A Framework for Continuous Regression Test

Hanuma Prasad T\textsuperscript{1}, DR. Shantharam Nayak\textsuperscript{2}
\textsuperscript{1}MTech Scholar, \textsuperscript{2}Professor - RV College of Engineering, Bengaluru - 59

Abstract- Continuous Regression Test Framework is an approach towards performing Continuous Regression Testing for Various releases of any operating Systems. These releases are continuous and often come to release frequently if it’s an open source operating system like Linux. The operating systems must be validated before releasing to the market for live experience of the end user. The Regression Testing framework proposed here helps to deal with this validating or testing before making it available to the users. The framework is automated rather than going for manual testing. The whole purpose is to build a framework to automate regression testing and log results of the tests performed on the dashboard for analysis.

I. Introduction

Over the years of software era, Testing of software or an application plays a key and vital role during software development life cycle (SDLC). Major effect will be utilised for testing in SDLC. The fact implies that testing software is time consuming and is critical phase as it is the final arbiter after which the software should be delivered to the end user. Producing high quality software systems has been one of the most important aspects in software development. In this direction software architecture and software testing play a key role [13]. Testing a software means you validate the software with the requirements that are collected during requirement phase of SDLC. Before delivering the software to the client there is a lot of regression test that should be performed until the product is accepted by the user and is satisfied. As there is no system that is cent percent perfect, there is a need for lot of continuous testing and reviews that should be carried out from the end users product perspective.

Automation is preferred as it has more advantages, time efficient and resource beneficial compared to manual testing. The proposed framework uses various open sources like GITHUB, JENKINS, and AUTOTEST. The framework proposed in fig 1 can be used to test various releases of Operating System which are frequent and later analyse the results after logging into particular repository.

Fig 1: Overview of framework
II. Background Research

Real time systems are designed to evolve over time due to requirements of new changes and technology improvements. Software Testing can consume between 50 to 80 percent of project efforts [10]. The current trends of organising software testing activities, practices and testers experience are discussed in [12]. Regression Testing must make sure that no functionality is affected by the new changes. A good Regression test framework must be able to provide the flexibility of new requirements and technologies are fit as per the requirements. Manual testing are typically slow and error prone.[1]. Testing involves modification of various modules according to the requirements and in indispensable[14].

There are various testing approaches for various domains. For example an agent-based approach for Web based application testing has been proposed in [16],[17]. And even there are test tools using intelligent agent technology to test the software. The Process of Regression Testing is expensive in software testing activity and maintenance. This is performed on programs that are modified after additional changes to make sure that it does not affect the unmodified code of the software [2].

Any software that is designed will consider three important factors namely: Testability Maintainability Reliability [8]. Regression testing considers these as important aspects. 80% of overall testing cost is consumed by regression testing and will take 50% cost of overall software maintenance. Regression Testing amounts for most of the software maintenance [5] [9]. This implies that regression test in software development plays a important role in software development. To reduce the cost of Regression Testing we can go for selective regression testing. A variety of regression testing techniques have been proposed in various papers [3] [4] [6] [7]. Manual Testing of Regression Test is typically slow and error prone[1] and as a result of this there is a need to go for automating the regression testing which has several benefits over manual testing. Few of them include:

- Optimisation of test resource utilisation.
- Speed up testing process.
- Automate process of testing and reduce laborious work.
- Provide uniform test framework for different test environment.

III. Methodology

The framework must be able to download the latest release of the operating system for testing. It must be able to setup test environment and start executing tests on the various combination of software and hardware platforms. Once the tests cases are executed it should be able to log the results for later examining and analysis. The framework proposed here involves usage of different open sources like GITHUB for source code Managment, JENKINS for deployment of the application and AUTOTEST Framework for the test case development. Without test cases testing would not be possible [11]. All these sources are integrated into framework and used.

The end product would be a Regression Framework which is automated to continuously check for releases of the operating system and start downloading the releases to perform testing on different combination of hardware and software. It must be capable of loading the different OS on various hardware environments like bare metal, Virtual and non virtual environments.

Open source operating systems vary according to the developers across globe. There are multiple layers of software in which modifications have to be done. This includes application, Libraries etc. A good user experience is the ultimate goal of any application development. The physical resources must be efficiently used underlying this framework.

IV. Implementation

The implementation of continuous regression test framework involves following techniques
The main focus of implementation is to realize the requirements and design into software code which can be effectively and efficiently used for specified purpose. The framework must facilitate the downloading of latest available release of Operating System into a specific repository as required. This can be achieved by using any programming language, Preferably Object oriented. Once the releases are available, set up the test environment i.e. hardware and software environments. We need GITHUB for source code management across different developers and JENKINS Server to deploy the application and continuously check for the latest releases of an operating system and setup operating environment like bare metal, virtual or non virtual environments. Once the setup is done, start executing test cases on the operating system under different hardware platforms and publish results on the dashboard.

As the proposed framework methodology in fig 2 involves usage of lot of open sources and it is an efficient approach in terms of cost and even better performance is achieved since open sources has contributors across the globe and issues will be fixed soon. And this framework is automated with Jenkins server which is again an open source.

![Methodology for framework](image)

**Fig 2: Methodology for framework**

**V. Conclusion**

There are many contributors to the open source Operating System across the globe. As a result of this, open sources will be having an modification or update continuously throughout the existence of an open source OS. As there is an update to the open source continuously framework is needed which checks for the update and carry on the task ahead of the release like start testing the released OS.

The proposed framework provide a method for downloading the latest release of the OS.Install on the system and start executing under different operating environments like virtual or non virtual environments. It involves various test cases that perform various different operations on the OS and Hardware of the System. The entire framework is automated as it should continuously monitor the availability of the release and start downloading and execute the test cases and publish the result.
VI. Acknowledgement

I would like to thank Dr. Shantharam Nayak for his valuable suggestions, expert advice and moral support in the process of preparing this paper.

References