

Redefining Regression Testing and Details under It Regression for Fix and Regression for Risk

Daniel L Asfaw

PhD Student in Computer Science and Software Testing, Inter-University Program, Universidad Azteca and Universidad Central de Nicaragua (UCN), Mexico City, USA

Abstract Regression testing has two shared goals. The prime one would be to help building a bug free application, while the other equally important goal could be to maintain the healthiness of an existing system. If the emphasis is on the first objective, that testing would conventionally be named as regression for fix, and if otherwise, it is termed as regression for risk. The former implicates activities destined to verify the entire software development process and validate the end product. The later centers on retaining the wellbeing of an existing product by periodically checking on its quality, to make sure if it still meets standards and requirements. This article, which is a part of an ongoing PhD dissertation in computer science and software testing in an Inter-University Program at Universidad Azteca and Universidad Central de Nicaragua (UCN) in Mexico City – Managua Nicaragua, attempts to redefine regression testing focusing on the aforesaid two details. To this end, related literatures have been reviewed. Best industry practices have also been assessed and analyzed. 50 expertise from various levels and walks of skill sets have been involved through questionnaire, discussions, and interviews. The finding reaffirms regression for fix as a functional testing directed to examine if the fix is done properly, and if the defect is resolved and has no aftermaths. By the same token, it redefines regression for risk as a recurrent functional testing routed to check on the ability of an existing software to meets standards and expectations. Regression testing, therefore, is an umbrella term encompassing, at least, both of the above.

Keywords Regression Testing, Regression for Risk, and Regression for Fix

1. Introduction

Information technology is promptly surfacing out. The world has rapidly moved from 1 web-based application in 1991 to over 700 million in 2013. This number is expected to hit a billion by the end of June 2014 (Total number of websites, 2014). In result, testing has turned out to be a key signal in everyday life contributing to the incidences of utterly novel industries and dramatic changes in quality assurance process.

In this instantaneous evolvement, software testing, has established itself as a rock-solid methodology of verifying the entire process of any given software development and validating the end product (Hanip, 2014). Some research indicate that there are over 150 diverse types of software testing put together under various categories (Bertolino, 2003). Among them, regression testing is widely practiced in the industry (Samuel R., 2014). The crux of this article cores on the two essentials of regression testing: regression for fix and regression for risk. Both seem to be either a

regular or a periodical practice in the mainstream (Verma, 2014). Hitherto, very little is researched and know about them.

The purpose of this study is to contribute towards narrowing down this research gap exhibited in the field. To this end, a significant amount of review of relevant literatures have been made. Contemporary industry practice has also been captured through questioner, discussions, and interviews which was held with 50 expertise in the field.

The paper has five crucial sections. The first one reflects on the research purpose, hypothesis of the study, along with, the statement of the problem, and limitation of the study. The second part reviews relevant literatures in addition to assessing and analyzing contemporary best industry practices. The third part gives details on the methodology used in the course of this study. Finally, the fourth part divulges the discussion made on its findings while the last one wraps up the discussion with a brief conclusion.

2. Statement of the Problem

Google defines testing as a deliberate action or trial to find out how well something works (Google Dictionary, 2014). Likewise, software testing is a process of running an application with the intent of finding defects (Hanip, 2014).

* Corresponding author:
talktothedaniels@gmail.com (Daniel L Asfaw)
Published online at <http://journal.sapub.org/computer>
Copyright © 2014 Scientific & Academic Publishing. All Rights Reserved

Regression testing is not any different. Nevertheless, scholars in the testing industry do not seem to settle in one clear designated definition of regression testing and the two details (regression for fix and regression for risk) under it (Rudd, 2014).

First of all, regression testing is jumbled and confused with retesting, even so, with dry running and debugging a script¹. What's more, there appears to be a dearth of vibrant set of distinctions assembled between regression for fix and regression for risk (Hanip, 2014). They are taken as one and identical by most of us while having quite a lot of inconsistencies between them². This exhibits the fact that regression testing is either not well delineated, or it might have undergone through fairly a significant amount of misperceptions, even by the stakeholders.

