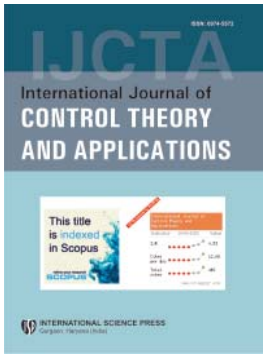

International Journal of Control Theory and Applications

ISSN : 0974–5572

Volume 9 • Number 40 • 2016

Reducing the Number of Test Cases using Prioritization Technique

S. Selvakumara Samy[a] Ayyagari Abhishek[b] Karthikbyju[b] Saisantosh[b] and Naga Surya Narayana[b]

[a]*Asst prof (O.G), Department of software Engineering, SRM university*

[b]*Student, Department of software Engineering, SRM university*

Abstract: The use of Dependency algorithm not only increases the efficiency of the project but also reduces the time complexity a lot due to pre-prioritization by either the consumer or the develop. As dependency algorithm can be used in many scenarios and implementing all the scenarios takes a team of 11-12 members what we did is take up a single scene and apply the dependency algorithm. The situationmade is job search in a particular portal that is "Dice.com". We are using Seleniumintegrated intoChrome.Using this project the developer can reduce the time complexity of his project to a noticeable amount thus increasing the productivity.Test case prioritizing is done here which will enhance the efficiency of the program/software as a whole.This will help the end users to access the software and the test cases in a much simpler way thus helping them to understand the process.

Keywords: *Selenium, dependency algorithm, time complexity, Regression testing.*

1. INTRODUCTION

With the scale and multifaceted nature of programming framework getting to be plainly bigger and bigger, the issue of programming unwavering quality which draws in designers, is an ordinarily huge research. Programming testing, going for discovering mistakes and setting up the certainty of programming quality, is a strenuous and costly process expending no less than half of the aggregate programming cost [1]. To build the viable of experiment upkeep inside constrained time and assets, experiment prioritization can be performed. To date, a substantial number of prioritization systems have been advanced by research foundations. Keeping in mind the end goal to give prior input and before imperfection settling to analyzers, experiment prioritization strategies [2-4] reorder tests cases, booking tests cases with the most noteworthy need to accomplish some execution objectives prior in the testing procedure. It is additionally revealed that these reviews can altogether build the viability of testing and enhance the rate of blame identification as ahead of schedule as could reasonably be expected. While this exploration has gained extensive ground in programming testing, one appealingly significant issue has been neglected. The conditions and relations between experiments, particularly in utilitarian test suites, ought to be thought about for executing the test effectively and properly. It is likely that these conditions are firmly identified with the coupling and associations between the parts making up programming frameworks. Rothermel, et al., [5] gives a far reaching theory that testing the parts of the framework which incorporates greater many-sided quality as quickly as time permits may enhance the deformity recognition rate.

Along these lines, the strategy that appointing the experiment with the more conditions to the higher need may improve the probability of finding issues prior in the product testing cycle. We propose to address this need by making and exactly contemplating prioritization methodologies in view of the relationship of conditions between experiments, which we call reliance based prioritization (DBP). We gather some data about testing reliance, and receive a weighted profundity first pursuit calculation to discover a request with the best rate of blame identification.

2. RELATED WORK

In this section, we present an overview of background work of current test case reduction techniques for the single scenario "**JOB SEARCH".**We have taken a website "DICE.com" which is a US job searching portal. Now We have asked a few candidates to fill in the google form created by us whose link is :https://goo.gl/forms/OmxrcTOXhQe7nUSB3. Once the candidate submits the Google form, We get his info, and then we write it in our database. The database we are using is an MSExcel sheet with the byte code SDK embedded on it.

