Multi Domain Supported and Technology Neutral Performance Testing Process Framework

U.N. Rajapakse
Advanced Technology Center,
Zone24x7 Private Limited,
Colombo, Sri Lanka

J.M.S.J. Jayasooriya
Advanced Technology Center,
Zone24x7 Private Limited,
Colombo, Sri Lanka

A.S. Siriwardene
Advanced Technology Center,
Zone24x7 Private Limited,
Colombo, Sri Lanka

C.D. Manawadu
Advanced Technology Center,
Zone24x7 Private Limited,
Colombo, Sri Lanka

ABSTRACT

There is a growing necessity in assessing and streamlining enterprise applications against the desired performance characteristics from the end users' perspective. There are significant numbers of commercial tools and performance measuring algorithms designed for mentioned purpose. However, from the process framework perspective, the contribution is limited in the software industry. This paper discusses an overview to performance testing process framework that employs at Zone24x7 in various business dynamics such as Banking, Mobile and Retail industries. An in-depth analysis was carried out within associated clientele to construct the proposed framework with the aim of identifying value preposition that it derives. In addition, deciding performance objectives, designing appropriate strategies, execution and analyzing test results will be discussed along with the key benefits to the clients who are keen on conducting performance testing to achieve immediate operational excellence hence the sustainability dimension in long run. In conclusion, the paper will further discuss about the potential domains that could adopt this proposed performance testing process framework to gain before mentioned benefits.

General Terms

Process framework

Keywords

Software industry, performance testing, performance testing process framework

1. INTRODUCTION

Meier et.al [1] defined performance testing as "*the technical investigation done to determine or validate the speed, scalability, and/or stability characteristics of the product under test".* These activities include performance testing plus further tuning of the software application. It is an intention to achieve the benchmarked response times, throughput and resource-utilization levels which was stated in performance objectives. Testarconsulting [2] stated, *"in software engineering, performance testing is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage".* Therefore, performance is a decisive measurement that indicates how good a software application meets its non-functional requirements and fit for the intended purpose. When considering a real business scenario that handles one million transaction volumes in daily basis, if the home page of the web application may take extra 2 milliseconds to load, what will be total cost by end of the day? It may be the reputation, losing market share, challenging brand loyalty or what else? The impact might be list longed and might not be recovered in company's lifetime! In order to eliminate such an unpleasant experience, organizations need to think through about their performance testing strategy in order to go ahead with closer rivalry to gain the competitive advantage in modern business dynamics.

However, there are various reasons why performance issues erupt in the industry and it is vital to assess the reason for such concerns. Xiang [3] explains that reasons leading to performance failures can be basically categorized as *"the lack of performance estimates, the failure to have proposed plans for data collection and the lack of a performance budget"*. Hansjorg [4] described, *"New technologies are increasingly abstract and complex, which leads to an almost infinite number of combinations. This increases the risk of hidden performance problems".* In addition, performance testing facilitates to find the faults which were not discovered in manual testing at the different level of testing; component integration, system or UAT (User Acceptance Testing) testing. This leads to detect bottle necks in systems [4]. Therefore, performance testing becomes *an extra-value addition* for organizations to position their software solutions by superseding the closer rivalry as well as to avoid unwanted repercussions in latter stages of development.

There are significant numbers of commercial tools and performance measuring algorithms that were designed for this mentioned purpose. However, from the process framework perspective, the contribution is limited in the software industry and the researchers have conducted a comprehensive literature review to comprehend this inadequacy.

There are many articles written on performance testing and most of them are based on a specific tool or a technology which uses to solve a performance problem within the software applications. In parallel, some articles describe the algorithms which propose to measure the performance of the applications in different perspectives. There are many different ways to assess the performance testing enterprise applications and some of them are more complex than others. As a result, each tool has specific disadvantages over the advantages.