The researcher thinks that invoking discussion among the interested parties might assist to acquire more about the existing awareness on the subject matter³. It might also expedite ventures to connect minds and hearts among the concerned bodies. By so doing, an attempt could be made to draw a kind of courteous consensus towards formulating a straight forward definition for regression testing, in general and its details in particular⁴.

3. Research Purpose

This article focuses on exhibiting the contemporary awareness among the stakeholders in the software testing industry about regression testing and its details and make an attempt to redefine same based on consensus.

4. Hypothesis of the Research

This research assumes that there is no shared and corporate awareness or designation about regression testing and the two details under it⁵.

5. Limitations of the Research

All the interviewees of this research are from the scholars, industry experts, other stakeholders, and authorities in the software testing industry. Though, most of them are from immigrants from different countries and continents, such as India, China, Africa, and Europe, all of them are currently residing in the USA. And yet, the researcher does not claim to have included higher volume of interviewees that has a good representation of the stakeholders at a global level.

In fact, it would be impractical and unruly if it were tried. The number of interviewees is limited to 50, in fact, constituted from both genres and range of expertise and experiences. This number is not good enough to signify a good sum of expertise in the field. Nevertheless, an attempt has been made to review the works of these other authorities and stakeholders through their articles, books and publications.

It goes without saying that, the aforesaid points institute some of the chief areas of the limitation of this study. Hitherto, they could also be viewed as strength since they have facilitated the process in making the research so focused and practical to propose such a substantiated findings with regard to the goal set in the research.

6. Review of Related Literature and Best Industry Practices

Regression testing is a process of testing changes to computer programs to make sure if the previous programming still works well with the new changes (Bertolino, 2003). In sustenance to this perception, (Rothermel, 1996) stresses such testing prerequisites some factors as alteration in the code, occurrence of a new build, incidence of detected defect, existence of maintenance work, and the likes. This type of testing is also referred to as program revalidation (White, 1989). She further noted that it is carried out to confirm if no new error have been introduced into previously validated code.

In support of (Bertolino, 2003), (L.White, 1997), (T. Graves, 2001), (R. Gorthi, 2008), and (Xest, 2010) deliberate on regression testing as a consequence of certain visible changes that might have occurred in any one of the artifacts that constitute or establish the functionality of the application. It is initiated when programmer fix a given bug or add a new code for new functionality to the system. There can be one or more dependencies in freshly added code on the existing functionality. It is a quality measure to check that new-fangled code complies with old code and the unmodified code is not getting affected (DeMillo, 1991).

This type of regression testing would have two goals to realize. First and for most, it will check up on the system to make sure if the fix is done the right way, in the right manner, and achieve the desired outcome. Secondly, it stabs to find out if the fix has caused or resulted in any damage to the enduring parts of the system (Broekman, Testing Embedded Software., 2002). The above notion a key point that gained due support by some present industry experts. (Udin, 2014), (Samuel R., 2014), and (Jane, 2014) are worth mentioning in this regard. However, they don't have sound backup that exhibits if their perspective is the only true cause for regression testing to get the feasible visibility.

This researcher, conversely, considers that the above genus of regression testing doesn't encompass and/or represent all genera of regression testing. It is, instead, a partial illustration of the intact regression testing. In other

1 Personal observation of the researcher from the informal discussions held with the informants of this research and other experts involved in the field.

2 84% of the informants of this research affirm to this point.

3 90% of the informant of this research affirm to this point.

4 94% of the informants of this research support the idea of crafting a clear definition for regression testing and the details under it.

5 80% of the informants of this research support the assumption in this hypothesis.

words, regression for fix is just one category of regression testing. According to Verma, this category is known as regression for fix (Verma, 2014).

Without the inclusion of regression for risk which could occur periodically to check on the system and see if it still meets requirements, one can't get the full picture of regression testing (Samuel S., 2014). Regression for risk occurs on a smoothly running system with the desire to find out if the system works right and up to the expected standard. This testing type, is not dictated by prerequisites. Instead, it is practiced without any precondition in place (Bennet, 2014). Nevertheless, it has, everything to do with making sure if the system is still healthy enough without even having any prior symptom of defect detection (Huizinga, 2007).

