ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

A Survey on Test Automation

Samiksha R. Rahate, Uday Bhave

M.E Student, Department of Computer Engineering, Shah & Anchor Kutchhi Engineering College, Mumbai, India

HOD, Department of Computer Engineering, Shah & Anchor Kutchhi Engineering College, Mumbai, India

ABSTRACT*:* Test automation is a field of research around for a very long time. Still, testing and related concepts are considered to be one of the most time-consuming and expensive parts of the software life cycle. Although it is a field with a relatively long research background, many existing test automation systems are still relatively simple and not very different from the early days. They still focus on executing an existing, usually manually crafted, set of tests over and over again. In this Survey, we are exploring about Automation testing. Automation testing is useful to achieve and maintain quality of software's. And for this purpose detailed planning is required. There are different approaches to implement automation testing. There are no of Automation tools that we can use as per our requirements. Here some tools and their use is mentioned. And working of the tool "**Bugzilla**" is explained in detail
.

KEYWORDS: Automated Testing, Bugzilla

I. INTRODUCTION

Software testing is most important phase in software development life cycle in order to ensure that our product is up to the requirement. If testing is not done properly them all other phases right from requirement review to implementation will be useless. Software testing mainly has two types: **manual** testing and **automation** testing. Testing software manually means without using any script or automated tools. Here, software tester plays role of user and looks software from user's perspective to find the defects in system. In, automation testing is process in which testing activities are done through some script. Here, tester uses another software test product.

Today most of the companies are implementing automation testing for ensuring quality of their product from this only we can understand the importance of automation tools in software engineering. Creating automated test cases will cause initial development overhead but compared to manual execution, overall execution time will be decreased for each test case regression. [3] Test automation is not the same as testing, and the skills needed to be a good automation tester, are not the same skills needed to be a good tester. [11]

For each and every test activities we are having different tools still we cannot make testing fully automatic we still need some manual testing. Automation testing has different approaches and methodologies of implementation. The testing methodologies can be used together or individually. We have to select according to our requirements of testing because this methodologies provides different resources for different tasks. In this report, approaches and methodologies are explained further. Main objective of this report is to study automation testing in detail

II. AUTOMATION TESTING

Today's world is full of competition and speed. As technology is becoming a strong weapon to acquire market place for product. Along with this quality and cost of product are becoming most important factor to stay in market. This is the reason why companies' selects automation testing for their product. Automation process has many advantages like efficiency, cost effectiveness and coverage. But automation testing cannot eliminate all the bugs from system so in some extent manual testing is necessary. According to "Douglas Hoffman**"** test case is complete when all of the following elements are present [12]:

- Ability to run two or more specified test cases
- Ability to run a subset of all the automated test cases
- No intervention is needed after launching the tests
- Automatically sets-up and/or records the relevant test environment parameters

DOI: 10.15680/IJIRCCE.2016. 0406298

- Runs the test cases
- Captures the relevant results
- Compares actual with expected results and flags differences
- Analyses and reports pass/fail for each test case and for the test run

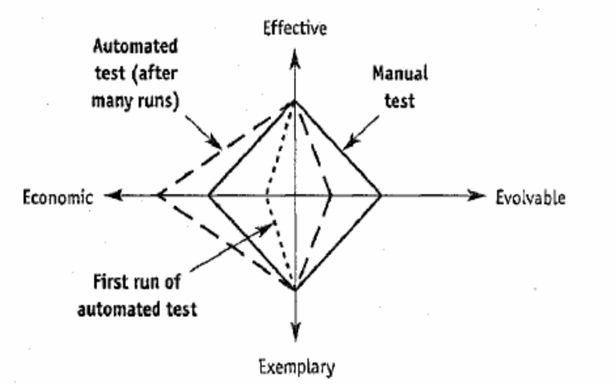


Figure 1. The goodness of a test can be illustrated by considering the four attributes in this Keviat diagram. The greater the measure of each attribute the greater the area enclosed by the joining lines and better the test case.

A. WHEN TO AUTOMATE?

Now, question is when we should start test automation? In SDLC testing phase is shown after the implementation phase but in actual testing should takes place at each phase and Automation can be start at implementation phase. Below are some points at which we can have test automation:

1) **Large and critical projects**
2) **Regression testing**
3) **Requirements changing frequently**
4) **Accessing application for load and performance with many users**
5) **Stable software w.r.t. manual testing.**

Availability of time is less.

