www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 10 October, 2014 Page No.8859-8864

Taxonomy of Testing Techniques

Gautam Chaturvedi[1], Shubha Chaturvedi[2], Kiran Bhagnani[3]

[1] H.O.D, Department of Computer Science,
Sophia Girls College, Ajmer, Rajasthan, India
chaturvedigautam@yahoo.com

[2] Department of Computer Science,
Sophia Girls College, Ajmer, Rajasthan, India
shubha.sophia28@gmail.com

[3] Department of Computer Science,
Sophia Girls College, Ajmer, Rajasthan, India
kiranbhagnani2609@yahoo.com

Abstract: ***"As fire is the test of gold, being patient is the test of men, so is the testing is the test of assessing the quality of the product!!!" With the development of Fourth generation languages (4GL), the proportion of time devoted to testing has been increased and is truly effective means. Testing technique research leads to the destination of practical testing methods and tools. Progress toward this destination requires fundamental research, and the creation, refinement, extension, and popularization of better methods. The methodology for the paper is case study of our college (Sophia Girls' College) Office Automation Software. Our study focuses on the state of the art in testing techniques, as well as the latest techniques which representing the future direction of this area.***

Keywords: GUI, ST, SDLC, SMS, life cycle

1. Introduction

Software Testing (ST) is a process of executing a program or application with the intent of finding software bugs. ST is to ensure that software should always be defect free and easily maintained. It's simply an investigation conducted to provide stakeholders with information about the quality of the product or service under test. [1].It's a very broad area, which involves many other technical and non-technical areas, such as specification, design and implementation, maintenance, process and management issues in software engineering .[2]It is been said that Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process. [3]
Before stepping into any detail study of techniques, let us have a brief look at some important topics that are relative to our research.

2. Different Goals of Software Testing

Software testing is the mechanism of knowing that what's the expected result and what the actual result software project or product has given. The ultimate goal of software testing is to help designers, developers, and managers construct systems with high quality.[3] Thus research and development on testing aim at efficiently performing effective testing – to find more errors in requirement, design and implementation, and to increase confidence that the software has various qualities which also gives develpoer and user a feeling of satisfaction.

Few points to remember-

1. We need to first identify the bugs as early as possible.

2. Than needs to prevent the bugs in a project and product also.
3. Next is to check whether the customer requirements criteria is met or not.
4. And finally main goal of testing is to measure the overall quality of the product and project.

3. Fundamentals of Software Testing

Testing objectives include the following :
1. It is a process of executing a program with the intent of finding an error.
2. A good test case is one that has a high probability of finding an as yet undiscovered error
3. A successful test is one that uncovers an as yet undiscovered error.
In a minimum amount of time and with a minimum amount of effort testing should systematically uncover different classes of errors. A secondary benefit of testing is that it demonstrates that the

software appears to be working as stated in the specifications.

4. Basic Testing life cycle

Although variations exist between organizations, there is a typical cycle for testing.[3,7] The sample below is common among organizations employing the Waterfall development model.

1. Requirements analysis: Testing should begin in the requirements phase of the software development life cycle. During the design phase, testers work with developers in determining what aspects of a design are testable and with what parameters those tests work.
2. Test planning: Test strategy, test plan, tested creation are the many activities which will be carried out during testing for which, a plan is needed.
3. Test development: It includes test procedures, test datasets, test scenario test cases, and test scripts to use in testing software
4. Test execution: Testers execute the software based on the plans and test documents then report any errors found to the development team which can be further be solved.
5. Test reporting: Once testing is completed, testers generate metrics and make final reports on their test effort and are ready for release.
6. Test result analysis: The development team usually along with the client does Defect Analysis, in order to decide what defects should be treated, fixed, rejected (i.e. found software working properly) or should be deferred to be dealt with later.
7. Defect Retesting: Once the development team has dealt with a defect, it is retested by the testing team.
8. Regression testing: It is common to have a small test program built of a subset of tests,for each integration of new, modified, or fixed software, in order to ensure that the latest delivery has not ruined anything, and that the software product as a whole is still working correctly.
9. Test Closure: Once the test meets the exit criteria, the activities such as capturing the key outputs, lessons learned, results, logs, documents related to the project are archived and used as a reference for future projects.