Within the performance testing domain, there were many researchers conducted by experimenting the various tools in different business aspects. In 2010 Oracle [5] came up with an approach in performance testing and they have discussed about testing the upper limit of the system for a high end user load. Based on the type of performance testing the engineer intends do, will depend on what type of results s/he wants to achieve. As per [5], for repeatability testing, benchmark testing is one methodology. However, to test the upper limits of the application from the perspective of concurrent user load, capacity planning tests should be used.

One of the other important points that Oracle discussed in this article is explaining about the suitability in applying load for a J2EE application. In comparison with proposed process framework, key important facts have been missed in this methodology for applying load for a J2EE application (in this method they specifically stick in to the load test and how to generate that load using the user count). Hence, this methodology can only apply for the load level testing. Further, comparing to the process framework we present, it does not apply to performance load testing. The process framework can apply to different types of performance testing and it is technology independent. In peak-rest testing, the article discuses about how it can generate the load repeatedly, how to bring that load to the peak level and test the application at the peak level. Though the article considered on the load generating, key facts like identifying the environment, planning the test and make proper estimation, and finally the evaluation of the results in order to come to a conclusion sections are missing.

In 2013, Jamo Solution [6] came up with a method of measuring performance of mobile application and the process they defined was only applicable for mobile applications. In their paper, Jamo Solution [6] vastly discussed about User Interface based performance testing, simulating network conditions, monitoring the performance and load generating. Thus, Jamo Solution [6] is device biased and in addition the paper was not paying enough attention on initial performance test planning. Planning stage becomes critical and that is the phase, the engineers to make proper estimations on on-going assessments. Further, Jamo Solution [6] has not discussed about the setting up environment and execution environment is critical as it needs to set up according to the performance criteria. In contrast, in the proposed process framework, clearly discusses about the above phases in a sequential process order and finally sharing of results in client preferable formats for convincing clients.

In 2011, research team in Queens University [7], came up with another solution is called, *"Automated Verification of Load Tests Using Control Charts"* [8] and this solution mainly discusses about the load testing; how testing and results analyzing can be done using control charts. As before mentioned in Jamo Solution [10], the research team also fully focused on load testing and solution was not discussing about the other testing types. School of Computing Queen's University Kingston team submit another paper on *"A Methodology to Support Load Test Analysis"* [9] and the paper again, broadly discuss about the load testing approaches and results analysis after run the test.

According to Stuart [10], the pitfalls built-in open-source tools includes that they have a lengthy learning curve and less in support for quick implementation. In the other hand, as per him, *"the vendor tools support far more protocols than the open-source tools, but as long as you are staying in the web space, and you are looking at HTTP/S, IMAP and POP3, the open-source tools are pretty good".* However, when considering the commercial products, they usually offer some form of trial version that can be freely downloaded, but they typically come with strict limitations on the number of virtual users that can be created (often less than 10) [11].

Furthermore, Software Testing Software blog [12] states that the disadvantages of using commercial software testing tools include expensiveness, need to pay for additional plug-ins and patches, and needed to upgrade when newer version had been released to the market. In oppose, open source testing tools have steep learning curve, need programming skills to script, not supporting platform neutrality and no dedicated support team for fixing the possible issues.

Agile performance testing life-cycle proposed in HP forum in 2013 [13], the performance testing life cycle was sub divided into 5 phases; planning, pre-drop, regression, update and new baseline. The activities in planning phase are parallel and it should be executed while planning the next drop. The overall process architecture is ignored the criticality inter-mediatory estimations, in designing the performance goals of the application in agile environment. The performance engineers were not given proper solid sequential flow to follow and reporting mechanism was totally ignored in here. The confidence of adopting the total architecture in agile set-up is questionable and the assessing the associated risks and preventive mechanisms are uncertain.

By considering above factors we can conclude, there is no 100% guarantee that perfect process framework or tool which will be suited for measuring every kind of performance problems in application. Each tool has its' advantages as well as disadvantages and complete process framework is not available as of date. That is the main motivation behind why we are purposing this process framework which can be used in multi domains and it is technology independent for solving any performance issue.