Even if very little is known in a scientific manner about this regression testing, it has been, routinely, referred as regression for risk. As indicated above, maintenance of a software product is often imposed to fix defects, to introduce novel functionalities, improve the existing system, reengineer, and/or acclimate the current functionalities, or to port it to different environment (Kolawa & Huizinga, 2007). Whenever a system is modified regression for fix is executed to check if the alteration works accurately in harmony with the rest of the codes in the system (Rudd, 2014). In fact, after a defect is detected and got fixed, regression for fix is run to make sure the fix is properly done, addressing the issue and without affecting the existing functionality of the application under test (Samuel R., 2014).

This is once again a onetime activity which is all about regression for fix. Nevertheless, those same test cases built for the executing regression for fix might be periodically run against the system to make sure if the system is healthy and functioning well meeting expectations and set standards without any quality compromise. This test is what usually refereed as regression for risk (Samuel S., 2014).

Scholars like (E. Engström, January 2010), (Harrold, A Safe, Efficient Regression Test Set Selection Technique, 1997), (G. Rothermel and M. J. Harrold, 1996) have circuitously indicated that regression for risk is a quintessential part of regression testing. (Antonie, 2014), (Rudd, 2014), (Hanip, 2014), (Kinf, 2014), and (Samuel S., 2014) tend to compile to the above insight.

To sum up, both regression for fix and regression for risk are essential parts of the regression testing process for ensuring software quality (Hanip, 2014). Both denote process of verifying and validating software either case by case after a fix has occurred, and/or periodically without any preexisting situations⁶. In both instance, they are done to provide confidence that the behavior of the software remains intact and has not encountered any major unexpected variation (Bennet, 2014; Samuel S., 2014; Verma, 2014). This testing never pre-necessitate the existence of change, defect, or bug for the test to take place. It runs intermittently to make sure everything runs as expected (Rudd, 2014).

7. Methodology

This article, as a part of an ongoing PhD dissertation in computer science and software testing in an Inter-University Program at Universidad Azteca and Universidad Central de Nicaragua (UCN) in Mexico City – Managua Nicaragua, has profoundly depends on data collected from experts who have been engaged in the software testing industry for years ranging 3 to 40 at various capacities and skill sets here in the USA. 60% of these experts are females, which might signpost the fact that females are more involved in the testing industry compared to males.

Of them, 90% have experience less than 10 years, while over 90% of the male expertize have over 10 years hands-on experience. This might indicate that the females might have started⁷ joining the field in the last few years while the males have stayed longer and are now withdrawing from it. What's more, only 30% of the female informants have first degree or above relevant to the field while 85% of the male informants have first or second degree which has direct relevance to the field.

I have classified the informants into three groups: GA, GB, and GC. The GA (Group A) informants constitute only 4% of the total number of the entire informants involved in the research. The job titles of these GA people is Computer Scientist, its equivalent or better. They have a minimum of 30 years of experience in the field. They have taken quite a lot trainings and possess a minimum of second degree with direct relevance with the field. As a matter of fact, all members in this group are males.

GB (Group B) establish 10% of the informants. Of them 2% are females while the remaining 8% are males. Members in this group have stayed 10 to 30 years in the field. All of them have at least a second degree with academic relevance to the software testing industry. Their respective job title is either manger /test lead or its equivalent, each having 5 or more testers working under them.

GC (Group C), on the other hand, holds people who have three years and above hands-on experience and expertise in the testing industry. 58% of them are females. 8% of these group are automation engineers whose primary responsibility is automating test cases, and building frameworks. 60% of the automation engineers are females. 92% of this group is instituted by manual testers.

A questionnaire has been dispatched to all of the aforementioned informants. The feedback has been gathered in due time, analyzed and synthesized using SPSS. Peer discussion and in person interview have also been conducted with them. The discussion points were driven from the findings acquired as outputs of the analysis carried out on the answers to the questionnaire. Then, a brief discussion has been conducted with GB and GC members to re-define regression testing and its common details. Likewise, a kind of in person interview has been conducted with GC members. This interview questions were formed out of the consensus

⁶ Note taken from the discussion held with the informants of this research.

⁷ Suggestions captured from an interview with two of the high level experts in the field software testing.

reached in the discussion conducted with GA and GB members.

