

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/267823533>

AUTOMATED API TESTING APPROACH

Article in International Journal of Engineering Science and Technology · March 2012

CITATIONS

5

READS

4,081

4 authors, including:



Sunil Bangare

Sinhgad Technical Education Society

30 PUBLICATIONS 126 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Brain Tumor detection from MRI [View project](#)



Usable Authentication Systems [View project](#)

AUTOMATED API TESTING APPROACH

SUNIL L. BANGARE¹, SEEMA BORSE², PALLAVI S. BANGARE³,
SHITAL NANDEDKAR⁴

^{1,2,3,4} Department of Information Technology, STES's Sinhgad Academy of Engineering,
Pune-48, Maharashtra, India.
sunil.bangare@gmail.com

Abstract

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. With the help of software testing we can verify or validate the software product. Normally testing will be done after development of software but we can perform the software testing at the time of development process also. This paper will give you a brief introduction about Automated API Testing Tool. This tool of testing will reduce lots of headache after the whole development of software. It saves time as well as money. Such type of testing is helpful in the Industries & Colleges also.

KEYWORDS: *Software testing, API (application programming interface), Manual Testing, Automated Testing, API testing.*

1. INTRODUCTION

Software testing can provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs [1]. We can verify or validate the software product as per the user requirements, we can check that it is working as expected or not. In general we can perform the testing after defining the whole requirements as well as the coding part has been completed. This will produce more headache, requires more time & money also.

In this proposed work testing will be done at the time of development process for reducing time, money & headache. Such type of testing will not reduce 100% errors but reduce 70%-80% errors which are hard to recover after whole development process.

2. GENERAL APPROACH OF MANUAL TESTING

This is the oldest type of software testing. In such type of testing tester can perform the manual test without any help of the test automation. Manual testing is a laborious activity that requires the tester to possess a certain set of qualities; to be patient, observant, speculative, creative, innovative, open-minded, resourceful, unopinionated and skillful [1].

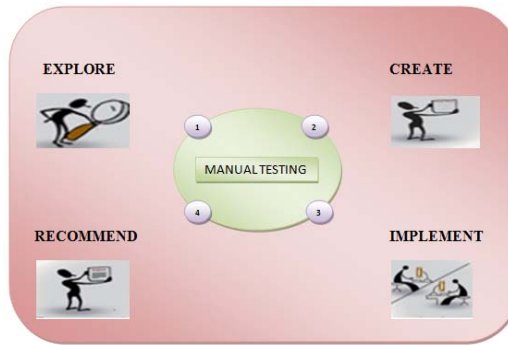


Fig. 1 Manual Testing

3. LIMITATIONS OF MANUAL TESTING

1. Performing manual testing becomes very bored for tester. It requires checking every link on the site and every aspect of the product under test.
2. It requires so much time for testing. We have to do same operations again & again. Regression testing is being done for checking errors after making changes on the one part of software have no ill effects on the other part of software.
3. Manual testing is time consuming, error prone and requires lot of infrastructure and manpower [1].

These limitations can be overcome if the testing process is *automated*. The automated API testing tools reduce manual testing to a large extent and the testing can be done automatically as given in name.

4. AUTOMATED TESTING

The process of automating the manual test is automated testing. Currently we are using the automated testing. This type of automated testing will reduce the need of manual tasks which is very bored & time consuming. In automated testing, we need not to do same task again & again. This type of testing saves lots of time.

This automated testing includes:

- 1) Detailed test cases, including predictable "expected results", which have been developed from Business Functional Specifications and Design documentation.
- 2) A standalone Test Environment, including a Test Database that is restorable to a known constant, such that the test cases are able to be repeated each time there are modifications made to the application.

Functional Testing, Regression Testing, Stress Testing, Exception or negative Testing, Performance Testing & Load testing are the some types of the Automated Testing. Automated Testing is very useful because it is reusable in which we can reuse different versions of applications even though the user having change in the interface. It is reliable because we can reduce the human errors. This testing is repeatable in which we can check that how software perform under repeatable test cases.

Also another advantage of this kind of testing is that the system can continually be regression tested as it is being developed [8]. The Tool is mentioned as "Automated" because testing is done automatically once we gave the inputs, expected outputs & API (Code or procedure which you want to test), after that we get the reports regarding errors if present in that API (application programming interface).

This is the challenge to measure the quality of objects oriented software modularization also. Modularization of object oriented code is distribution of the software in to modules and these modules should communicate with each other through some application programming interface (API) [9]. The main problem is of communication between the modules. Generally this should be done through some application programming interface (API) [10].

5. API TESTING

Application Programmable Interfaces (API's) are collections of procedures & functions that can be used by other applications to fulfill their functionality [8]. "A collection of methods wrapped in a library or DLL that

interact to meet a functional requirement is a component. Rather than a developer having to call each method individually to achieve some usually repetitive functional outcome from the library that functionality is usually exposed via a call to a single API" [11]. Each API is supposed to behave the way it is coded, i.e. it is functionality specific. These APIs may offer different results for different type of the input provided. The errors or the exceptions returned may also vary. However once integrated within a product, the common functionality covers a very minimal code path of the API and the functionality testing / integration testing may cover only those paths. By considering each API as a black box, a generalized approach of testing can be applied. But, there may be some paths which are not tested and lead to bugs in the application. Applications can be viewed and treated as APIs from a testing perspective [8].

