ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 1, Issue 6, October 2016

Study of Different Pragmatic Approaches and Testing Strategies for Stand Alone/ Desktop Application

[1] Prof. Vaishali Jawale [2] Dr. K. Nirmala Kumaraswamy
[1] Assistant Professor [2] Director
[1] ASM's Institute of Business Management and Research, Chinchwad, Pune
[2] Dr. DY Patil Institute of Master of Computer Applications, Akurdi, Pune

Abstract: **- Test strategy is one of the most important documents for QA team. These strategies are changes as per the changes in functionality of application. It may vary according to Application type also . Software application development began with desktop applications, which could be used on standalone machines only. In this research paper we have studied various pragmatic approaches and testing strategies to test the standalone/ Desktop based application. The major contribution of this paper includes its insightful discussion on the various strategies/checklist designed to test the standalone/desktop application.**

Index Terms:-- **Desktop Application, Checklist, Testing Strategies.**

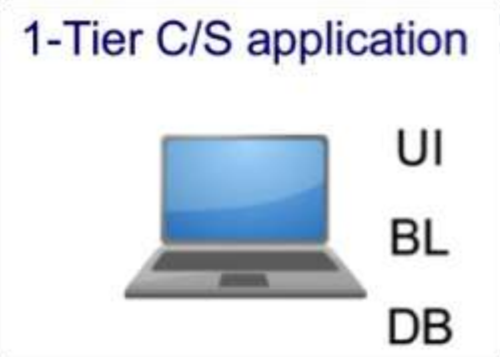
I. INTRODUCTION

Standalone/Desktop Application:

The standalone applications are the applications that can run only on the one system on which it is being installed. The application which being developed using c and c++ are the standalone application because they do not form platform independency. Applications like MS Excel, MS Word, and Outlook. Some desktop applications made by technologies like HTML and JS which allow the developers to write code. Thus the desktop applications are also made of these technologies.

If you choose standalone installation, this means the database and all the information is stored on the local computer and no server is needed. For installing a standalone application you don't need an internet connection. All the files will be included the setup file itself. So the size of the file will be greater compared to client server application.

The installed application, having user interface, business layer and database on the same machine without the need of a network connection. As an example, think of Microsoft Money (personal finance management software). This is a single tier application that runs in a single system driven by a single user. [1]



This standalone application, which consists of a call to a single function, forms the basis of our distributed application. The function, **HelloProc**, is defined in its own source file so that it can be compiled and linked with either a standalone application or a distributed application.[2]

```
/* file hellop.c */ #include <stdio.h> #include <windows.h>

void HelloProc(char * pszString)
{printf("%s\n", pszString);}

/* file: hello.c, a stand-alone application */ #include
"hellop.c" void main(void)
{    char * pszString = "Hello, World";
  HelloProc(pszString);}
```

II. CHARACTERISTICS OFSTAND ALONE APPLICATION:

- ❖ Application which run on single system /computer or workstation.
- ❖ There are a vast number of different architectures for stand-alone applications. Some can be quite similar to a web hosted application; others are quite different.
- ❖ These are desktop driven application.
- ❖ The environment is the user machine.
- ❖ We can have a total control over the desktop applications and protect it from various vulnerabilities.
- ❖ The programmer has full control (within the capabilities of the operating system and with support of a broad selection of UI frameworks) over the user interface.
- ❖ The business logic layer and data layer may reside on the same machine or on a remote server.
- ❖ In Desktop applications we test application features like GUI, backend and load.
- ❖ If one or more remote servers are involved in the solution, many different remote communication protocols can be selected from. There's no requirement to use HTTP over TCP/IP (and in fact, more efficient protocols are often selected).
- ❖ Desktop Application there is only one user accessing it and the application may or may not require authentic access.

III. CHECKLIST FOR STANDALONE /DESKTOP APPLICATION TESTING:

Desktop applications run on work stations and personal computers. When testing desktop applications, we are focusing on a specific environment. Testing will encompass categories such as GUI, functionality, Load, and back-end results (i.e DB).

Level 1 - User Interface Testing (GUI Testing):

a. Content wording used in the web pages should be correct.

b. Wrap-around should occur properly.

c. Instructions used in the application should be correct (i.e. if you follow each instruction does the expected result occur?

d. Image spacing – To verify that images are displaying properly with text.

Level 2 - Functional Testing

Functional Software Testing for Desktop Applications is a type of black box testing that explores desktop application aiming to determine whether it functions properly and to answer the next important questions:[4]

- ❖ Can users operate with this?
- ❖ Does this function work as intended?

