Test Plan Document

|  |  |
| --- | --- |
|  | Testing and monitoring are critical aspects of delivering a new system. In this task, you will identify which test types are relevant for each mobile application or analytical tool: unitary, performance, volume, regression, and/or user acceptance. If a test type is required, you will also identify the system environment in which the testing will take place and by which team. Lastly, you will document your rationale for your decision on test type requirement, system environment, and team. |

## Farmer Agenda (Task 1 – Activity 1)

|  |  |
| --- | --- |
|  | *Document the test plan for the Farmer Agenda functionality.* |

Fill in the table below. Use one of the following values for each row:

* Test type requirement: Required; Not required; Optional and recommended; Optional but not recommended
* System environment (*Fill this only if you mentioned Test type requirement as ‘Required’*): All systems; Sandbox; Evaluation; Development; Quality; Training; Production
* Team (*Fill this only if you mentioned Test type requirement as ‘Required’*): Project team; Customer team; Both
* For rationale, provide free text mentioning why you chose a specific input

Table 1.1 Farmer Agenda Functionality

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Farmer Agenda** | **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 need to check if the work is producing 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 requires 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/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 since test environment can have more data, more hardware resources, and only contain configurations that were already part of unit testing or existed in the previous solution. Besides unit testing, all other tests take place in test environments (quality, consolidation, test, pre-production, integration.)

## Farmer Bookkeeper (Task 1 – Activity 2)

|  |  |
| --- | --- |
|  | *Document the test plan for the Farmer Bookkeeper functionality.* |

Fill in the table below. Use one of the following values for each row:

* Test type requirement: Required; Not required; Optional and recommended; Optional but not recommended
* System environment (*Fill this only if you mentioned Test type requirement as ‘Required’*): All systems; Sandbox; Evaluation; Development; Quality; Training; Production
* Team (*Fill this only if you mentioned Test type requirement as ‘Required’*): Project team; Customer team; Both
* For rationale, provide free text mentioning why you chose a specific input

Table 1.2 Farmer Bookkeeper Functionality

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Farmer Bookkeeper** | **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 need to check if the work is producing 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 requires 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/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 since test environment can have more data, more hardware resources, and only contain configurations that were already part of unit testing or existed in the previous solution. Besides unit testing, all other tests take place in test environments (quality, consolidation, test, pre-production, integration.)

## Crop Planner (Task 1 – Activity 3)

|  |  |
| --- | --- |
|  | *Document the test plan for the Crop Planner functionality.* |

Fill in the table below. Use one of the following values for each row:

* Test type requirement: Required; Not required; Optional and recommended; Optional but not recommended
* System environment (*Fill this only if you mentioned Test type requirement as ‘Required’*): All systems; Sandbox; Evaluation; Development; Quality; Testing; Production
* Team (*Fill this only if you mentioned Test type requirement as ‘Required’*): Project team; Customer team; Both
* For rationale, provide free text mentioning why you chose a specific input

Table 1.3 Crop Planner Functionality

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Crop Planner** | **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 need to check if the work is producing 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 requires 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/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 since test environment can have more data, more hardware resources, and only contain configurations that were already part of unit testing or existed in the previous solution. Besides unit testing, all other tests take place in test environments (quality, consolidation, test, pre-production, integration.)

## Fleet Planner (Task 1 – Activity 4)

|  |  |
| --- | --- |
|  | *Document the test plan for the Fleet Planner functionality.* |

Fill in the table below. Use one of the following values for each row:

* Test type requirement: Required; Not required; Optional and recommended; Optional but not recommended
* System environment (*Fill this only if you mentioned Test type requirement as ‘Required’*): All systems; Sandbox; Evaluation; Development; Quality; Training; Production
* Team (*Fill this only if you mentioned Test type requirement as ‘Required’*): Project team; Customer team; Both
* For rationale, provide free text mentioning why you chose a specific input

Table 1.4 Fleet Planner Functionality

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fleet Planner** | **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 need to check if the work is producing 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 requires 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/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 since test environment can have more data, more hardware resources, and only contain configurations that were already part of unit testing or existed in the previous solution. Besides unit testing, all other tests take place in test environments (quality, consolidation, test, pre-production, integration.)

## Shipments Planner (Task 1 – Activity 5)

|  |  |
| --- | --- |
|  | *Document the test plan for the Shipments Planner functionality.* |

Fill in the table below. Use one of the following values for each row:

* Test type requirement: Required; Not required; Optional and recommended; Optional but not recommended
* System environment (*Fill this only if you mentioned Test type requirement as ‘Required’*): All systems; Sandbox; Evaluation; Development; Quality; Training; Production
* Team (*Fill this only if you mentioned Test type requirement as ‘Required’*): Project team; Customer team; Both
* For rationale, provide free text mentioning why you chose a specific input

Table 1.5 Shipments Planner Functionality

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Shipments Planner** | **Test type** | | | | |
|  | **Unit/unitary** | **Performance** | **Volume** | **Regression** | **User acceptance** |
| **Test type requirement** | Required | Optional and recommended | Optional but not recommended | Not required | Required |
| **System environment** | Development | Quality | --- | --- | Quality |
| **Team** | Project team | Project team or Both | --- | --- | Both |
| **Rationale** | \*Note 1 | \*Note 6 | \*Note 3 | \*Note 4 | \*Note 5 |

**\*Note 1**: After building/configuring, you need to check if the work is producing 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 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/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 since test environment can have more data, more hardware resources, and only contain configurations that were already part of unit testing or existed in the previous solution. Besides unit testing, all other tests take place in test environments (quality, consolidation, test, pre-production, integration.)

**\*Note 6**: Any analytical tool for real-time decision support should be tested for performance. During user acceptance testing a verification of performance 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 at the same time.

## Farming Planner (Task 1 – Activity 6)

|  |  |
| --- | --- |
|  | *Document the test plan for the Farming Planner functionality.* |

Fill in the table below. Use one of the following values for each row:

* Test type requirement: Required; Not required; Optional and recommended; Optional but not recommended
* System environment (*Fill this only if you mentioned Test type requirement as ‘Required’*): All systems; Sandbox; Evaluation; Development; Quality; Training; Production
* Team (*Fill this only if you mentioned Test type requirement as ‘Required’*): Project team; Customer team; Both
* For rationale, provide free text mentioning why you chose a specific input

Table 1.6 Farming Planner Functionality

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Farming Planner** | **Test type** | | | | |
|  | **Unit/unitary** | **Performance** | **Volume** | **Regression** | **User acceptance** |
| **Test type requirement** | Required | Optional and recommended | Optional but not recommended | Not required | Required |
| **System environment** | Development | Quality | --- | --- | Quality |
| **Team** | Project team | Project team or Both | --- | --- | Both |
| **Rationale** | \*Note 2 | \*Note 6 | \*Note 4 | \*Note 1 | \*Note 3 |

**\*Note 2**: After building/configuring, you need to check if the work is producing 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 6**: Any analytical tool for real-time decision support should be tested for performance. During user acceptance testing a verification of performance 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 at the same time.

\***Note 4**: The application description (attributes/notes) clearly states that no large amounts of data are involved.

\***Note 1**: 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 3**: After unit testing, the new programs/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 since test environment can have more data, more hardware resources, and only contain configurations that were already part of unit testing or existed in the previous solution. Besides unit testing, all other tests take place in test environments (quality, consolidation, test, pre-production, integration.)