5. Testing Methods [4]

It includes:

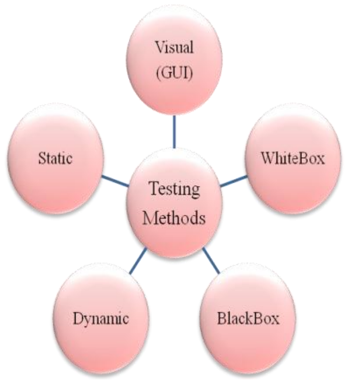


Figure 1: Testing Methods

A. STATIC:

Static testing starts early in the life cycle. It is the testing of the software whose work products manually, or within a set of tools, but they are not executed e so it do not need computer, as testing is done without program executing.. Few examples are walkthrough, inspections and reviewing.

B. DYNAMIC:

It is a testing the dynamic behavior of code. The software is executed by using the computers. It involve working with the software ,giving input values and checking the output is as expected by executing the specific test cases either manually or by automated process. Example: unit testing, integration testing and system testing.

C. BLACK BOX TESTING:

It is also known as Specification based testing technique or Input/output Driven Testing technique.
This testing treats the software as a 'Black box' examining functionality without giving any information of internal implementation. The tester is only aware 'of what' the software is supposed to do and not, how does it do.

D. WHITE BOX TESTING:

It is also known as Structured based technique or 'glass box' testing technique. This testing tests internal structures or workings of a program. Here the testers require knowledge off how the software is implemented, how it works.

E. VISUAL (GUI) TESTING:

GUI testing is the process of testing a product's graphical user interface to ensure its meets its written specifications like testing images and buttons, icons alignments on any webpage.

6. Testing levels:

Every System Development Life Cycle (SDLC) phase goes through the testing. Hence there are various levels of testing.

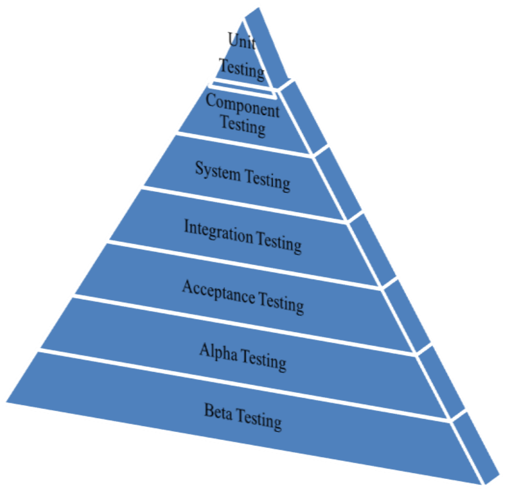


Figure 2: Testing levels

A.**UNIT TESTING:**

Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or derived/child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module. Unit Testing is performed by using the White Box Testing method. Unit Testing is the first level of testing and is performed prior to Integration Testing.

B.COMPONENT TESTING:

It's also known as modules and program testing. Component testing is a method where testing of each component in an application is done separately. It is the phase in which individual software modules are first tested to find defects and later been verified for the functioning of software modules. This testing is done in isolation from rest of the system depending on the development life cycle model chosen from that particular application.

C. INTEGRATION TESTING;

It is a level in which individual software modules are combined and tested as a group .It occurs after Unit testing and before validation testing. It is done by specific integration testers or test team. The purpose of this level of testing is to expose faults in the interaction between integrated units. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

D. SYSTEM TESTING:

System Testing is a level of the software testing process where a complete, integrated system software is tested. System testing is usually required before and after a system is put in place. A series of systematic procedures are referred to while testing is being performed. These procedures tell the tester how the system should perform and where common mistakes may be found. Testers usually try to "break the system" by entering data that may cause the system to malfunction or return incorrect information. It is most often the final test to verify that system to be delivered meets the specifications and its purpose. System testing must investigates both functional and non functional requirement of the testing.

E. ACCEPTANCE TESTING:

Once the system test has corrected all or most of the defects, the system will be delivered to the user or customer foe acceptance. Acceptance testing is conducted to determine if the requirement of the specifications is met prior to its delivery or not. User or customer side usually does this testing.

F. ALPHA TESTING:

Alpha testing is testing of an application when its development is just about to be finished. This test takes place at the developer's site who observe the users and note problems. Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site.Minor design changes can still be made as a result of alpha testing. It is a final testing before the software is released to the general public.

G. BETA TESTING:

It is also known as 'Field' testing. Beta testing comes after alpha testing and also takes place at consumer site .It is released to selected customers for testing under normal, everyday conditions of use to spot the remaining flaws.It sent the system to user who install it and use it under real world working conditions.

