|  |  |  |  |
| --- | --- | --- | --- |
| Checklist for Test cases review during the Test Preparation Phase | | | |
| **S.No.** | **Description** | **Y/N/NA** | **Remarks** |
| 1 | Has the correct template been used? | Y |  |
| 2 | Have the following details been filled up correctly? Requirement reference, Test script description, Author’s name, Date created, Setup Procedure, Pre-requisites – where applicable. | Y |  |
|  | Have the Test conditions (scenarios) been identified along with the Risk factor, if applicable? | NA |  |
|  | Have all the scenarios specified in the requirement – both explicit and implicit, been converted into Test conditions? | Y |  |
|  | Have the related areas that could possibly be affected by the implementation of the requirement been identified and included in the test cases? (Identify the impact areas and check with the test cases) | Y |  |
|  | Has equivalence partitioning been done? Have all the classes of the domain been identified correctly? | Y |  |
|  | Has the test data set, if required been generated appropriately? | Y |  |
|  | Have the boundary values, special values and invalid values been identified and included in the Test data set? | Y |  |
|  | Has the Test data been embedded into the test cases? | Y |  |
|  | Have the required negative scenarios been identified in the test conditions? | Y |  |
|  | Have the steps been correctly given in appropriate sequence for each test scenario? Steps/Actions should state very clearly the sequence of actions to be carried out on the system by the user. All statements should be definite. Ensure that terms like “If”, “In case” etc are not used. | Y |  |
|  | Have the Expected Results been identified correctly?   * Expected Results should clearly state how the system should respond to the user actions given in each step/action. * Ensure that too many things are not included to be verified under one expected output. * Ensure that separate cases are written for multiple verifications of the application’s behavior. * Vague statements like “Appropriate message/value/screen” etc, should not be part of expected result. Every detail should be clearly spelt out. | Y |  |
|  | Are all the statements free from grammatical errors? | Y |  |
| **Reviewed by:Rohini Manjappa** | | | |
| **Reviewed on:06 September 2021** | | | |

During the **test execution** phase, doing a review after the cases are executed is very important though not many follow this process. At this stage, ensuring that the test cases have been actually run successfully and that the results have been documented clearly is vital. In this phase, apart from the items that have been listed in the following checklist, additional project specific checks can be included.

**Areas of focus**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Execution Phase** | | | |
| **S.No.** | **Description** | **Y/N/NA** | **Remarks** |
|  | Have the Actual Results been updated for each of the steps? Has the actual result been documented for a failed step and for its subsequent re-runs? | NA |  |
|  | Have all the steps been executed successfully? Every failed step should have retest details or some disposition if it is not fixed. | NA |  |
|  | Have the defect details like Defect id, description etc. been given for a failed step? | NA |  |
|  | Has the reason for the failure been recorded? (For example, invalid input data, new functionality not tested before, existing problem) | Y |  |
|  | Did a peer recreate the defect before logging it in the Defects database? Have these details been documented? | NA |  |
|  | Has the defect been retested and if so, have the retest details and the result documented along with the date on which the retest was done? | NA |  |
|  | Have the Execution details like executed by and executed date, been filled up correctly? | NA |  |
|  | Have the results from different environments (Browsers, for example) been recorded? (If applicable) | NA |  |
|  | Have the metrics related to the test cases been updated in all applicable metrics documents? (Number of Test cases prepared, executed, Number of test case executions with defects, Total Number of defects etc.) | NA |  |
|  | Are all the statements free from grammatical errors? |  |  |
| **Reviewed by:** | | | |
| **Reviewed on:** | | | |

During the **Test Reporting** phase, it would help to ensure that all the required documents are prepared, metrics are collated and all the project specific formalities are completed. Since this would be different for different organizations and projects, a checklist has not been provided here. But the team should try to come up with one, that suits the project needs and use it during the Reporting phase.

,

**Most Common Test cases Review Defects**

When these checklists are used diligently and defects are identified, it is advised to categorize the defects into any one of the following categories:

* Incomplete Test Cases
* Missing negative test cases
* No Test Data
* Inappropriate/Incorrect Test data
* Incorrect Expected behavior
* Grammatical errors
* Typos
* Inconsistent tense/voice
* Incomplete results/number of test runs
* Defect details not updated
* Changes to requirements not updated in Test case

The above list was prepared after analyzing the review data for a period of two years. This exercise would help in identifying the defects that are repeatedly reported so that preventive action can be taken to avoid such repetition. By doing this, the quality of the deliverables can be improved and the amount of time spent on fixing review defects can be reduced.

**Common Review Defects, their Causes and suggested Preventive Measures**

The most common defects reported during the review of test cases and the causes for each of them along with the suggested preventive actions are tabulated below. Depending on what the review defects trend is in the project, these actions can be implemented accordingly.

|  |  |  |
| --- | --- | --- |
| **Defects** | **Causes** | **Preventive Actions** |
| Incomplete Test Cases  Missed negative test cases | * Inadequate Functionality Knowledge * Inadequate Testing experience * Too Much or too little information in the requirements * Oversight * Changes to requirements after preparation of test cases | * Provide application specific Training * Provide training in Testing concepts and methods * Do a thorough Requirements review before test case preparation * Use Test Case Review Checklist for all reviews * Periodic Requirements review before submitting test cases for review |
| No Test Data  Inappropriate/Incorrect Test data | * Inadequate information in the Requirements * Inadequate Functionality Knowledge | * Thorough Requirements review before test case preparation * Provide application specific Training |
| Incorrect Expected behavior | * Inadequate Functionality Knowledge * Changes to requirements after preparation of test cases | * Provide application specific Training * Do periodic Requirements review before submitting test cases for review |
| Documentation Errors  Grammatical errors  Typos  Inconsistent tense/voice  Incomplete results/number of test runs  Defect details not updated | * Oversight * Not attaching importance to grammar checks * Not updating the test cases with the actual results and defect details after every run. | * Do a thorough spell check before submitting the document for review * Provide training in written communication * Discuss and reiterate the process of updating the test cases for every test run |
| Changes to requirements not updated in Test case | * Changes to requirements not communicated by business. * Not checking the comments in the requirements document periodically | * Request the Business Analysts to pass on the information about changes to requirements. * Do periodic Requirements review and update the test cases, when required before submitting test cases for review |

Possessing the knowledge of what could go wrong and how it could be prevented will enable the team to prepare better test cases. This, coupled with the usage of the review checklists in each phase would definitely pave way to achieve effective and efficient review of test cases.