware resources. It only contains configurations already part of unit testing or existing in the previous solution.

## Customer engagement app

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test type** | | | | |
|  | **Unit/unitary** | **Performance** | **Volume** | **Regression** | **User acceptance** |
| **Test type requirement** | Required | Optional and recommended | Optional but not recommended | Required | Required |
| **System environment** | Development | Quality | --- | Quality | Quality |
| **Team** | Project team | Project team or Both | --- | Project team | Both |
| **Rationale** | \*Note 1 | \*Note 2 | \*Note 3 | \*Note 4 | \*Note 5 |

**\*Note 1**: After building/configuring, you must check if the work produces the expected results. Unit testing takes place in the development system. Only if it is successful the new function is deployed in a test environment (e.g., Quality). Testing is not performed in training or production systems.

**\*Note 2**: The app is the first customer touch point through which sales are generated. Hence performance is very critical. During user acceptance testing, performance verification is implicit, but a performance test can involve simulations with multiple users concurrently accessing the system. Performance testing can involve both teams or just one and can even be achieved with automation tools. The relevant part is creating a scenario where multiple requests will be produced simultaneously.

\***Note 3**: The application description states that no large amounts of data are involved.

\***Note 4**: This app extends the functionality of the existing application. The output from this app will be an input to existing systems and thus a close integration with existing systems. Therefore, regression testing is required.

\***Note 5**: After unit testing, the new programs and configurations are applied in a test environment where end users do acceptance testing. This can also be done with the participation of project team members. This is because the test environment can have more data and hardware resources. It only contains configurations already part of unit testing or existing in the previous solution.

## Sales Data Analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test type** | | | | |
|  | **Unit/unitary** | **Performance** | **Volume** | **Regression** | **User acceptance** |
| **Test type requirement** | Required | Optional but not recommended | Optional but not recommended | Not required | Required |
| **System environment** | Development | --- | --- | --- | Quality |
| **Team** | Project team | --- | --- | --- | Both |
| **Rationale** | \*Note 1 | \*Note 2 | \*Note 3 | \*Note 4 | \*Note 5 |

**\*Note 1**: After building/configuring, you must check if the work produces the expected results. Unit testing takes place in the development system. Only if it is successful the new function is deployed in a test environment (e.g., Quality). Testing is not performed in training or production systems.

**\*Note 2**: The application description (attributes/notes) does not point to any requirements that might lead to performance problems. No goals are set that require a KPI to be achieved.

\***Note 3**: The application description (attributes/notes) clearly states that no large amounts of data are involved.

\***Note 4**: No existing functions are to be changed. No output that might impact subsequent automated activities will be generated (human review of results takes place.) Functions to be replaced still exist in the system and can be executed for double-checking or assessing the quality of the new analytical planning tools.

\***Note 5**: After unit testing, the new programs and configurations are applied in a test environment where end users do acceptance testing. This can also be done with the participation of project team members. This is because the test environment can have more data and hardware resources. It only contains configurations already part of unit testing or existing in the previous solution.

## Operational Data Analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test type** | | | | |
|  | **Unit/unitary** | **Performance** | **Volume** | **Regression** | **User acceptance** |
| **Test type requirement** | Required | Optional but not recommended | Optional but not recommended | Not required | Required |
| **System environment** | Development | --- | --- | --- | Quality |
| **Team** | Project team | --- | --- | --- | Both |
| **Rationale** | \*Note 1 | \*Note 2 | \*Note 3 | \*Note 4 | \*Note 5 |

**\*Note 1**: After building/configuring, you must check if the work produces the expected results. Unit testing takes place in the development system. Only if it is successful the new function is deployed in a test environment (e.g., Quality). Testing is not performed in training or production systems.

**\*Note 2**: The application description (attributes/notes) does not point to any requirements that might lead to performance problems. No goals are set that require a KPI to be achieved.

\***Note 3**: The application description (attributes/notes) clearly states that no large amounts of data are involved.

\***Note 4**: No existing functions are to be changed. No output that might impact subsequent automated activities will be generated (human review of results takes place.) Functions to be replaced still exist in the system and can be executed for double-checking or assessing the quality of the new analytical planning tools.

\***Note 5**: After unit testing, the new programs and configurations are applied in a test environment where end users do acceptance testing. This can also be done with the participation of project team members. This is because the test environment can have more data and hardware resources. It only contains configurations already part of unit testing or existing in the previous solution.

## Mill Equipment Failure Prediction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test type** | | | | |
|  | **Unit/unitary** | **Performance** | **Volume** | **Regression** | **User acceptance** |
| **Test type requirement** | Required | Optional but not recommended | Optional but not recommended | Not required | Required |
| **System environment** | Development | --- | --- | --- | Quality |
| **Team** | Project team | --- | --- | --- | Both |
| **Rationale** | \*Note 1 | \*Note 2 | \*Note 3 | \*Note 4 | \*Note 5 |

**\*Note 1**: After building/configuring, you must check if the work produces the expected results. Unit testing takes place in the development system. Only if it is successful the new function is deployed in a test environment (e.g., Quality). Testing is not performed in training or production systems.

**\*Note 2**: The application description (attributes/notes) does not point to any requirements that might lead to performance problems. No goals are set that require a KPI to be achieved.

\***Note 3**: The application description (attributes/notes) clearly states that no large amounts of data are involved.

\***Note 4**: No existing functions are to be changed. No output that might impact subsequent automated activities will be generated (human review of results takes place.) Functions to be replaced still exist in the system and can be executed for double-checking or assessing the quality of the new analytical planning tools.

\***Note 5**: After unit testing, the new programs and configurations are applied in a test environment where end users do acceptance testing. This can also be done with the participation of project team members. This is because the test environment can have more data and hardware resources. It only contains configurations already part of unit testing or existing in the previous solution.

## Resource Optimization Reports

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test type** | | | | |
|  | **Unit/unitary** | **Performance** | **Volume** | **Regression** | **User acceptance** |
| **Test type requirement** | Required | Optional but not recommended | Optional but not recommended | Not required | Required |
| **System environment** | Development | --- | --- | --- | Quality |
| **Team** | Project team | --- | --- | --- | Both |
| **Rationale** | \*Note 1 | \*Note 2 | \*Note 3 | \*Note 4 | \*Note 5 |

**\*Note 1**: After building/configuring, you must check if the work produces the expected results. Unit testing takes place in the development system. Only if it is successful the new function is deployed in a test environment (e.g., Quality). Testing is not performed in training or production systems.

**\*Note 2**: The application description (attributes/notes) does not point to any requirements that might lead to performance problems. No goals are set that require a KPI to be achieved.

\***Note 3**: The application description (attributes/notes) clearly states that no large amounts of data are involved.

\***Note 4**: No existing functions are to be changed. No output that might impact subsequent automated activities will be generated (human review of results takes place.) Functions to be replaced still exist in the system and can be executed for double-checking or assessing the quality of the new analytical planning tools.

\***Note 5**: After unit testing, the new programs and configurations are applied in a test environment where end users do acceptance testing. This can also be done with the participation of project team members. This is because the test environment can have more data and hardware resources. It only contains configurations already part of unit testing or existing in the previous solution.