B. *HOW TO AUTOMATE?*

Automation process works in following steps:

Identify which area of your application needs automation testing. Each type of testing has different tools so from range of automation tools choose appropriate tools by comparing your requirements and tools configurations so that you will get more appropriate tools. With the help of tools create test scripts, these test scripts means a set of instructions. Now create a test suit from this test scripts, test suit is collection of test cases. Then execute these test cases to get some result report. By analyzing this reports identify is there any potential bugs in the system or is there any performance issue in the system.

DOI: 10.15680/IJIRCCE.2016. 0406298

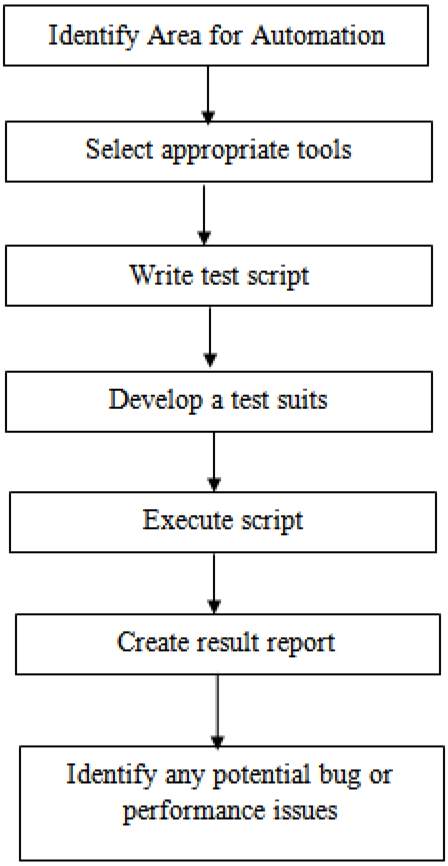


Figure 2 Steps for automation testing

III. FRAMEWORKS AND METHEDOLOGIES

A. FRAMEWORKS AND AUTOMATION

A test automation framework is a set of assumptions, concepts and tools that provide support for automated software testing. The main advantage of such a framework is the low cost for maintenance. [7] Automation testing has 4 frameworks as follow: [17]

- Module Based
- Data Driven
- Keyword Based
- Hybrid Based

1) Module based

The Modularity testing framework is built on the concept of abstraction. This involves the creation of independent scripts that represent the modules of the application under test. These modules in turn are used in a hierarchical fashion to build large test cases. Thus it builds an abstraction layer for a component to hide that component from the rest of the application. Thus the changes made to the other part of the application do not effect that component

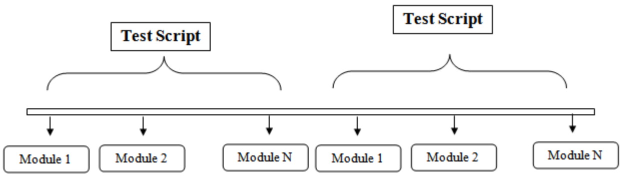


Figure 3 Module based Framework

- **Advantages of Module based Framework**

  - Modular division of scripts leads to easier maintenance and also the scalability of the automated test suites.
  - The functionality is available in easy to use test libraries so creating new driver scripts for different tests is easy and fast.

- **Disadvantages of Module based Framework**
  - The main problem with modular frameworks is that the test script have test data embedded in them. So when the test data needs to be updated we need to change the code of the script. This becomes a big problem when the test script is large.

    For this purpose, data- driven testing frameworks have been introduced

2) Data driven

Data driven framework helps user segregate the test script logic and test data from each other. It lets the user to store test data in an external database. This data is conventionally stored in “key-value” pairs this keys can be used to access data within the script

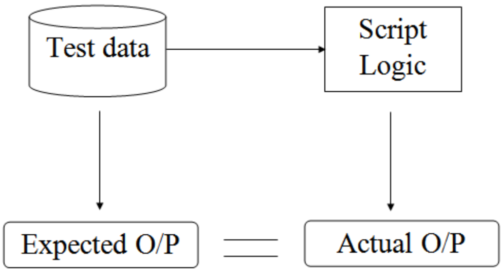


Figure 4 Data driven Framework

- **Advantages of Data driven framework**:

- This framework reduces the number of overall test scripts needed to implement all the test cases.
- Less amount of code is required to generate all the test cases.
- Offers greater flexibility when it comes to maintenance and fixing of bugs.
- The test data can be created before test implementation is ready or even before the system to be tested is ready

- **Disadvantages of Data driven framework**:
- The test cases created are similar and creating new kind of tests requires creating new driver scripts that understand different data. Thus the test data and driver scripts are strongly related that changing either requires changing the other.