Based on the systematic literature review, the prevailing process question driving this study is, how can multi domain supported and technology neutral performance testing process to be used in application performance testing process? Therefore, the study sets forth to examine the following two research questions,

1. How does theory inform courses of action and preferred approaches to applying multi domain supported and technology neutral performance testing process? And
2. The applicability of such process framework in real business cases

2. CASE STUDY

Rowlands [14] discusses that there are several research strategies available to a researcher to conduct their studies and those are mainly experiment, survey, case study, ethnography, action research and grounded theory based. Furthermore, Yin [15] used various terminologies to explain the multiplicity in the case study methodologies and sub divided them further as explanatory, exploratory, descriptive, multiple-case studies, intrinsic instrumental and collective. The authors thereby focuses mainly on the case study based research strategy and further drills down that approach to use multiple-case studies because of these after mentioned reasons.

Quantitative fact finding techniques such as interviews and surveys have not been considered for this mainly due to time constraints. In addition, authors believe that the information gathered through these techniques among the selected sample would generate somewhat bias information that will tarnish the research outcome.

The main reason to use these multiple-case studies is, *"A multiple case study enables the researcher to explore differences within and between cases. The goal is to replicate findings across cases. Because comparisons will be drawn, it is imperative that the cases are chosen carefully so that the researcher can predict similar results across cases, or predict contrasting results based on a theory"* [15]. Pattonm [16], stated that researches do not always get good theoretical backing when they explore new or cutting edge issues and in those instances it's appropriate to create logical model which is also known as *"theory of action"* to proceed.

Therefore, in this study, the researchers have proposed and deployed a framework for performance testing as a logical model to conduct the research considering the practical environment and how each business entity faces performance related issues. In order to proceed, authors have conducted multiple case studies by interviewing existing projects that covers retail industry, point of sale, banking and finance and product engineering. Data was gathered from 8 respondents from these areas to identify and take facts about each of their project cases related to software performance by conducting series of interviews based on a structured questionnaire. Each interview was properly time spanned within one (1) hour's time and this information was recorded and later created transcripts for each case for coding for clearly differentiating each of these cases where all these responses are stored as facts. Furthermore, these facts which are then stored in a data matrix by recording each respondent's response to each variable that was captured though transcripts. And later on, this information was used to generate descriptive statistics to confirm the hypothesis and also to arrive at conclusions.

3. PROPOSED FRAMEWORK

This section comprehensively illustrates, what is the generic approach that authors proposed for performance testing for varies business domains ranging from mobile solutions to retail businesses that operate in different scales; small and medium scale businesses to multi-nationals and Fig 1 depicts the performance testing process framework in a holistic view. The main phase data flow depicts in red arrow and sub-phase data flow depicts in yellow arrows.