7. TESTING TYPES

A test type is focused on particular test objectives:

A.Functional Testing

Functional Testing is a type of software testing whereby the system is tested against the functional requirements/specifications by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is not concerned with how processing occurs, but rather, with the results of processing. Functional testing is normally performed during the levels of System Testing and Acceptance Testing.

It includes the following:

- Installation
- Development
- Usability
- Smoke
- Sanity
- Regression
- User acceptance
- Automated
- Destructive

- Recovery

B.NON FUNCTIONAL TESTING:

Non-functional testing is the testing of a software application for its non-functional requirements. It is also the testing the application against client's and performance requirement.

It includes the following:

- Compatibility
- Performance
- Security
- Accessibility
- Internationalization/Localization

8. FUTURE OF SOFTWARE TESTING [5]

While technology continues to evolve at a rapid pace, software testing is evolving right along with it.After evaluating for more than 3 the testing area of the development life cycle is starting to be appreciated on a much wider scale for its value and is being seen as a professional career in its own rightway . There are many Masters Courses in software testing and universities are starting to use industry developed text books as required reading in their computer science (or equivalent) courses. We see many certification courses and development programs established in many countries and see skills such as time management, team development, line management and communication all being part of the effective testers repertoire. We still see many familiar problems repeatedly causing issues at organizations. In a great many environments there is:

•Late lifecycle testing – finding the majority of bugs late in the day

•80% of all testing still being manually performed Poor quality requirements being utilized to drive a project, and thus the testing.

From academic studies and reports at conferences such as StarEAST we also know that:

Its users utilize

•A recent study has highlighted that only 40% of a typical software product

•Multiple studies have shown that around 65% of defects in software project can be directly linked back to discrepancies, ambiguities and errors in the requirements.

9. RESULT ANALYSIS

Through this paper we would like to elaborate a case study of Office Automation System done in Sophia Girls College, Aimer. We have a concept of sharing information of students to their parents by sending SMS to their mobile through this Office Automation. Here we would like to draw your attention on how the SMS are first designed, coded, tested, verified and then further been implemented after administrator‘s approval. The software consists of Visual Studio 2008 as a Front-end and SQL Server2005 as a back-end software. Below the case study is been shown in the similar modal as SDLC. We have also discussed what are the difficulties faced during the whole process.

Note: The form for all the steps are already been designed we are just filling and approving it.

A. Requirement Gathering, Feasible Study and Analysis Phase

As we know that it includes gathering, analyzing, validating, and specifying requirements. At the end of this phase, the Software Requirement Specification (SRS) document is prepared. SRS is a formal document that acts as a written agreement between the development team and the customer.

Similarly why and what type of SMS is to be created is been discussed and further get analyzed and than a Login form is been filled which have a USER ID and PASSWORD.

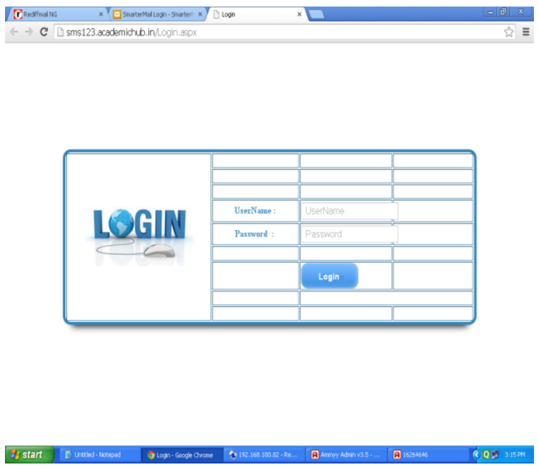


Figure 2: Form for Login

Limitation:

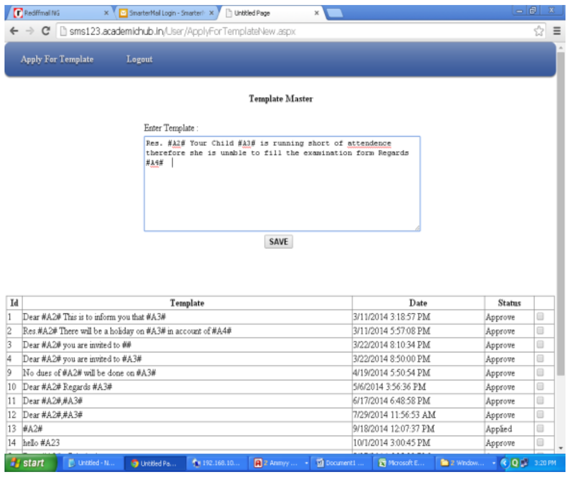
Only predefined password an only be used.