For this purpose keyword driven testing frameworks have been introduced

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

3) Keyword Based

It is an extension to data driven framework along with test data it also keeps certain set of code belonging to test script into an external file. These set of codes are known as keywords. These keywords are self-guiding as what action need to be performed on the application

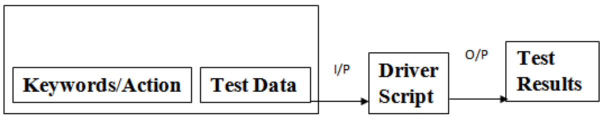


Figure 5 Keyword driven Framework

- **Advantages of Keyword driven framework**:
  It has all the advantages that data driven testing has
  - Automation expertise is not required to maintain or create a new set of test cases
  - Keywords are reused across multiple test cases.

- **Disadvantages of Keyword driven framework**
  - The main problem is that this requires a more complicated framework than the data driven framework.
  - With the keyword driven approach the test cases get longer and complex and this is due to the greater flexibility that this approach offers.

So, in order to combine the strengths of all the frameworks and mitigate their weaknesses we use the hybrid testing framework

4) Hybrid Based

Hybrid testing framework is the combination of modular, data-driven and keyword driven testing frameworks. This combination of frameworks helps the data driven scripts take advantage of the libraries which usually accompany the keyword driven testing.

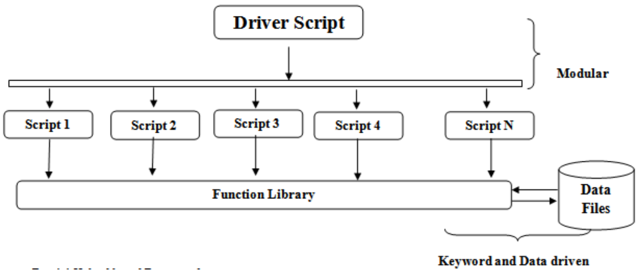


Figure 6 Hybrid based Framework

B. Methodologies used for automation tools

There are different methodologies that are can be used when implementing testing tools. The testing methodologies can be used together or individually. It all depends on what you are testing as they provide better resources for different tasks. Below are a few of the main different types that are used but there are other types of testing that can be used:

1) Unit testing

Unit Testing is first phase of testing after implementation. This is generally performed by developers. Unit testing tool is the software used for verification and validation. It’s a method in which the programmer can test all the separate code and see if it is viable to use. This type of testing is based on a small scale and uses small units of the program. When looking at procedural programming the unit can be any individual function or a procedure which is written in the same language as the production code.

2) *Integration testing*

This is the next phase in V- model where the separate software modules are put together and are then tested as a whole. This normally takes place after unit testing has been done. The point of doing integration testing is because it verifies the functional and performance requirements of the design items. It is component of Extreme Programming (XP). Two different ways of doing integration testing is bottom-up and top-down.

3) *Regression Testing*

This is a type of software testing which basically looks for software regressions. The regressions can happen whenever the software stops working as it is meant to. The testing makes sure that any bugs or changes have been fixed which may have caused previous working functions to have failed as a result of the newly added features. It is better to do this type of testing under the automated tool environment. The regression testing may take place once the programmer has tried to fix a problem or has purposely added in code to give out errors

4) *White box testing*

White Box testing is also known as clear box testing, glass box testing, translucent box testing or structural testing. It uses the internal perspective of the system and then designs test cases based on this internal structure. Basically the code itself and all the conditions, statements and paths along with it are tested. Programming skills are required for noticing all the paths through the software. White box testing does many things such as analyzing the data flow, control flow, information flow and coding practices.