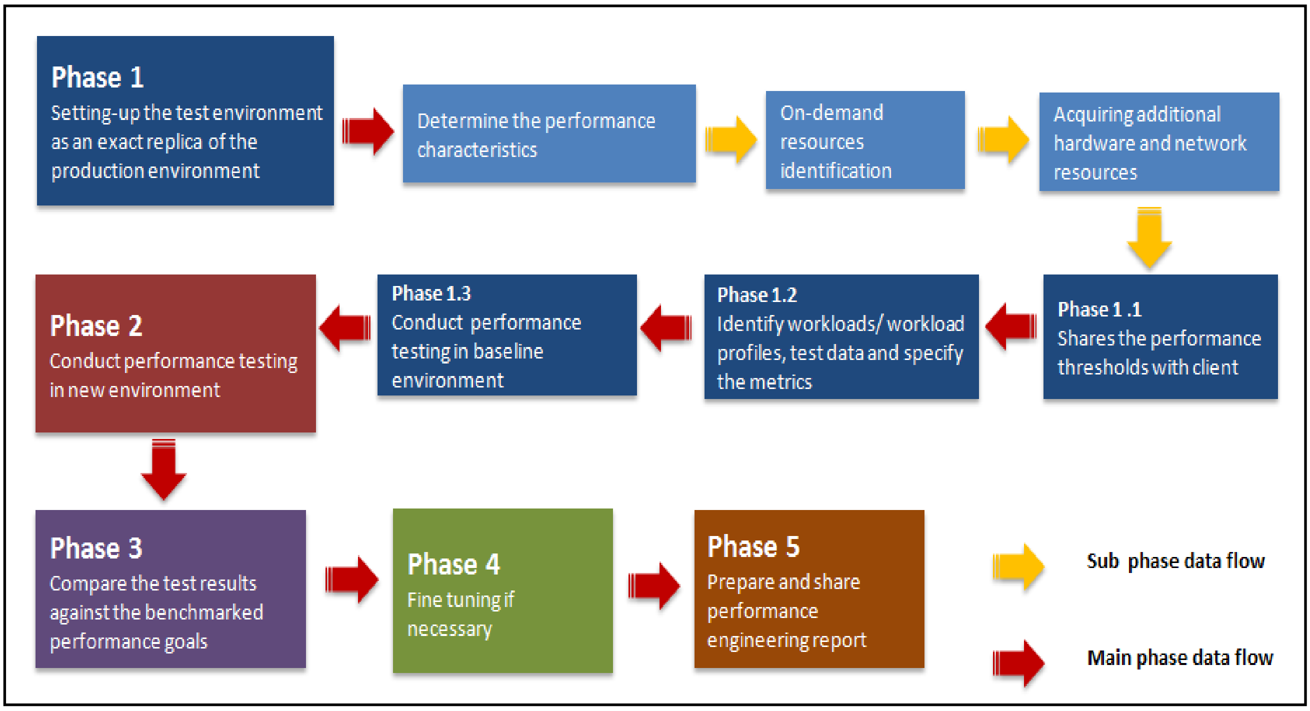
Phase 1
Setting-up the test environment as an exact replica of the production environment
Determine the performance characteristics
On-demand resources identification
Acquiring additional hardware and network resources
Phase 1 .1
Shares the performance thresholds with client
Phase 1.2
Identify workloads/ workload profiles, test data and specify the metrics
Phase 1.3
Conduct performance testing in baseline environment
Phase 2
Conduct performance testing in new environment
Phase 3
Compare the test results against the benchmarked performance goals
Phase 4
Fine tuning if necessary
Phase 5
Prepare and share performance engineering report
Sub phase data flow
Main phase data flow

Fig 1: Proposed Generic Performance Testing Process Framework

3.1 Phase 1

Web/ middle tier and database level performance testing in baseline environment

In order to set-up the performance environment, the performance testing strategy document should be formulated and finalized. This may be formed as a policy document which include *"what to be tested"*, *"by whom"*, *"how to test"* and *"when to test"*. In high level, the policy would include the performance testing objectives, the performance characteristics that is going to be measured, the business scenarios and the appropriate expected results, entry and exit criteria, incident reporting and fixing criteria, maintenance of testware and test environment for reusing in future etc… Further, business, technical and financial feasibilities should be taken into account for returning better results. Getting the consent from all stakeholders need to be accomplished in prior to start-off the performance testing phase.

First, the performance test environment needs to be set-up by identifying the necessary resources and in absence of the resources, appropriate steps need to be taken to acquire additional hardware and network necessities. In ideal condition, the goal was to determine the performance characteristics in the production environment, the test environment should be an exact replica of the production and that will be set-up after an agreement with the client. Thereby it enables the 100% confidence and assurance in accessing the production environment/s and our uniqueness and technical expertise is the predominant player in this business scenario.

In second sub-phase, the performance estimations are shared with the client for equating desired performance characteristics of the application and benchmark them. Classes of characteristics can be varied based on the types of the applications that needs to be measured and frequently correlate characteristics include,

[1] *Response time* - For example, no. of milliseconds taken to load the batch status page in banking application.

[2] *Throughput* - For example, 25 deposits per second must support by the banking application.

[3] *Resource utilization* - For example, processor utilization would not be more than 25 percent and memory utilization would be less than 20 reads/writes per second etc…

Next, sub-phase involves, identifying workloads and workload profiles by assessing and deriving key scenarios, using appropriate variability across users, indentifying test data and specifying the metrics to in-line with real-world simulations for enabling clients to make proper business decisions. Key performance scenarios derivation are based on the application typically surface during the process of identifying the desired performance characteristics of the application and associated metrics along with the indicators if applicable.