B. Designing Phase and Coding Phase

In this phase we designed and coded a module in which **Template master window** gets open. Here we specify two things:

Firstly, we have to choose a format for our template specified from the below list and which so ever template is selected **Applied** will be written on the status field.

Secondly whatever message has to be forwarded les it has to be written in the **Enter Template window**. Finally we click on save button.


Rediffmail NG
SmarterMail Login - Smarter
Untitled Page
sms123.academichub.in/User/ApplyForTemplateNew.aspx
Apply For Template
Logout
SAVE
start

Template Master

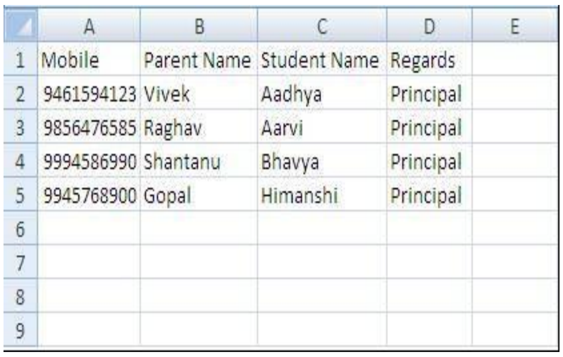
Enter Template :

```
Res. #A2# Your Child #A3# is running short of attendence
therefore she is unable to fill the examination form Regards
#A4#
```

Id	Template	Date	Status	
1	Dear #A2# This is to inform you that #A3#	3/11/2014 3:18:57 PM	Approve	☐
2	Res.#A2# There will be a holiday on #A3# in account of #A4#	3/11/2014 5:57:08 PM	Approve	☐
3	Dear #A2# you are invited to ##	3/22/2014 8:10:34 PM	Approve	☐
4	Dear #A2# you are invited to #A3#	3/22/2014 8:50:00 PM	Approve	☐
9	No dues of #A2# will be done on #A3#	4/19/2014 5:50:54 PM	Approve	☐
10	Dear #A2# Regards #A3#	5/6/2014 3:56:36 PM	Approve	☐
11	Dear #A2#,#A3#	6/17/2014 6:48:58 PM	Approve	☐
12	Dear #A2#,#A3#	7/29/2014 11:56:53 AM	Approve	☐
13	#A2#	9/18/2014 12:07:37 PM	Applied	☐
14	hello #A23	10/1/2014 3:00:45 PM	Approve	☐

Figure 2: Form for Template Master

Below is also the Excel list has been shown which includes mobile number of parents, father and student name faculty wise. This file will be further used in Implementation Phase.



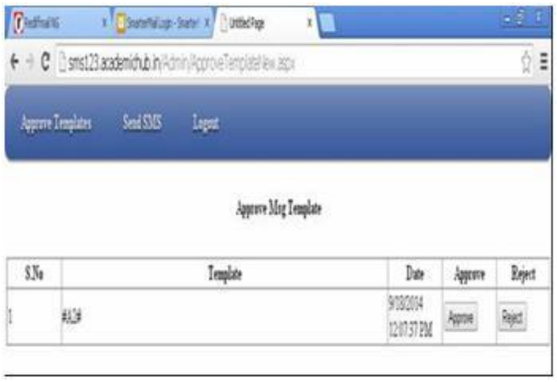
	A	B	C	D	E
1	Mobile	Parent Name	Student Name	Regards	
2	9461594123	Vivek	Aadhya	Principal	
3	9856476585	Raghav	Aarvi	Principal	
4	9994586990	Shantanu	Bhavya	Principal	
5	9945768900	Gopal	Himanshi	Principal	
6					
7					
8					
9					

Table 1: Information of Students.

Limitation:
Here the only limitation we faced is that we have to write #A2# , #A3#,#A4#... which are name of the columns present in our Excel sheet.
For example in the above Figure #A2# means name of the Student Parents and #A3# means

C. Verifying and Testing Phase

Next step is for the **Verificatio**n and **Approva**l of the template by the administrator. She can approved or reject the SMS. Once the SMS is selected we are ready to send and if not changes are again made and again sent for approval..