IV. TESTING TOOLS

A. *Types of testing tools*

There are many different types of testing tools. Each different testing tool is capable of doing different things as each one has different abilities to do different things. These are considered to be in the testing tools environment. Over the years there have been much advancement on the different types of testing tools and it doesn't stop many more coming in the future in which they all have their own capabilities of being used for testing. Below is a list of each one available and an explanation of what they do. [5]

1) *Unit Testing Tools*

Using Unit testing tools it is for programmer to easily test each module separately. JUnit is a very good testing tool for using with unit testing. It is a java-based tool, which is capable of going through code especially with extreme programming as it takes up less time to do.

2) *Regression testing tools*

Regression testing tools are used to save time and resources. Many different tools can be used for regression. It helps to automate the testing as the program can be reused. Example of this type of testing tools is Rational's Team Test.

3) *Web testing tools*

This is used over web based applications. It tests for bugs and problems that appear within the application. An example of this may be that once a problem is detected such as having dead links, line checking or html validation then it provides help in repairing this problem.

For example: Selenium, WATIR

4) Security testing tools

These can help aid with the security of the system. It can help against unauthorized access, hacking and any coding damage which deals with the code of application. It uses very sophisticated testing techniques. QA inspect is a very good program to use for this type of testing.

5) Functional testing tools

Functional testing tools helps tester to check functionalities of applications. There are so many functional testing tools like Mercury Interactive Company WinRunner, QuickTest ProJBM Rational Company's Rtional Robot, Rational Xde Tester , Compuware Company's QARun, RadView Company's WebFT, Empirix Company's eTester, etc

6) Performance Testing Tools

This tool is used to check performance factors like load, stress. There so many tools available like Mercury Interactive Company's Loadrunner, Compuware Company's QALoad, Microsoft Company's web Application Stress Tool, etc.

7) Database Testing Tools

Database testing tools are very good to use for checking and testing databases. It helps with creating tables and data to test the database. An example program is SQL DB Validator. It performs database and data cube verification and validation.

8) Communication Testing Tools

Communications testing tools is used for communication purposes. It can work wirelessly, through sockets and SOAP, GPRS and other network communication methods. A program that can be used for this type of testing is Cheetah. It allows you to proactively test and monitor your VoIP and VoD performance whilst it maintains the integrity of other critical applications.

9) Requirement Management Tools

This is used to analyze the requirements for testing, maintaining and logical inconsistencies. A program used for this is called Requisite Pro which is a management tool to help improve communication goals and enhance collaborative development and increase the quality of applications before deploying. This helps the tester to validate the program correctly.

B. Some Recent testing tools

Below are some recent tools that are used by most of the companies using now days.

1) QTP/UFT (Quick test professional/Unified functional testing tool)

It is product of HP. Using QTP we can have regression and functional testing. Both technical and non-technical user can easily handle QTP. It has record and playback feature. We can test both web based as well as desktop based application. It supports module based, data driven, keyword based frameworks. It comes with inbuilt IDE. It can be integrated with tools like quality centre, Test director and WinRunner.

2) Selenium

Selenium is a free automated testing suit for web applications across different browsers and platforms. It is quite similar to QTP. It is not just single tool it has four components.

1) Selenium IDE
2) Selenium Remote control (RC)
3) Web Driver
4) Selenium grid

Selenium 1 refers only selenium RC but Now RC and webdriver are merged into a single framework to form selenium 2.Selenium IDE is simplest framework. It is Firefox plugin that you can install easily because of its simplicity it can be used only as prototyping tool for more advance test cases we need RC or web driver

Using Selenium RC you are free to use various programming languages to write test script like Java, C#, Perl, Python, Ruby. Web driver proves itself to be better than both IDE and RC. As it controls browser by directly communicating to it. Selenium Grid is tool used together with Selenium RC to run parallel tests across different browsers.

3) WATIR

This is pronounced as “water”. It is family of ruby libraries. It supports all web application no matter which language it is developed in. It supports multiple browsers in different platforms. Some famous companies that uses WATIR are Yahoo, SAP, Honeywell, Oracle, Facebook etc.