In fourth sub-phase, executing tests is base environment by closely monitoring the following sub-tasks.

- Communicate the test execution details with the team by validating test suite/s, tool/environment configurations, the settings in the base environment and associated test data.
- Start test execution: while the test is being in execution, we need to continuously monitor and validate scripts, systems and data.
- Upon the test completion, Engineers need to quickly review the test results for obvious variations and re-run the test suits if necessary by correcting the environmental or other influenced factors etc…
- Eventually, archiving the test suits, results and test data to repeat the test later if needed are mandatory.

3.2 Phase 2

Web/ middle tier and database level performance testing in new environment

The performance test engineer/s repeats the above steps in the change occurred environment (newly implemented changes are going to be tested here) with the identical associated IT infrastructure. In addition, before running the real test, it will execute a quick smoke test to make sure that the test script and remote performance counters are working correctly and environment will reset the system after successful completion of smoke testing to start a formal test execution.

3.3 Phase 3

Compare the test results against the benchmarked performance goals

The baseline and new test environments will compare against the performance characteristics, matrices and indicators by using standard analytical tools from basic MS-Excel application to more sophisticated analytical tools that use for data analytics.

3.4 Phase 4

Fine tuning

In most of the cases, test engineers are well-dominant in the performance test engineering process and baring the vast industry experience, are keen to take necessary steps in real-time without waiting till this phase and as a result rarely encounters performance re-run. However, if there is a need, test engineers can repeat the steps from very beginning in order to eliminate the application bottlenecks or undesired performance issues.

3.5 Phase 5

Prepare and share performance engineering report with clients for sign-off

The clients need more than just the results from various tests in a more detailed analysis along with side by side comparison data in possible graphical representations and using tabular formats. This proposed framework concerns more on this and provides a wide range performance engineering report formats based on client’s requirements. Further, provides the expertise and technical guidance to minimize or eliminate the performance gap if it is voidable in future in the form of a conclusion in the performance engineering report.

4. RESULTS

Performance testing is not an eccentric theme anymore and many ICT providers are investing their potentials to pioneer in that specialty. However, the hallmark of any organization positions firmly, simply because it’s proficiency in effectively managing the growing challenges of the eco-system of the organization. Added concerns with the advent of Mobile POS and diverse peripherals that are vital for any system to thrive in performance. BYOD (Bring Your Own Device) reduces the burden of security and intensive monitoring requirements thus promisingly leading to provide an enhanced end-to-end performance testing. Further, guaranteed reliability of the enterprise payment devices by securing the devices against unauthorized device tampering while considering the significance of having proper monitoring and management capabilities in place for the highly critical devices and associated hardware peripherals. Following is the business case studies, the co-authors had gathered to evaluate the proposed framework at Zone24x7.

During early 2000s, POS (Point of Sales) became one of the fasted growing areas in the field of IT solution [4]. Group of team in Canada had developed a POS system that caters the services for small and large scale of restaurant in Canada. They started with the small team and within a limited time the system became very popular. The POS consists with main server which has the POS core application and that application connected to the Oracle 11g database. There are client applications deployed in the client restaurant-end and it has a SQL (Structured Query Language) server connected to the client application. Whenever transaction happens in the client application, it will first add an entry to the local client side SQL database. In real-time, those data will sync-up with the core application server - Oracle 10g database. Also if there is a change made to the core server application, for example if they need to add an item to the restaurant menu with a new price, the only action that the POS operators have to do is to add that item to the core server application. Afterwards the data will be synced in real time with all the applications that were operating in the client restaurant.

With the increased number of clients, one of the major problems the POS Operators had to face was the performance and that forced them to drill down the bottleneck to identify route causes.