A. Check for broken links (Broken link refers to a hyperlink which does not work).

B.

Warning messages: User input should get verified at system level according to business rules and error/warning messages should be flash to user for incorrect inputs.

Resolution change effect on the application: Ensure that application's functionality and design is compatible with the different resolutions

C.

Print: Following points must be verified- Test the print functionality of the application when no printer connected - application should behave correctly if printer is not available. - Test the print functionality of the application when printer is connected- Ensure that application queues prints in printer if papers are not available in printer.- To ensure that lengthy description of an event is not truncating on print layout in selected event printing.

D.

Change: Ensure the successful launch of application after theme change.

E. Installation Testing (Upgrade/Downgrade): Verify application is included in Programs and Features list after installation. Also Verify application is removed from Programs and Features list after un installation. Keep in mind that older version of application should not be install on latest version.

F.

Testing with multi user accounts: Open Control Panel, User Accounts, and add 2 user accounts (Standard and admin) to the system. With the application running, press Start, the Switch User to the user account just created. Verify application launches and runs correctly on the newly created user account. Switch back and forth between user accounts and use the application in both. Watch for any performance decreases and check functionality. Sleep: While the application is running, put the system to sleep (S3). Wake the system up after two minutes.

a) Verify the application is still running.

b) Verify there is no distortion or error.

G. Cache: - Delete the application's cache, launch the application and verify that application should work properly.- Delete the application's cache while application is running and verify that application should work properly.

H. A stand alone application is used by single user so, during functional testing, if the application responded fine than it should work all the times in future.

Level 3 - Compatibility Testing:
a. Test on different Operating systems: Some functionality in your web application may not be compatible with all operating systems. All new technologies used in web development like graphics designs, interface calls like different API's may not be available in all Operating Systems.

Test your web application on different operating systems like Windows (XP, Vista, Win7 etc), Unix, MAC, Linux, Solaris with different OS flavors.

Level 4 - Performance Testing:
Performance testing of Desktop Applications is a type of software testing executed in order to measure the time required to perform the specific actions, along with determining the intensiveness of such actions, to identify the possible number of users who can work simultaneously on a desktop app and to explore the performance under the high, limited and stress workload.

a. Long period of continuous use: Is site able to run for long period, without downtime.
b. Memory: Note down the average memory usage in Comments column.
c. Generate "Power Efficiency Diagnostics Report" by running the command powercfg /energy.

Level 5: Usability Testing*:*

Usability testing for Desktop applications is a black-box technique which evaluates how easily the user can learn to operate a desktop app or level of its usability.

As this type of testing must be done by human-computer interaction, our testers set the next quantitative goals:

- Efficiency: Testers aim to detect the time and steps which are required for user to complete the basic tasks of a desktop app.
- Accuracy: Detecting the number of mistakes which were made by people when using your desktop app and whether they were critical.
- Recall: Testers determine how much users remember after using a desktop application.
- Emotional response: Detecting the feelings left after using a desktop app to know whether people will recommend it to their friends.

After doing vigorous testing we come to know that, Performance testing is the most important testing in case of Standalone/ Desktop Application.

IV. USEFUL FINDINGS ON DESKTOP APPLICATION PERFORMANCE:

- Many instances of badly designed SQL were subsequently optimized
- Several statements taking minutes were improved to sub-second
- Several incorrect views were identified
- Some table indexes that were not set up were also identified and corrected
- Too much system memory consumed by the desktop application
- Program crashes often occur when repeated use of specific features within the application causes counters or internal array bounds to be exceeded.
- Reduced performance due to excessive late binding and inefficient object creation and destruction
- Memory leak identified, when the application was opened and left for a longer period of time (few hours)..

V. CONCLUSION:

This paper provides detailed testing approach, strategies on standalone/Desktop application testing by summarizing the related concepts, issues, characteristics for the standalone/Desktop application testing and also the checklist for the same. Obviously these checklists and parameters have its own advantages and disadvantages. And the selection criteria are totally based on your requirements of standalone/Desktop application testing. Moreover Desktop application testing opportunities, related research work is also reviewed.

REFERENCES:

[1] http://www.todaysoftmag.com/article/745/an-overview-of-performance-testing-on-desktop-solutions

[2]https://msdn.microsoft.com/en-us/library/windows/desktop/aa378892(v=vs.85).aspx

[3]http://stackoverflow.com/questions/7845451/technical-difference-between-a-stand-alone-app-vs-web-app

[4]https://testfort.com/functional-testing-desktop