4) Bugzilla

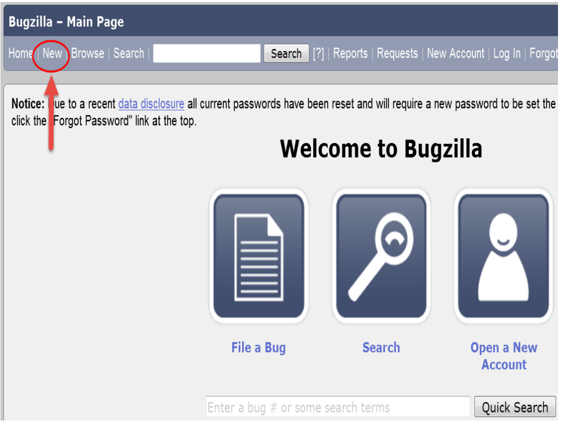
Bugzilla is a "**Defect Tracking System**" or "**Bug-Tracking System**". Defect Tracking Systems allow individual or groups of developers to keep track of outstanding bugs in their product effectively. It

- Track bugs and code changes
- Communicate with teammates
- Submit and review patches
- Manage quality assurance (QA)

C. Working with Bugzill

Creating a Bug-report in Bugzilla

Step 1: To create a new bug in Bugzilla, visit the home-page of Bugzilla and click on NEW tab from the main menu


Bugzilla – Main Page

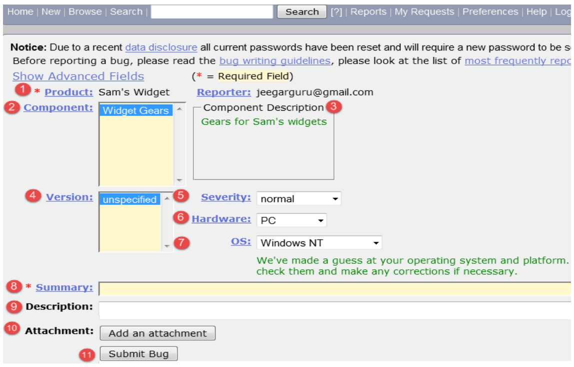
Notice: Due to a recent data disclosure all current passwords have been reset and will require a new password to be set the click the "Forgot Password" link at the top.

Welcome to Bugzilla

Home | New | Browse | Search | Search | [?] | Reports | Requests | New Account | Log In | Forgot
File a Bug
Search
Open a New Account
Enter a bug # or some search terms
Quick Search

Figure 7 Main page of Bugzilla

Step 2: Fill all the details about your product. If you do not fill mandatory fields you will get error


Notice: Due to a recent data disclosure all current passwords have been reset and will require a new password to be s
Before reporting a bug, please read the bug writing guidelines, please look at the list of most frequently repo

Show Advanced Fields (* = Required Field)

1 * Product: Sam's Widget Reporter: jeegarguru@gmail.com

2 Component: Widget Gears

Component Description 3
Gears for Sam's widgets

4 Version: unspecified

5 Severity: normal

6 Hardware: PC

7 OS: Windows NT

We've made a guess at your operating system and platform. check them and make any corrections if necessary.

8 * Summary:

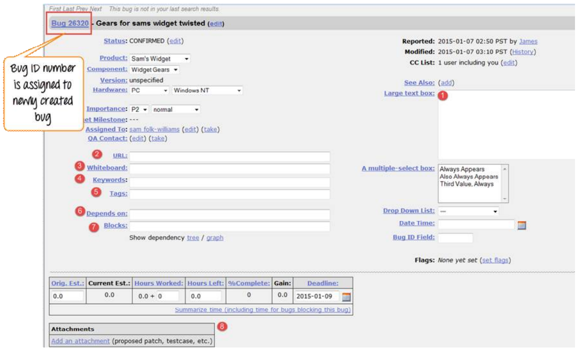
9 Description:

10 Attachment:

Home | New | Browse | Search | Search | [?] | Reports | My Requests | Preferences | Help | Log
Add an attachment
11
Submit Bug

Figure 8 Fill details about product

Step 3: Now your new bug is created here you can also add additional information to the assigned bug like URL, keywords, whiteboard, tags, etc. This extra information is helpful to give more detail about the Bug you have created.



First Last Prev Next This bug is not in your last search results.