1. They noticed that data sync-up does not happen in real time between the server and the client
2. In order for real time synchronization they needed to compress the data packet from 20kbps to 2kbps

The task which was causing tremendous issues for the client was assigned to the off shore team in Sri Lanka and our Performance Engineers were able to compress the 20kbps data packet to 2kbs. In addition, without teams expertise knowledge they provided a database level optimization to manage real time data while effectively syncing-up with the client server and other operating servers in restaurants and most importantly eliminating the performance issue which was a burden to the client.

With the introduction of the internet in early 2000 web shopping became very popular in the United States and as a result, customers' attraction to web shopping retail stores tended to reap. More customers interacted with online retail website than the traditional brick and mortar stores. Since online retail web sites became popular, Web Developers had to add more features that meet such demand to attract more customers. To accommodate this sudden necessity, Web Developers used common APIs (Application Program Interface) to communicate data in which sometimes used by POS and websites as well. In parallel, to increase the purchasing volume in both stores and websites, Web Developers have designed more innovative solutions using various APIs which were already existing.

With the amount of concepts been added to both website and POS, the API was heavily been used for various services access and the usage increment of such services resulted in performance degrade and it affected adversely. Offshore Innovation Team had proactively identified this matter and they had suggested conducting a performance test for the backend services. As a result, Performance Engineers were able to identify the following performance bottlenecks.

1. Within a 10 second time stamp, POS application can only handle 102 users load requests and responses from the data access layer-API and the DB (Database)
2. Peek CPU (Central Processing Unit) usage for this load testing is 98%.
3. Also noticed that, one service is using 59 milliseconds during the peak time

These bottlenecks were solved by our Performance Tuning Team and as a result the CPU usage reduced up to 70% and service time accessing issue was reduced close to 40 milliseconds while enabling greater user experience in the system.

With the increase of the ICT (Information and Communication Technologies) enabled solutions replacing traditional banking solutions, remote depositing of cheque processing solutions stands outs as one of the most challenging. World leading solution providers in this business enables, customer to scan the cheque via mobile and upload to their depositing mobile application for processing. This method became very popular in the United States and one of the leading banks in United States has started to use this solution. With the increase of number of mobile uploads, performance of the application became very vital. There were various enhancements which were applied in the enhancement cycles and every cycle meeting the pre-agreed performance benchmark was critical. In the performance testing, offshore team had to meet the performance criteria defined by the client. For every enhancement, performance should not be dropped and there should be 10% performance improvement for every application enhancement as well.

Offshore team in Sri Lanka developed a product that is capable of managing servers, devices and monitoring network resources throughout an enterprise network. Our Product Engineering Team had a burning issue in registering higher number of devices. In the devices registration phase, product was only able to register a maximum of 200 devices even their initial target was to register more 1000 devices at a time. The Performance Engineering Team carefully verified the matter and they noticed that at that point the virtual memory usage 80 Mbs. Performance Team was able to identify the matter as a memory and thread handling issue and communicate to the Development Team for performance tuning. Development Team was able to solve the matter quickly and after fixing, the virtual memory usage reduced to 40 Mbs and this helped the Device Team to load close to 400 devices at a time. To provide a professional judgment, precise understanding on real business case/s or use cases, actors and the interfaces needs to identify in prior and experience, intuition and rational or unbiased judgments would be key for that success.

The web based legacy banking application which accepts financial transactions done by cheques via digital check images. The application consists of two databases, two sets of application servers and with a load balanced web server cluster. The users are able to scan checks through a check scanner and upload images, enter data associated with it, balance the amount and finally deposit. The web tier and middle tire of the current production system is developed on top of Microsoft .NET framework 2.0 and recently the management planed to migrate web application (in Web server) and Application services into Microsoft .NET framework 3.5 considering the new features available in the new framework.

The intended performance goals was to measure,

1. Measure the increase/decrease in performance after the migration from .NET framework 2.0 to .NET framework 3.5.
2. Quantifiably determine the effect on the system and side effects before and after migration.
3. Identify the areas in which where bottlenecks in the system

The designed performance metrics are as below.