The questionnaire was dispatched before and after a peer discussion. For the sake of convenience the informants were classified into four groups G1, G2, G3, and G4. Each group selects its own chairperson who leads and handles the discussion. The discussion was conducted until everybody clears up his/her questions about the subject matter. Consensus is not a must. Everyone could go out of the meeting without reaching in one accord with the rest of the peer groups.

The points of discussion concentrates on the basic concepts of regression testing and its details. These points were prepared by the researcher. However, the discussion was made based on the everyday practice of each peer group in their respective companies. The questionnaire was dispatched before and after this discussion to analyses and synthesize its impacts.

This data gathered from these questionnaire has been double assessed and analyzed from various angles using SPSS. The finding has been supported by available and relevant literature in the field. The process was a little tedious.

8. Discussion and Finding

This part brings forth the findings of the research. The discussion has two major parts. The first part is general assessment which tries to discuss what the informants know about regression testing and its details before and after the peer discussion. This helps in drawing a clear line on the depth of the awareness the stakeholders has about regression testing and the details under it. The second part focus on redefining regression testing as a whole and its details in particular, based on the findings in the research.

9. General Assessment

As indicated above, the general assessment was conducted to find out if the informants have clear knowledge of the regression testing they practice. This assessments was accompanied through a questionnaire designed for this purpose. This questionnaire was disseminated a couple of times. First without conducting a peer discussion on some of the important concepts of regression testing. Latter, that same questionnaire was dispatched among that same group to see if they have a change of mind after the peer discussion.

10. Pre-peer Discussion

Answering to the first question which asks if the informants do regression testing, all of them said they do. However, when asked the frequency and the timing of their testing endeavor, 84% said that they do such a testing whenever there is a change in the system, while the

remaining 16% said, they do sporadically without the presence of any prerequisite, in addition to what they usually do whenever a certain change occurs in the application caused by developers during new builds and maintenance work.

While asked if they clearly know the type of regression testing they do, 74% said they are not sure about it. They added, the only thing they know is that they run existing test cases or new test cases against the application at any time where there is a need to do so. The rest of them (26%), however, know what type of regression testing they do, be it regression for risk or regression for fix. They further indicated that they do the former one periodically while the latter one happens after a certain change has been introduced into the application under test.

While asked if the regression testing which they do requires a precondition, 64% of the informants think that precondition is a necessity, and 26% said precondition is not a compulsion, and 10% said they are not sure about it. Expelling the type of changes, they further said, the changes might include but not limited to new build, detected defect, system maintenance, system update, system reengineering, system upgrade, and the likes. The finding evidently demonstrations that all of the members of GA group with the exception of 4% of them, flagged for this notion. None of GB and GC members have put their decrees towards this notion.

As mentioned above, all of GA and GB and 12% of GC members, whose percentile sums up to 26%, supposed that prerequisites are not indispensable for conducting a regression testing, and yet, they still believe that certain regression testing are caused by some sort of preconditions as mentioned above. They totally respect and fully value the opinion raised by the 64% of the informants. According to them, the prerequisites are imperative to certain types of regression testing but not imperative to all sorts and conditions.

11. Peer Discussion

The purpose of this peer discussion is to clear out and clean up some ambiguity among informants in connection to regression testing from theory can concept perspective. It was a kind of brainstorming discussion focusing on the basics of regression testing and its details. This discussion was led and managed by the informants with a certain degree of the involvement of the researcher.

The discussion has three questions in it. The first question asks about what a regression test is all about. Likewise, the second and third question asks identical idea about regression for fix and regression for risk, respectively. The peer groups had to discuss and explain these questions based on their respective experience and expertise. The discussion was open to questions and answers. As a matter of fact, it was not a must to reach to a consensus.

The discussion was conducted after the first round questionnaire was posted and feedback was collected. After

this discussion that same questionnaire was dispatched among same informants. The feedback has a whole a lot of difference paralleled with the previous one. The following sub topic discusses the difference.

12. Post-per Discussion

The answer to the first question which asks if they do regression testing at all has not changed. Altogether, unanimously, agreed that they do. However, with regard to the setting (the when and where) question there was a significant difference compared to the pre-peer discussion result. 94% said that they do regression testing whenever a change occurs to the system under test. They also confirmed that they do regression testing periodically to make sure if the applications runs well meeting expectations. It was only 6% of them who were not still sure about it.

