Notepad

TEST PLAN

Version 1.0



CONTENTS

[Purpose 3](#_Toc493505524)

[Test plan identifier 3](#_Toc493505525)

[Project overview 3](#_Toc493505526)

[Test items 3](#_Toc493505527)

[Features to be tested 3](#_Toc493505528)

[Features not to be tested 3](#_Toc493505529)

[Approach 4](#_Toc493505530)

[Pass/Fail criteria 4](#_Toc493505531)

[Test Deliverables 4](#_Toc493505532)

[Test Tasks 4](#_Toc493505533)

[Environmental needs 4](#_Toc493505534)

[Responsibilities 4](#_Toc493505535)

[Risks and contingencies 5](#_Toc493505536)

[Approvals 5](#_Toc493505537)

# **Purpose**

This test plan describes the testing approach and overall framework that will drive the testing of Notepad Version 6.1 on Microsoft Windows 7. It identifies the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for the tasks, and the risks associated with this plan.

# **Test plan identifier**

NN\_PP 01

# **Project overview**

Notepad is a simple [text editor](https://en.wikipedia.org/wiki/Text_editor) for [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) and a basic text-editing program which enables computer users to create documents. Notepad is a common text-only (plain text) editor. The resulting files—typically saved with the .txt extension—have no format tags or styles, making the program suitable for editing system files.

# **Test items**

* Each main menu
* Each sub menu
* Resize functions
* Text inputs
* Hot keys functionalities
* Click buttons
* Context menus
* Edit abilities

# **Features to be tested** (the features are to be tested in the following order)

1. Create a new file
2. Open a new file
3. Save a new file
4. Save an existing file
5. Save in different formats
6. Page parameters
7. Print the text
8. Edit text
9. Copy and paste the text
10. Cut out the text
11. Delete the text
12. Search for a pattern
13. Replace a pattern
14. Cancel the latest actions
15. Insert time and date
16. Highlight everything
17. Format the text
18. Change the view
19. View the documentation

# **Features not to be tested**

All the features will be tested.

**Approach**

Functional and Non-functional testing (Black Box testing)

With the help of functional testing the program will be tested from the user's point of view.

With the help of non-functional testing the program will be verified by such criteria: performance testing; usability testing; failover and recovery testing; configuration testing; localization testing.

# **Pass/Fail criteria**

We can start performing tests as soon as enough test cases or checklists are created. Enough here may be defined as the requirements coverage measured via RTM ( 80%). Then, it will be time to proceed from testing to production.

# **Test Deliverables**

* Test plan
* Test case report
* Bug report
* Test summary report

# **Test Tasks**

1. Identify automation tools for carrying out functional and non-functional testing.
2. Create test requirements and test cases.

# **Environmental needs**

* Windows OS
* Minimum hardware configuration required to run desktop application
* Minimum hardware configuration required to run automation tool.

# **Responsibilities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Role | Responsibility | Name | Contact info |
|  | Project Manager | Reviews the content of the Test Plan, Test Strategy and Test Estimates. |  |  |
|  | Test Lead | Acknowledges the completion of a section within a cycle. Gives the OK to start next level of testing. Facilitates defect communications between testing team and technical / development team. |  |  |
|  | Testing Team | Develops test conditions, test cases, expected results, and execution scripts. Performs execution and validation. Identifies, document and prioritize defects according to the guidance provided by the Test lead. Re-tests after software modifications have been made according to the schedule. Prepares testing metrics and provides regular status. |  |  |
|  | Development Team | Reviews testing deliverables (test plan, cases, scripts, expected results, etc.) and provides timely feedback. Assists in the validation of results (if requested). Supports the development and testing processes being used to support the project. Certifies correct components have been delivered to the test environment at the points specified in the testing schedule. Keeps project team and leadership informed of potential software delivery date slips based on the current schedule. Defines processes/tools to facilitate the initial and ongoing migration of components. Conducts first line investigation into execution discrepancies and assist test executors in creation of accurate defects. Implements fixes to defects according to schedule. |  |  |

# **Risks and contingencies**

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Prob. | Impact | Mitigation Plan |
| Testing start delayed | high | high | The testing team can control the preparation tasks (in advance) and the early communication with involved parties.  Some buffer has been added to the schedule for contingencies, although not as much as best practices advise. |
| Not enough resources | medium | high | Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing. |
| Defects are found at a late stage of the cycle or at a late cycle | medium | high | Defect management plan is in place to ensure prompt communication and fixing of issues. |
| Scope completely defined | medium | medium | Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing. |

# **Approvals**

|  |  |
| --- | --- |
| Position | Signature |
| Project Team Leader |  |