- Response time for each web request
- Image processing time
- The throughput of the system
- Image upload time
- Maximum concurrent requests that can be handled

- Maximum number of users that can connect
- Total processor and Memory usage

In addition, system and workload parameters were designed as per the business case explicitly required by the client and workload is generated through a script developed by Microsoft Visual Studio Team Suite. The tool is capable of generating load collaboratively with more than one workstation and simulates concurrent user connections.

The final performance engineering report extracted the results as *"Analyzing the statistical data it can be observed that even the CPU time and response time is increased with the increase of user load for both platforms, however the .Net framework 3.5 shows better response time and hence more performance than the .Net framework 2.0. Furthermore, it can be noted that there is a slight increase in the memory usage in .Net framework 3.5 than 2.0. Considering all the statistics and parameters it can be concluded that system performance is around 7% higher in Microsoft .NET framework 3.5 than Microsoft .NET framework 2.0 but with extra amount of memory usage"* which was a proof-driven explanation to the client for making their decision on framework migration.

5. BENEFIT ANALYSIS

Following are additional service excellences that co-authors are experiencing by this proposed process framework.

Cost/Revenue Benefits – To harness the performance testing process for minimizing costs lead operational excellence and hence the key drive for achieving product excellence. This proposed process framework is capable of avoiding potential operational costs in 10% margin in average for experimented sample of four (4) clients and these costs were associated with early detection of future risk of performance failures. Thus, this would support as a risk minimizing function especially in cost recurring nature [1].

It is a norm that extra license are necessary for acquiring sophisticated software solutions and such fees are necessarily added to the initial proposal of the contracts. The proposed process framework, indispensably associated with one-time license fee platforms and sub-components and a lease system for one-off test campaigns which reduces hardware costs while supporting cross-platform capabilities that allows the load to be generated on a wide variety of platforms and Internet Browsers.

Professional Guidance and Care – Businesses are heavily relying on IT applications that needs to be executed in every second while catering to unusual high demands such as promotional or seasonal trading which can be a regular occurrence, making it crucial that these software applications are continuously prepared for every extreme and the workload. Regular performance test in every cycle will allow team to benchmark their product or a release, and at the end they can share the results with all stakeholders. The proposed framework and associated workflows can be easily plugged into to CMMI (Capability Maturity Model Integration) standards in order to streamlined performance testing process and artifacts to continuously maintain the quality standards.

Easy Customization – Performance testing framework that the authors proposed never define a *"hard rule"* for deciding circumstance under which performance testing would be needed in implementation life cycle. As a result, it makes easy to customize the test suite based on the client's requirement with a minimum effort and time. In parallel, from the maintenance perspective, the best practices use at the script development stage, enables our Test Engineers to revisit the code efficiently. The parameter driven architecture pre-configured for every test scripts ensures that the changes are versatile to accommodate and our repository of test scripts, provides the capability to assess whether the previous system was scaled to the intended new demand. These test results can be used to plan for future growth and avoid over-investment in hardware and investigate the possibility of assessing stability, whether the application remain stable or whether there are other technical faults that may compromise the stability of the application with new hardware.

6. CONCLUSION

Carl [17] state that, risk related expenses, opportunity cost, business continuity or reputation can be addressed by conducting performance testing. Failure to adhere to performance testing would affect the market share, decrease in ROI (return on investments) in overall IT spending and eventually the revenue margin since disgruntled users and user abandonment are consequences of going live in to production without conducting standard performance testing [18].

In modern business arena, the healthcare, insurance, omni-channels and education have become highly competitive, dynamic and evolving industries. These service providers would rely on technology to add multiple channels and parallel processing of high transaction volumes in the shortest time possible in order to meet consistent delivery of strong customer service associated with various hardware and network devices in par with rapid technological replacements. Thus, this proposed process framework should be tested in above domains with massive data volumes.

Measuring the performance in such hardware peripherals are a fundamental challenge and needs to inhabit a vast range of technical competencies as well as versatile on business process and their impact to final deliverables. Thus, the performance testing in such domains are critical in improving efficiencies, enhancing the customer experience and achieving new regulatory compliances in future. Therefore, the authors see a potential opportunity to test this proposed process framework against the hardware manufacturing.