Now we use the dependency algorithm which is embedded in the selenium driver for Chrome. We just need to give the criteria and the number to which the test cases must be reduced. The screenshot related to the database is provided below:

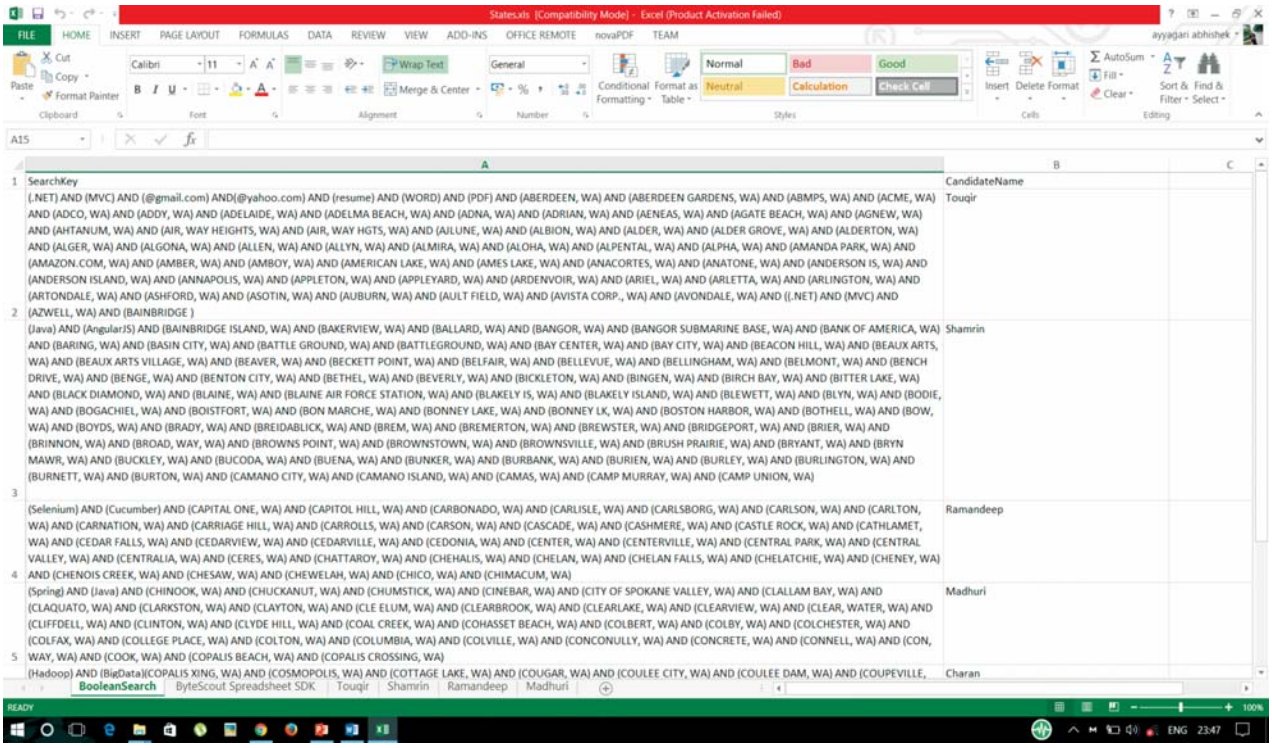

Figure 1

3. USER INTERFACE

Now coming to the user interface, we have two main elements. One is the website itself and the second one is Command Prompt. Screenshots of both the itemsare provided below. So once the app.config file is copied and pasted on to the bin directory, we can start the first user interface that is the "Command Prompt." Now here the user enter the required code to switch the second interface. In the second interface user cannot do anything but just wait for the program to end.