With regard to the second question as well, there was a significant amount of difference observed in the feedback. 90% of them said that some regression testing needs preconditions as new build, change in the code and the likes, while some could be done without these changes periodically to check up on the system health. However, 8% of them still insist that a precondition is a necessity for regression testing to come to life while 2% of them supports neither side.

When asked if they know what type of regression testing they do, 96% said they do know while the rest 4% they are not sure about it. Those who said they know, have plainly sign posted what kind of regression testing they run intermittently, and on a regular basis as well. The finding evidently indicates that the peer discussion has enhanced the understanding of the informants about what they do and what they know with regard to regression testing and the two details under it.

13. Final Discussion and Interview

Eventually, the researcher has conducted a higher level discussion with GB and GC members and finally concluded with a brief interview with the GC members. The purpose of this discussion and interview was to redefine regression testing and its two common details, in fact, based on the finding in the research.

14. Consensus of the Current Industry

As the industry practice designate there are two kinds of regression testing: regression for fix and regression for risk (Varma, 2014), (Bennet, 2014), (Hanip, 2014), (Samuel R. , 2014), (Kinfe, 2014), and (Samuel S. , 2014). All these expertize tend to agree that regression for fix is usually conducted after a certain change has been introduced to the system under test. This change could have multiple causes. The most common ones however, might include, the presence of defect in the system, the introduction of new changes into the system, new build, system reengineering

and update, maintenance, and the likes.

On the other hand, regression for risk needs none of these preconditions, and/or they are not a compulsory factor to it. It gets carried out periodically to evaluate the status of the application and to checkup if it is still working as expected (Varma, 2014), (Bennet, 2014), (Hanip, 2014), (Samuel R., 2014), (Kinfe, 2014), and. It is usually done to proactively find out if the system is well and functioning as expected. It could also be run to discover any foreseeable defect/s and address them in due time before they cause severe impairment on the system⁸.

15. Re-defining Regression for Fix

Eventually, discussion has been held with GB and GC members in the presence of the researcher as a moderator. The purpose of this discussion was to redefine the key words in this article: regression testing, regression for fix and regression for risk. The first term which came up for the discussion was regression for fix. The house suggested that, the intent in this term is to denote a regression test executed against a system to make sure if the fix is properly done and has no consequential impact on the entire application.

This definition is well supported by contemporary scholars in the field. Rothermel said that regression for fix is usually carried out as a part of software life cycle development to make sure if the fix is, first of all, well done or has addressed the issue, and secondly to check if it has not incurred any consequential damage in the rest of the functionality of the application (Rothermel, 1996).

Maintenance of a software product is frequently necessitated to fix defects, to add, enhance or adapt existing functionalities, or to port it to different environments (Kolawa & Huizinga, 2007). Whenever an application program is modified for carrying out any maintenance activity, resolution test cases are de- signed and executed to check that the modified parts of the code work properly (Rudd, 2014).

Regression for fix, therefore, is initiated when developers fix any bug/s or add new code/s for new functionality to the system. There can be many dependencies in newly added and existing functionality. It is a quality measure to check that the new code complies with old code and unmodified code is not getting affected (DeMillo, 1991).

16. Re-defining Regression for Risk

Not very much is written or said in a scientific manner about this type of regression testing⁹. Nevertheless, habitually it has been referred as regression for risk, since it is associate with risk not fix. Regression for risk transpires without any precondition on a system which is not under

⁸ Interview note with Samuel Sanchez (Computer Sciences, May 03, 2014)

⁹ Interview note with Raj Varma (Senior Manger Quality Assurnace, April 27, 2014)

construction or maintenance. On the contrary, it is sporadically carried out on a running system to find out if still it meets requirements (Hanip, 2014).

In spite of this, the discussion denotes that regression for risk, is a recurrent functional test that occurs periodically to check on the application if still it meets requirements without any compromise. Such testing never requires the presence of any prerequisite such as detection of defects. Some companies run this type of testing annually, monthly or quarterly while others do it on weekly or by-weekly basis¹⁰. The timing depends on the needs and interests of the individual companies. However, for the testing to be leveled as regression for risk, it has to run without any precondition periodically within a range of a given time to make sure the healthiness of the system (Verma, 2014).