Social technologies are merely managed to survive without a proper performance testing. Further, many businesses use social technologies as a realistic source for getting direct and open customer feedback in their co-creation process. Therefore we have a precious opportunity to study and enhance the capabilities of this proposed process framework in various social technology platforms with possible *"big data"* sphere.

7. ACKNOWLEDGMENTS

The completion of this paper was made possible through the help and support from many well wishers, including: participants, process consultants, advisors and in essence, all sentient beings. Especially, we dedicate our acknowledgment of gratitude towards the following significant contributors and advisors.

First and foremost, we would like to thank Dr. Sankalpa Gamwarige for his most inspirational support and encouragement given for writing this paper. Secondly, we would like to thank Mr. Dimuthu Nawarathna who personally helped us review this paper and provided valuable insights for improvements.

8. REFERENCES

[1] Meier, J., et al. 2007. Type of Performance Testing, [Online]. Available: http://msdn.microsoft.com/en-us/library/bb924357.aspx

[2] Testarconsulting., , n.a. Performance Testing, [Online]. Available: http://www.tecstarconsulting.com/?page_id=72

[3] Xiang, G. 2006. Software Performance Testing, [Online]. Available: http://www.cs.helsinki.fi/u/paakki/gan.pdf,

[4] Hansjorg, S. 2010. Load and Performance Testing for Software Applications, [Online]. Available: http://www.swisscom.com/content/dam/its/documents/whitepapers/wp_performance-testing_de.pdf

[5] Oracle. 2010. Performance Testing, [Online]. Available: http://www.oracle.com/technetwork/articles/entarch/performance-testing-095962.html

[6] Jamo Solutions. 2013. Performance Testing of Mobile Applications, [Online]. Available: http://www.jamosolutions.com/wp-content/uploads/2013/08/White-Paper-M-eux-Test-Performance-Testing-of-Mobile-Applications.pdf

[7] Neotys. n.a. The web testing solution for optimizing all your applications, [Online].Available: http://www.neotys.com/product/overview-neoload.html

[8] Nguyen, T. H. D. et al. 2011. Automated Verification of Load Tests Using Control Charts. In 18th Asia-Pacific Software Engineering Conference, Ho Chi Minh, pp. 282-289

[9] Malik, H. 2010. A methodology to support load test analysis. In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, Cape Town, Vol. 2, pp. 421-424

[10] Stuart, M. 2006. Open source tools – Performance testing”[Online].Available: http://www.myloadtest.com/open-source-tools-performance-testing/

[11] Rapoza, J. 2006. Putting open source to the test”,[Online].Available:http://www.eweek.com/c/a/Web-Services-Web-20-and-SOA/Putting-Open-Source-to-the-Test/

[12] Samrat, J. 2012. Software Testing tools, [Online].Available:http://www.softwaretestingsoftware.com/software-testing-tools/

[13] Ben, Y. 2013. Agile Performance Testing Life Cycle, [Online].Available:http://h30499.www3.hp.com/t5/HP-Software-Developers-Blog/Agile-Performance-Testing-Life-cycle/ba

[14] Rowlands, B. 2005. Grounded in Practice: Using Interpretive Research to Build Theory, *The Electronic Journal of Business Research Methodology* Vol. 3 Issue 1, pp 81-92, [Online]. Available: www.ejbrm.com

[15] Yin, R. 2003. Case study research: Design and methods (3rd ed.). Thousand Oaks, CA: Sage

[16] Patton, M. 1997. Evaluation: The New Century Text, Thousand Oaks, CA: Sage Publications

[17] Carl, H. 2010. Performance Testing. [Online]. Available: http://maestro.equella.com/file/3b42dbaf-2f81-2f2b-a4e5-dfb3e0134269/2/EQUELLA%20Performance%20Testing%20Whitepaper.pdf

[18] Accenture. 2008. Calculation ROI of Performance Testing. [Online]. Available: http://isqtinternational.com/Lily%20-%20Accenture.pdf