Rediffmail NG
SmarterMail Login - Smarter
Untitled Page
sms123.academichub.in/Admin/ApproveTemplateNew.aspx
Approve Templates
Send SMS
Logout
Approve
Reject

Approve Msg Template

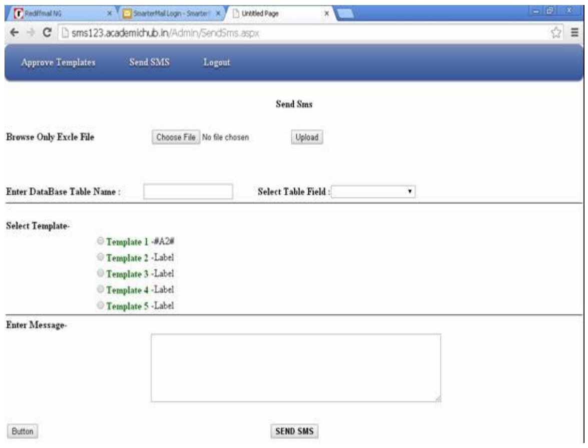
S.No	Template	Date	Approve	Reject
1	#A3#	9/18/2014 12:07:37 PM	Approve	Reject

Figure 2: Form for Approval Message Template

D. Implementation Phase

For sending the SMS we have to first choose the **Excel file** having the list of names of the students example Sheet1 and clicking on Upload button. We can do this by directly typing the Database Table Na

Now we are done with Designing, testing, verifying of the SMS the only step is left with implementation of the SMS by selecting on **Send SMS button.**



Send Sms

Browse Only Excle File: No file chosen

Enter DataBase Table Name :

Select Table Field :

Select Template-
- Template 1 -#A2#
- Template 2 -Label
- Template 3 -Label
- Template 4 -Label
- Template 5 -Label

Enter Message-

Figure 2: Form for Sending SMS

Limitation: The limitation lies in this form is that we can only write Sheet 1, 2,3 name oly for our sheet. Changing our name for example B.C.A-I will not allow to connect our list for sending the database.

Below is give a **Coding for a module** who is use to approved the SMS

```
 using System;
using System.Collections;
using System.Configuration
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;
public partial class ApproveTemplate :
System.Web.UI.Page
{
    cls_Common_Class comm = new
cls_Common_Class();
    protected void Page_Load(object sender,
EventArgs e)
    {
        if (!IsPostBack)
        {
            FillGrd();
        }
    }
protected void FillGrd()
    {
        SqlParameter[] s = new SqlParameter[2];
        s[0] = new SqlParameter("@Action",
"GetData");
        s[1] = new SqlParameter("@Status", "Applied");
        DataTable
dt=comm.getDataTableSpParams("spSmsTemplateM
aster",s);
        if (dt.Rows.Count > 0)
        {
            GridView1.DataSource = dt;
            GridView1.DataBind();
        }
        else
        {
            GridView1.DataSource = null;
            GridView1.DataBind();
        }
    }
```

10. CONCLUSION

At end we conclude that main goal of software testing is to show that application is working as per as the requirements defined by client. Software testing is a vital element in the SDLC and can furnish excellent results if done properly and effectively. Unfortunately, Software testing is often less formal and rigorous than it should, and a main reason for that is because we have struggled to define best practices, methodologies, principles, standards for optimal software testing. To perform testing effectively and efficiently, everyone involved with testing should be familiar with basic software testing goals, principles, limitations and concepts. Already lot of work has been done in this field, and even continues today. Implementing testing principles in real world software development, to accomplish testing goals to maximum extent keeping in consideration the testing limitations will validate the research and also will pave a way for future research. [6]

11. REFERENCES

[1] Cem Kaner,, “Exploratory Testing “ Florida Institute of Technology, Quality Assurance Institute Worldwide Annual Software Testing Conference, Orlando, FL, November 2006.

[2] Lu Luo,” Software Testing Techniques, Technology Maturation and Research Strategy”,Class Report for 17-939A. Institute for Software Research International, Carnegie Mellon University Pittsburgh, A15232,USA.

[3] (http://www.codeproject.com/Tips/351122/What-is-software-testing-What-are-the-different-l.

[4] http://www.slideshare.net/confiz/software-testing-methods-levels-and-types.

[5] http://www.automatedtestinginstitute.com/home/index.php?option=com_content&id=254:white-paper-the-future-of-software-testing

[6] S.M.K Quadri, Sheikh Umar Farooq, “Software Testing –Goals,Principles, and Limitations, International Journal of Computer Applications (0975 – 8887) Volume 6– No.9, September 2010.

[7] http//.en.wikipedia.org/wiki /Software_testing.