17. Re-defining Regression Testing as a Whole

The objective of regression testing is to identify unexpected defects, which remained in the system, due to several reason (Rudd, 2014). While changing the code, the developer possibly could not completely understand the internal correlations of the code (Verma, 2014). There could also be some other reasons for the bug to reside in the code. Be that as it may, the existence of these bugs within the structure of the code could put quality under a big question mark (Katrack, 2012). In this regard, regression testing becomes an essential activity to make sure if the system behaves right (Kinfe, 2014).

What's more, regression testing could also be accomplished periodically to check on the health of the system, to find out if it still works well meeting all standards and requirements without compromising quality. This type of regression testing is too repetitive, time consuming, labor intensive, expensive, and exorbitant (Harman, 2008). Nevertheless it is very imperative to guarantee the quality of the end product (Samuel S., 2014).

Hence, regression testing remains the only reliable means to provide adequate confidence that changes in the code are safe and are not liable to "break" the existing functionality of the application (Hanip, 2014). Having this in mind, it was suggested that regression testing, as a whole, is a combination of, at least, regression for fix and regression for risk¹¹. It is, therefore, a repeated sort of functional testing intended to ensure quality through regression for fix and maintain same via regression for risk. Nevertheless, it is way different from retesting, dry running and debugging a script.

This notion is well supported by the contemporary literatures in the field. To mention some by few, (Harman, 2008), (Mall, 2010), (Lin, 2010), (Rothermel, 1996), and (W.E. Wong, 1997) think that regression testing is an activity that is performed to provide confidence that changes

and/or settings do not harm the existing behavior of the software. The word 'change' might signify the importance of regression for fix which needs a prerequisite to take place. The other word 'setting', on the other hand, is much about time and environment. As days go by and the setting changes, the application might fail meeting standards and expectations. Henceforth, periodical testing might be needed to check on its health.

Regression testing, thus, is an umbrella testing class that hosts, at least, both regression for fix and regression for risk¹².

18. Conclusions

As seen in the discussion, regression testing seems to be one of the very least researched topics despite its great wide implementation and importance in the industry (Lin, 2010). This research designates that very little is known about it, even among the stakeholders. The variation of the feedbacks collected from the informants in this research before and after the peer discussion explains this awareness gap very well. This, in turn, highlights the need for more detailed and intensive scientific research on regression testing. In the meantime, the suggested definitions on regression testing and the two details under it, might help building consensus about the subject area in the mainstream.

REFERENCES

- [1] Antonie. (2014, May). Manual testing. (D. Asfaw, Interviewer).
- [2] Bennet, L. (2014, May 27). Manual testers team lead. (D. Asfaw, Interviewer).
- [3] Bertolino, A. (2003). *Software Testing Research: Achievements, Challenges, Dreams*. Pisa, Italy: Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" Consiglio Nazionale delle Ricerche.
- [4] Brenda. (2014, May). Manual testing. (D. ASfaw, Interviewer).
- [5] Broekman, B. a. (2002). *Testing Embedded Software*. Addison-Wesley.
- [6] Buzzle. (2014, May 27). Retrieved from Different Types of Application Software : <http://www.buzzle.com/articles/different-types-of-application-software.html>.
- [7] David Sims, C. S. (2013, October 4). Retrieved from An independent evaluation of the work experience placement trials between May 2012 and July 2013.: <https://www.gov.uk/government/publications/evaluation-of-the-work-experience-placement-trials>.
- [8] DeMillo, R. A. (1991). *Software engineering*. Proceedings of the 13th international conference on Software engineering. 13th international conference on Software engineering.

¹⁰ Personal experience and observations taken from informal discussions with some of the informants of this research and other experts in the field.

¹¹ Note taken from discussion with GB and GC members.

¹² Captured from the consensus reached by the informants of this research.