Bug 26320 - Gears for sams widget twisted (edit)

Status: CONFIRMED (edit)

Product: Sam's Widget
Component: Widget Gears
Version: unspecified
Hardware: PC Windows NT

Importance: P2 normal
et Milestone: ---
Assigned To: sam folk-williams (edit) (take)
QA Contact: (edit) (take)

2 URL:
3 Whiteboard:
4 Keywords:
5 Tags:

6 Depends on:
7 Blocks:
Show dependency tree / graph

Reported: 2015-01-07 02:50 PST by James
Modified: 2015-01-07 03:10 PST (History)
CC List: 1 user including you (edit)

See Also: (add)
Large text box: 1

A multiple-select box: Always Appears
Also Always Appears
Third Value, Always

Drop Down List: ---
Date Time:
Bug ID Field:

Flags: *None yet set* (set flags)

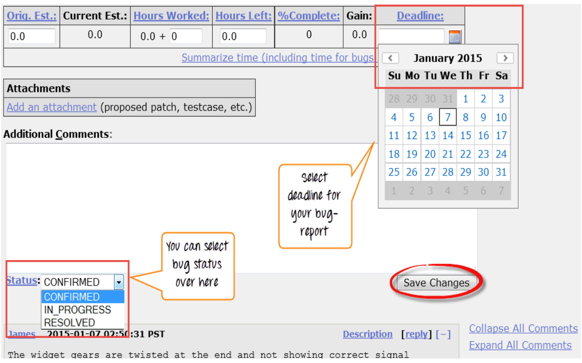
Orig. Est.:	Current Est.:	Hours Worked:	Hours Left:	%Complete:	Gain:	Deadline:
0.0	0.0	0.0 + 0	0.0	0	0.0	2015-01-09

Summarize time (including time for bugs blocking this bug)

Attachments 8
Add an attachment (proposed patch, testcase, etc.)

Figure 9 Bug is created

Step 5: In the same window if you scroll down further. You can select deadline date and also status of the bug. Deadline in Bugzilla usually gives the time-limit to resolve the bug in given time frame.



Orig. Est.:	Current Est.:	Hours Worked:	Hours Left:	%Complete:	Gain:	Deadline:
0.0	0.0	0.0 + 0	0.0	0	0.0	

Summarize time (including time for bugs

January 2015

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Attachments
Add an attachment (proposed patch, testcase, etc.)

Additional Comments:

Status: CONFIRMED
CONFIRMED
IN_PROGRESS
RESOLVED

James 2015-01-07 02:50:31 PST Description [reply] [−]

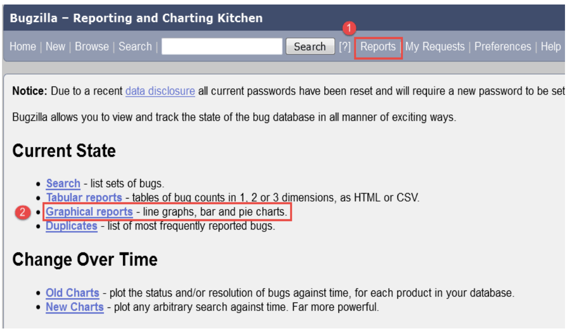
The widget gears are twisted at the end and not showing correct signal

Collapse All Comments
Expand All Comments

select deadline for your bug-report
You can select bug status over here
Save Changes

Figure10 Enter deadline and status

Step 6: To browse already created bug reports we can use Browse button in main window. Also you can get your report in different formats like tabular and graphical.



Bugzilla – Reporting and Charting Kitchen

Home | New | Browse | Search | Search [?] | 1 Reports | My Requests | Preferences | Help

Notice: Due to a recent data disclosure all current passwords have been reset and will require a new password to be set

Bugzilla allows you to view and track the state of the bug database in all manner of exciting ways.

Current State

- Search - list sets of bugs.
- Tabular reports - tables of bug counts in 1, 2 or 3 dimensions, as HTML or CSV.
- 2 Graphical reports - line graphs, bar and pie charts.
- Duplicates - list of most frequently reported bugs.

Change Over Time

- Old Charts - plot the status and/or resolution of bugs against time, for each product in your database.
- New Charts - plot any arbitrary search against time. Far more powerful.