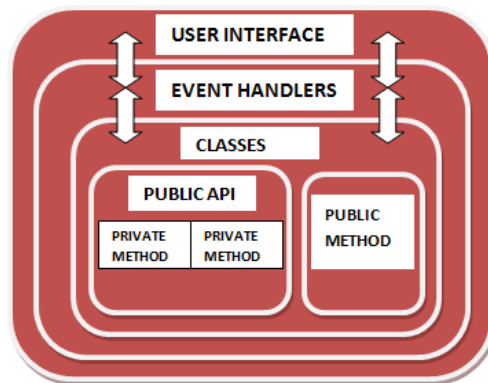


Fig. 2 API Testing

How testing the APIs is different than GUI testing?

- 1) It requires knowledge of inner workings.
- 2) Access to source code - The availability of the source code would help tester to understand and analyze the implementation mechanism used [8].

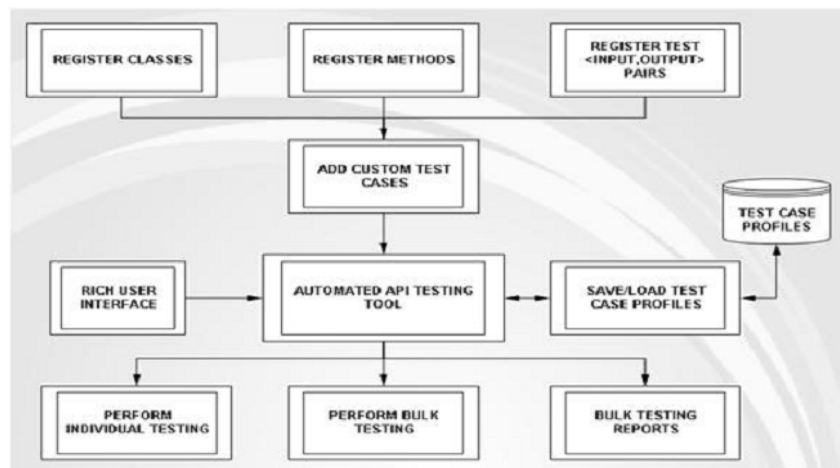


Fig.3. Overview and Approach of Automated API testing

6. GUI TESTING

If we have complete GUI of application then we can test that GUI with help of this tool. GUIs are fundamental components in today's software and have become an ideal way of interacting with computer programs. GUIs are important because they can make our applications easier to use by providing a front-end to the underlying software system [6].

GUI testing is vital to make the entire system safer and more robust. GUIs, even the simplest ones, enclose some level of complexity. Any complexity in software needs to be tested because untested code is a potential source of bugs [4]. In spite of its importance, the development community has been slow to include GUI testing as a core practice, primarily because GUI testing is difficult [5]. So Automated API Testing Tool includes GUI

testing also if we have completed GUI. We can directly test that GUI and get the appropriate reports and modify accordingly.

7. CONCLUSIONS

1. In Software Organizations, AUTOMATED API TESTING is used to do the manual testing automatically at the time of development. Developer can add his own test cases & he can test the module. Accordingly reports will be generated if API (any function or procedure) contains any error.

2. The appropriate error message will be displayed in the report. This is useful because developer can change his API (Function or Procedure) according to report generated by the System. We can reduce maximum testing work of tester at the time of development itself. It will also reduce the time which is required for testing after whole development of product or Software.

3. This Generic API Testing Tool will allow users to Define Test Cases, Manage Test Cases, Perform Individual Testing, Perform Bulk Testing and Manage Test Reports.

Thus Automated API Testing can be done using this approach.

References

- [1] R. S. Pressman, "Software Engineering", ISBN 0073655783.
- [2] Boris Beizer, "Software Testing Techniques".
- [3] Ilene Burnstein, "Practical Software Testing".
- [4] A. Ruiz, and Y. Price, "Test-Driven Development with TestNG and Abbot", *IEEE Software*, IEEE Computer Society, May/June 2007, pp. 51-57.
- [5] P. Hamill, *Unit Test Frameworks*, O'Reilly, 2004.
- [6] IEEE paper "GUI Testing Made Easy" By Alex Ruiz and Yvonne Wang Price.
- [7] "Testing fundamentals" [WWW] available from <http://softwaretestingtimes.com>
- [8] Ajitha and Amrit Shah, "SoftTReL Software Testing Guide Book Part1".
- [9] S. L. Bangare, A. R. Khare, P. S. Bangare, "Measuring the quality of Object oriented software Modularization: Defining metrics and algorithm", *International Journal on Computer Science and Engineering (IJCSE)*, ISSN: 0975-3397 Vol. 3 No.
- [10] S. L. Bangare, A. R. Khare, P. S. Bangare, "Code parser for object Oriented software Modularization", *International Journal of Engineering Science and Technology*, ISSN: 0975-5462, Vol. 2 (12), 2010, 7262-7265.
- [11] S. L. Bangare, A. R. Khare, P. S. Bangare, "Quality measurement of modularized object oriented software using metrics", *ACM-International Conference ICWET-2011 at Mumbai*, ACM 978-1-4503-0449-8/11/02, ISBN: 978-1-4503-0449-8.