- [9] E. Engström, P. R. (January 2010). A systematic review on regression test selection techniques. *Information and Software Technology*.
- [10] G. Rothmel and M. J. Harrold. (1996). *Analyzing Regression Test Selection Techniques*,. IEEE Transactions on Software Engineering, V.22,.
- [11] Galde, J. (2014, May 7). The co-evolution of mobile platforms and quality assurance . *SD Times*.
- [12] Google Dictionary. (2014, May). Retrieved from <http://dictionary.reference.com/browse/google>
- [13] Hanip, A. (2014, 05). Complexity of software testing. (D. Asfaw, Interviewer)
- [14] Harrold, G. R. (1966). *Analyzing Regression Test Selection Techniques*. New York.
- [15] Harrold, G. R. (1997). A Safe, Efficient Regression Test Set Selection Technique,. *ACM Transactions on Software Engineering and Methodology*, V.6,.
- [16] Hetzel, W. C. (1988). *The Complete Guide to Software Testing*, 2nd ed. New york: Wellesley.
- [17] Howden, W. E. (1987). *Functional program Testing and Analysis*. McGraw-Hill.
- [18] Huizinga, D. (2007). *Automated Defect Prevention: Best Practices in Software Management*. Wiley-IEEE Computer Society Press.
- [19] Importance of regression testing . (2014, May). Retrieved from Importance of regression testing: <http://www.docstoc.com/docs/34729120/The-Importance-of-Regression-Testing>.
- [20] Jane, M. (2014, May 2). Could testing be exhaustive? (D. Asfaw, Interviewer)
- [21] Katrick. (2012, May). QTP online. Vedio . QTPILearn.
- [22] Kinfe, S. (2014, May 7). Automation Engineer . (D. Asfaw, Interviewer)
- [23] Kolawa, A., & Huizinga, D. (2007). *Automated Defect Prevention: Best Practices in Software Management*. Wiley-IEEE Computer Society Press.
- [24] L.White, K. A. (1997). A firewall approach for the regression testing of object-oriented software. In *Proceedings of 10th Annual Software Quality Week*, (p. 27).
- [25] Lin, X. (2010). *Regression Testing in Research and Practice*. Lincoln: Computer Science and Engineering Department University of Nebraska.
- [26] Mall, S. B. (2010). *Regression Test Selection Techniques: A Survey*. Kharagpur, India: Dept. of Computer Science and Engineering IIT.
- [27] Maneging complexity. (2014, May). Retrieved from Maneging complexity :<http://www.economist.com/node/3423238>
- [28] Musa., J. D. (1997). *Understanding software testing*. Proceedings of the 1997 international conference on Software engineering.
- [29] Myers, Glenford J. (1979). *The art of software testing*. In G. J. Myers, *The art of software testing*. New York: Wiley. Retrieved from 1979. , Publication info: New York : Wiley, c1979. ISBN: 0471043281.
- [30] Nair, R. P. (2008 , June). Types of testing. Retrieved from <http://rajeevprabhakaran.wordpress.com>.
- [31] Nizan. (2014, May). Computer software. (D. Asfaw, Interviewer).
- [32] PC Magazine Encyclopedia. (2014). Retrieved from Definition of: test scenario: <http://www.pcmag.com/encyclopedia/term/52773/test-scenario>.
- [33] Peleska, J. (2008). *A Formal Introduction to Model-Based Testing Part I: Exhaustive Testing Methods*. Verified Systems International GmbH and University of Bremen.
- [34] Perlis, A. J. (2009, 10 15). The Origin of Software Bugs. Retrieved from The Origin of Software Bugs: <http://www.informit.com/articles/article.aspx?p=1398775>.
- [35] Quality Digest. (2014, May). Retrieved from Definition of quality: <http://www.qualitydigest.com/magazine/2001/nov/article/definition-quality.html>.
- [36] R. Gorthi, A. P. (2008). Specification-based approach to select regression test suite to validate changed software. In *Proceedings of the 2008 15th Asia-Pacific Software Engineering Conference*, (pp. 153–160).
- [37] Rahman, M. (2014, June 2). PhD Questionnaire . (D. Asfaw, Interviewer).
- [38] Reiner Musier, P. (2013). *Trends in Automated Testing For Enterprise Systems* . WorkSoft.
- [39] Roper, N. P. (1989). *Understanding Software Testing*. New York: John Wiley & Sons.
- [40] Rothmel, G. a. (1996). *Analyzing regression test selection techniques*.
- [41] Samuel, R. (2014, July 20). Automation Engineer. (D. Asfaw, Interviewer).
- [42] Samuel, S. (2014, May 03). Computer Scientist. (D. Asfaw, Interviewer).
- [43] Shin Yoo and Mark Harman. 2008. Regression Testing Minimization, S. a. (2008). *Regression Testing Minimization, Selection and Prioritization : A Survey* King's College London . Strand, London: Centre for Research on Evolution, Search & Testing.
- [44] Software Testing. (2014, May 19). Retrieved from Software Testing: <http://www.docstoc.com/docs/95018738/Software-Testing>.
- [45] T. Graves, M. H. (2001). A. Porter, and G. Rothmel. An empirical study of regression test selection techniques. *ACM Transactions on Software Engineering and Methodology*.
- [46] Test Condition. (2014, May 3). Retrieved from <http://www.allinterview.com/showanswers/67697.html>.
- [47] Testing Articles. (2014, May). Retrieved from What is Automated Software Testing: <http://www.testingperformance.org/definitions/what-is-automated-software-testing>.
- [48] Total number of websites. (2014, June 1). Retrieved from <http://www.internetlivestats.com/total-number-of-websites/>.
- [49] UCS Libraries . (2014, May 17). Retrieved from Organizing Your Social Sciences Research Paper: <http://libguides.usc.edu>