Figure 11 Browsing of bugs in Bugzilla

DOI: 10.15680/IJIRCCE.2016. 0406298

You will get graphical report like below one:

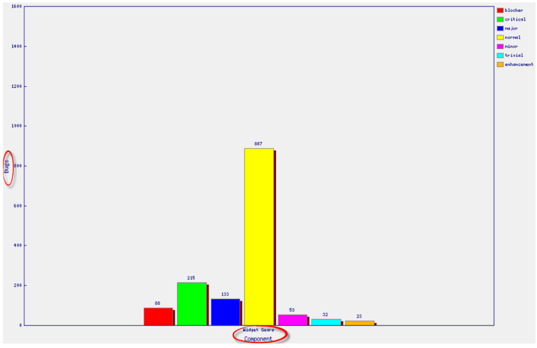


Figure 12 Graphical report Bugzilla

D. Comparision of tool

Criteria	QTP/UFT	Selenium	WATIR	Bugzilla
OS Support	Windows 8/8.1/7/XP/ Vista(No other OS)	Windows, MAC OS X, Linux, Solaris)	Windows, Linux, MAC OS X, Solaris(Need JSSH compiled)	Window Linux, MAC OS X
Browser support	IE(Version6-11), Firefox version(3-24), chrome (up to version24)	Firefox, IE, Chrome, Opera, Safari	Firefox, IE, Chrome, Opera, Safari	Firefox, IE, Chrome, Opera, Safari
Non browser based App support	Yes	NO	NO	Yes
Device support	Support iOS, Android Blackberry and Windows phones	Two major mobile platform iOS and Android	Two major mobile platform iOS and Android	Driver available for iOS(iPhone and iPad)
License	HP licensed product	Open source and free	Open source and free	Open source and free
Ease of support	Dedicated HP support	User and open source community support is available	Limited support on open source community	Open source community support is available
Language	VB scripting	Java, C#, Python, Ruby,	Ruby	Perl
Framework support	In built capability to build frameworks such as keyword based, data driven, and hybrid	JUnit, NUnit, RSpec, TestNG, unittest	Ruby supported framework Cucumber, Rpsec	Perl supported framework like catalyst
Continuous integration	Can be achieved through Jenkins	Achieved through Jenkins	Achieved through Ruby Script	Can be achieved using tools like CruiseControl and tinderbox

Table 1 Comparison of tools

DOI: 10.15680/IJIRCCE.2016. 0406298

V. BENEFITS AND DRAWBACKS OF AUTOMATION TESTING

A. Benefits of automation testing

It is way faster than the manual testing. Even highly trained manual testers will not outdo an automated suite of test cases. This is particularly valuable when rushing regression testing against the clock after any changes are introduced close to a project deadline.

Wider test coverage of application features. Items such as performance and stress tests can be covered only by automation scripts. Therefore, by automating your test cases you will expand your testing coverage.

Reliable in results. Functional Automation tests will always validate that their checkpoints fulfil the expected results.

Improves accuracy. Even the most conscientious tester will make mistakes during monotonous manual testing, since repetitive tasks numb the eye. Automated tests perform the same steps precisely every time they are executed and never forget to record detailed results.

Ensure Consistency. Automated test suits assure that the type of testing you make will be incremental as time goes by, yet consistent from the very beginning of the project.

Test scripts are re-usable. With basic maintenance, test scripts may be used across different platforms. Therefore, you will not need to start from scratch every time you need to test your product over a new OS or device.

Human Intervention is not required while execution. The execution of automated test suites can be triggered periodically and report any bugs or inconsistencies via email or many other means. Therefore, no human intervention is required.

Increases Team Efficiency. If you have your DEVs share the access to automated tests written by your QA team, they will be able to catch problems quickly before sending a stable build to QA. This means that their confidence in their builds will be higher, and the cycles of bug fixes will be way shorter.

Helps you test your product frequently and thoroughly. Testing areas of the application that are outside the core functionality may be hard when you are short of time. Once you have automated your test cases, ALL features will ALWAYS be included in your test plan. This means minimizing the chances of not catching bugs when you deliver builds under pressure.

More cycles of execution can be achieved through automation. Given that the execution time of an automated test suite is way lower than manual, you can achieve a higher number of executions in the same amount of time. This is particularly helpful in AGILE contexts, where lots of changes are made in a small period of time.

B. Drawbacks of automation testing

Though the automation testing has many advantages, it has its own disadvantages too. Some of the disadvantages are:
1) **Proficiency is required to write the automation test scripts.** Tester should have good knowledge of particular scripting language.