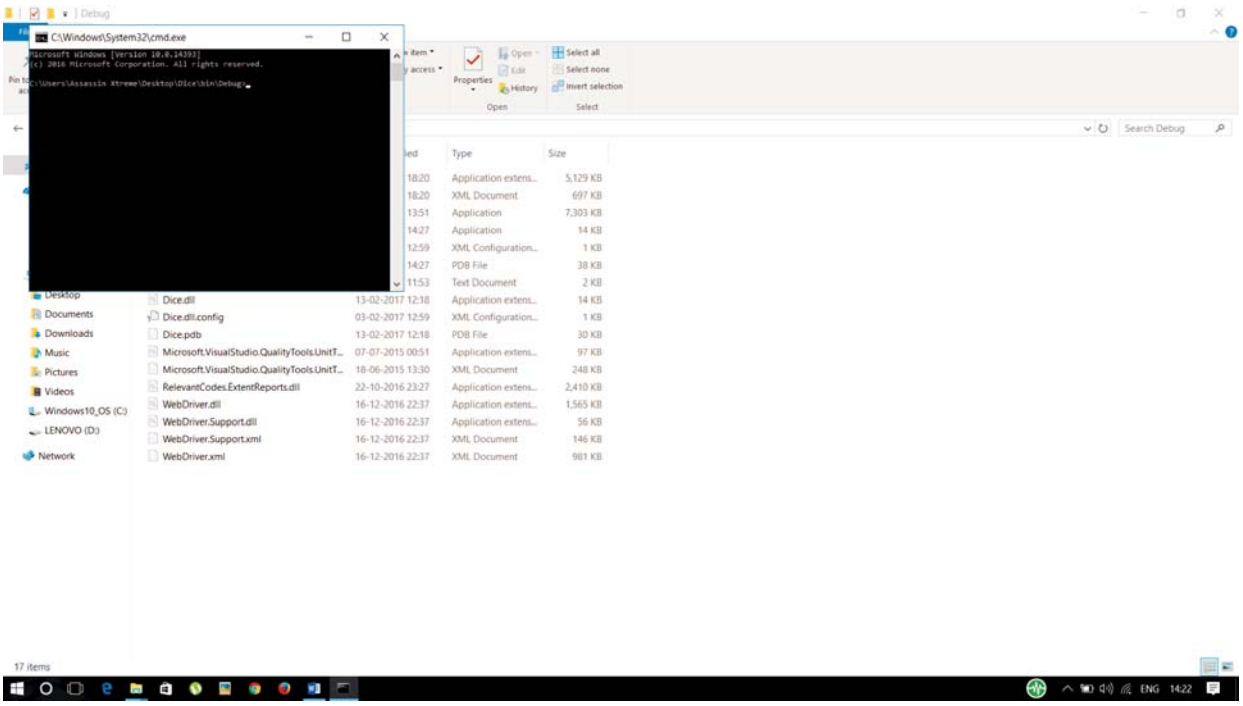

Figure 2

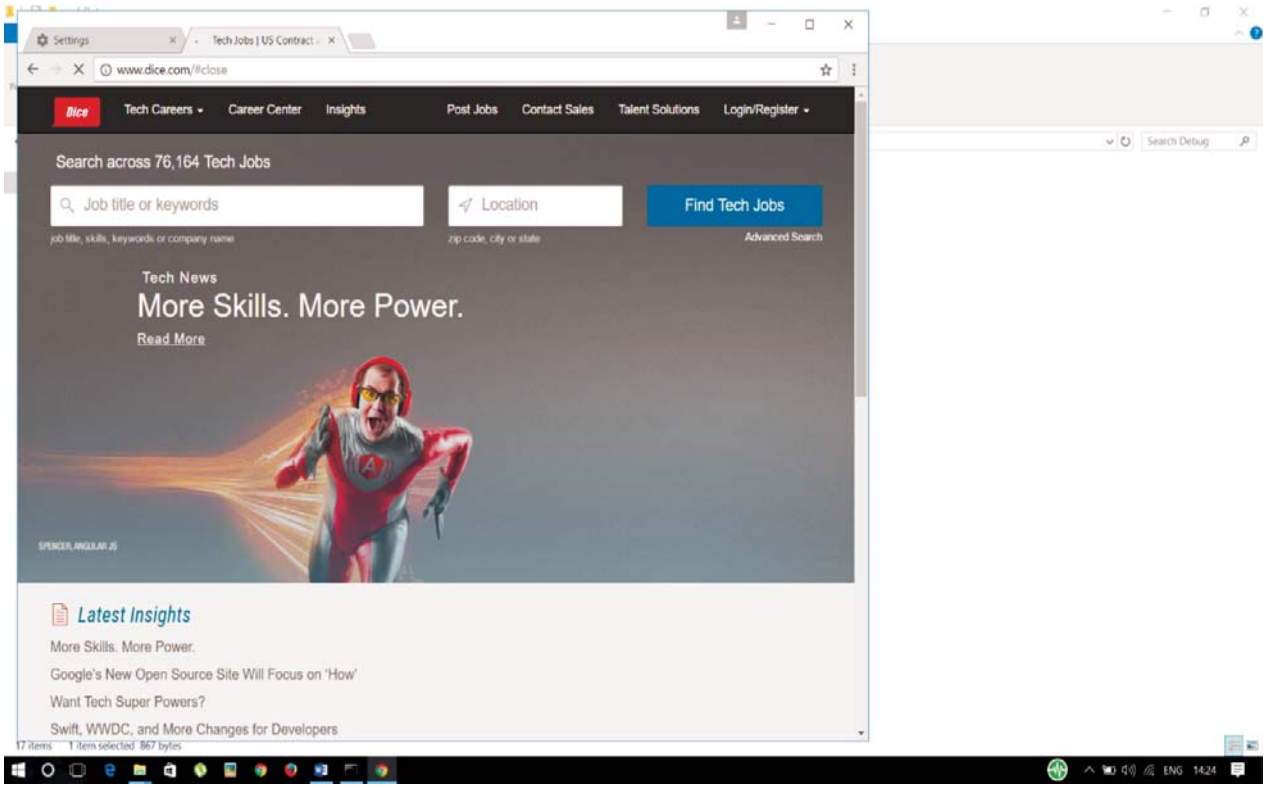

Figure 3

4. PRIORITIZATION TECHNIQUE

Dependencies between Test Cases In this segment, we exhibit the proposed prioritization procedure and the comparing calculation for taking care of the issue of requesting the experiments.

3.1. Reliance Structure A dependency structure is regularly determined by a coordinated non-cyclic chart (DAG), G = (V, E), where V is an arrangement of hubs and E is a gathering of the circular segments between these hubs. In this paper, V speaks to an arrangement of experiments, and E demonstrates the conditions between experiments.

5. WORKING

Let's see the detailed operation of the project step by step:

Step 1: First we need to collect the data from the google forms in the shape of a spreadsheet.

Step 2: Now the data is entered in the excel sheet which is inside the test data folder of the software. The path is as follows:

C:\Users\Assassin Xtreme\Desktop\Dice\TestData

Step 3: Now copy the app.config file and paste it insidethe bin folder. The path is as follows:

C:\Users\Assassin Xtreme\Desktop\Dice\bin\Debug

Step 4: Now launch cmd from the previous path and type in the following command:

ConsoleTestRunner "dice.dll" mstest

Step 5: Wait for the program to finish the test and then open the sheet inside the testdata folder

You can see that separate sheetsare created in the name of each candidate.

These are the test results for the input given.Now you can see which job was posted and when it was posted.By this software you only get the latest job posted at dice.com so as to reduce the time wasted on jobs posted a long time ago