- u/content.php?pid=83009&sid=616083.
- [50] Udin, M. (2014, May). Software testing. (D. Asfaw, Interviewer).
- [51] UK Essays. (2014, May 9). Retrieved from Customer satisfaction and quality assurance: <http://www.ukessays.com/essays/business/customer-satisfaction-and-quality-assurance.php>.
- [52] V. N. Mauraya, a. R. (2009). Analytical Study on Manual vs. Automated Testing Using with Simplistic Cost Model . India : Aligarh UP.
- [53] V.Kharytonov. (2012, December 15). Software Measurement: Its Estimation and Metrics Used. Retrieved from <http://it-cisq.org/software-measurement-estimation-metrics/>.
- [54] Verma, R. (2014, April 27). Manger, Quality Assurnace at Global ISI. (D. Asfaw, Interviewer).
- [55] W.E. Wong, J. H. (1997). A Study of Effective Regression Testing in Practice. Proc. Eighth Int'l Symposium Software Reliability.
- [56] Web Definition. (2014, May). Retrieved from <http://www.google.com/search?client=safari&rls=en&q=test+scenario+definition&ie=UTF-8&oe=UTF-8#q=manual+testing+definition&rls=en>
- [57] What is a Test Case. (2006, February 27). Retrieved from What is a Test Case?: <http://testingsoftware.blogspot.com/2006/02/test-case.html>.
- [58] What is automated software testing. (2014). Retrieved from <http://idtus.com/what-is-automated-software-testing/>.
- [59] What is scope and limitation of the study. (2013, May 12). Retrieved from <http://www.reference.com/motif/Education/what-is-scope-and-limitations-of-research>.
- [60] White, H. L. (1989). Insights into regression testing. In proceedings of the conference of software maintenance, (pp. 60-69).
- [61] Wikipedia. (2014, May 07). Retrieved from Internet-Speed Development: http://en.wikipedia.org/wiki/Internet-Speed_Development.
- [62] Wikipedia. (2014, May). Retrieved from To err is human: http://en.wikipedia.org/wiki/To_Err_is_Human.
- [63] Wikipedia. (2014, May). Retrieved from Software Testing: http://en.wikipedia.org/wiki/Software_testing#Testing_methods.
- [64] Xest, M. N. (2010). An automated framework for regression testing of embedded soft- ware. In Proceedings of the 2010 Workshop on Em- bedded Systems Education, WESE , (p. 8).
- [65] Zafar, R. (2014, May 6). What is software testing? What are the different types of testing? Retrieved from What is software testing? What are the different types of testing?: <http://www.codeproject.com/Tips>.
- [66] Zhang. (2014, May). Computer software. (D. Asfaw, Interviewer).
- [67] Ziegler, J. G. (1989, May). Ziegler, J., Grasso, J. M., and Burgermeister, L. G. 1989. An Ada based real-time closed-loop integration and regression test tool. Proceedings of the Conference on Software Maintenance. An Ada based real-time closed-loop integration and regression test tool.