2) **Debugging the test script is major issue**. If any error is present in the test script, sometimes it may lead to deadly consequences.
3) **Test maintenance is costly in case of playback methods.** Even though a minor change occurs in the GUI, the test script has to be re-recorded or replaced by a new test script.

4**) Maintenance of test data files is difficult,** if the test script tests more screens.

5) **Automated Functional Tests CANNOT be used to validate user experience in regards to ease of use, GUI friendliness, appearance and aesthetic consistency across the application.** The testing of those items can only be

DOI: 10.15680/IJIRCCE.2016. 0406298

automated indirectly, for example, searching the GUI for certain items with specific features. In other words: the acceptance criteria for items accounting for the user experience MUST be covered manually by humans at first. Then the "positive" impressions from the user can be translated into checkpoints, whose aspect and presentation can be used as validation points in an automated test suite

VI. Conclusion

Test automation can be powerful tool for organization if it is used with correct planning. It will help to deliver products in less time and money without compromising quality of product. To ensure successful software automation test automation should be involved at all phases of software development and people involved in this process should have sufficient knowledge about it.

References

[1] Tom Wissink and Carlos Amaro**, Lockheed Martin, Successful Test Automation for Software Maintenance ,** Integrated Systems & Solutions 22nd IEEE International Conference on Software Maintenance (ICSM'06) 2006 IEEE

[2] TeddieStenvi**, An automated testing strategy targeted for efficient use in the consulting domain**, Master Thesis Software Engineering Thesis no: MSE-2007:18 June 2007

[3] Patrice Godefroid, Peli de Halleux, Aditya V. Nori, Sriram K. Rajamani, Wolfram Schulte, and Nikolai Tillmann, Microsoft Research Michael Y. Levin, **Automating Software Testing Using Program Analysis,** Microsoft Center for Software Excellence September/October 2008 IEEE Software

[4] TeemuKanstrén. **A Framework for Observation-Based Modelling in Model-Based Testing.** Espoo 2010.VTT Publications

[5] XiangFengMeng, **Analysis of Software Automation Test Protocol,** 2011 International Conference on Electronic & Mechanical Engineering and Information Technology

[6]C.Bhuvana1,K.Munidhanalakshmi2, r.R.MahammadShafi3**T able - Driven Approach for Automated Testing of Web Applications**,Volume 3, Issue 7, July 2013 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

[7] Janani. V, Krishnamoorthy .K, Revathi. K**,Implementation of data driven automation testing of web applications using selenium**, IJARSE, Vol. No.4, Special Issue (03), March 2015

[8] Rajvir Singh, Preeti, **Review of model based approach for automating the test case generation for Object Oriented Systems**International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 4 Issue 6 June 2015, Page No. 12774-12780

9] Harsh Bajaj, Technical test lead ECSIVS, Infosys **Viewpoint, Choosing right automation tools and framework critical to project success**

[10] Amit Srivastava, **Test Automation and Software Development** Technology Review ©Tata Consultancy Services.

[11] Mark Fewster and Dorothy Graham **Software Test Automation Effective Use of Test Execution Tools** ACM PRESS BOOKS

[12] Douglas Hoffman **Test Automation Architectures: Planning for Test Automation,**

[13] Doug Hoffman **, A Course on Software Test Automation Design**

[14] Linda G. Hayes, **The Automated Testing Handbook**

[15]Michael T. Grater, **Benefits of Using Automated Software Testing Tools to Achieve Software Quality Assurance**

[16]D.-Graham-M.-Fewster, **Experiences-of-Test-Automation-software-Case-Studies**

[17]http://www.cs.colorado.edu/~kena/classes/5828/s12/presentations/testing-frameworks-by-gayat.html

[18]http://www.mactech.com/articles/mactech/Vol.13/13.10/SoftwareTestAutomation

[19] http://automated-testing.com/PATfinal.htm

[20]http://safsdev.sourceforge.net/FRAMESDataDrivenTestAutomationFrameworks.htm

[21]http://www.guru99.com/bugzilla-tutorial-for-beginners.html#2

[22]http://modernperl.net/developers/web-application-development.html

[23] https://wiki.mozilla.org/Bugzilla:Addons

[24]http://www.softwaretestinggenius.com/pros-and-cons-of-automated-testing