6. CONCLUSION

Regression testing is done on the test suite by applying one of its techniques that is test case prioritization which gives maximum number of faults and also provides effectiveness to the software. In this paper, problem is formulated to discover the maximum number of defects by prioritizing the test cases using model based dependencies. In future, test cases are prioritized with model based dependencies and efficiency of technique will be evaluated with APFD matrixes

REFERENCE

[1] G. Rothermel, R. H.Unteh, C. Chu and M. J. Harrold, "Prioritizing Test Cases For Regression Testing", IEEE Transaetions on Sonware Engineering, (2001).

[2] S. Elbaum, A. Malishevsky and G. Rothermel, "Incorporating Varying Test Costs and Fault Severities into Test Case Prioritization", Proceedings of the 23rd International Conference on Software Engineering, (2001) May 12-19, Ontario, Canada.

[3] A. Srivastava and J. Thiagarajan, "Effectively prioritizing tests in development environment", Proceedings of the International Symposium on Software Testing and Analysis, (2002).

[4] W. E. Wong, J. R. Horgan, S. London and H. Agrawal, "A study of effective regression testing in practice", Proceedings of the 8th International Symposium on Software Reliability Engineering, (1997) November 2-5, Albuquerque, NM. [5] G. Rothermel, S. Elbaum, A. Malishevsky, P. Kallakuri and B. Davia, "The Impact of Test Suite Granularity on the Cost-Effectiveness of Regression Testing", Proceedings of the 24rd International Conference on Software Engineering, (2002) May 25-25, Orlando, FL, USA. [Old, "Prioritizing Test Cases For Regression Testing" IEEE vol. 27 no.10, 2001.