International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 4, Issue 7, July 2015)

# Analysis of Object Oriented Software by Using Software Modularization Matrix

Anup[1], Mahesh Kumar[2]
[1]M.Tech Student, [2]Assistant Professor, Department of Computer Science and Application, RPS College, Mahendergarh, Haryana, India
[1]navdeepsourav@gmail.com

***Abstract:*** **- Software metrics deals with the measurement of the software product and the process by which it is developed. These measurements of the software process and product are studied and developed for use in modeling the software development process. These metrics and models are then used to estimate/predict product costs and schedules and to measure productivity and product quality. Following are the few metrics that are used to measure the software quality: Quality Metrics, Complexity Metrics, and Alstead's Product Metric. The goal of software metrics is identification and measurement of the essential parameters that affect software development. Good metrics should facilitate the development of models that are capable of predicting process or product parameters. Thus ideal metrics should be: Simple, Objective, valid, Robust. The challenge for software engineer is to be able to interpret metric data on an on-going basis, so that continuous improvement can be made to maximize the quality potential within a limited project budget. So we improve quality measurement metric of software using modularization concept.**

***Keyword:-*** **HP Load Runner, Modularization Matrix**

## I. INTRODUCTION

Software metrics are measures of software. Their primary use is to help us plan and predict software development. We can better control software quality and development effort if we can quantify the software. This is especially true in the early stages of development. Research has been done in the area of predicting software maintenance effort from software metrics. Software metrics are valuable entity in the entire software life cycle. They provide measurement for the software development, including software requirement documents, designs, programs and tests. Rapid developments of large scaled software have evolved complexity that makes the quality difficult to control. The successful execution of the control over software quality requires software metrics. The concepts of software metrics are coherent, understandable and well established, and many metrics related to the product quality have been developed and used.

It is essential to introduce definition of software metrics. Software metrics provides measurement of the software product and the process of software production. In this paper, the software product should be seen as an abstract object that begins from an initial statement of requirement to a finished software product, including source and object code and the several forms of documentation exhibited during the various stages of its development. Good metrics should enable the development of models that are efficient of predicting process or product spectrum. Thus, optimal metrics should be: [1]

- Simple, precisely definable—so that it is clear how the metric can be evaluated;
- Objective, to the greatest extent possible;
- Easily obtainable (i.e., at reasonable cost);
- Valid—the metric should measure what it is intended to measure; and
- Robust—relatively insensitive to (intuitively) insignificant changes in the process or product.

### *1.1 Classification Of Software Metrics*

There are three types of software metrics: process metrics, project metrics and product metrics. [2]

*1) Process Metrics:* Process metrics highlights the process of software development. It mainly aims at process duration, cost incurred and type of methodology used. Process metrics can be used to augment software development and maintenance. Examples include the efficacy of defect removal during development, the patterning of testing defect arrival, and the response time of the fix process.

*2) Project Metrics:* Project metrics are used to monitor project situation and status. Project metrics preclude the problems or potential risks by calibrating the project and help to optimize the software development plan. Project metrics describe the project characteristics and execution. Examples include the number of software developers, the staffing pattern over the life cycle of the software, cost, schedule, and productivity. [3]

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 4, Issue 7, July 2015)

3) *Product Metrics:* Product metrics describe the attributes of the software product at any phase of its development. Product metrics may measure the size of the program, complexity of the software design, performance, portability, maintainability, and product scale. Product metrics are used to presume and invent the quality of the product. Product metrics are used to measure the medium or the final product.

Software quality, as stated earlier, depends on a number of factors. Also as theorized by David & Garwin, quality is a complex as well as multifaceted concept, which can be viewed according to different points of view as follows:

1) *User View:* The user viewpoint of software quality tends to be a lot more concrete and can be highly subjective depending upon the user. This view evaluates the software product against the user's needs. In certain types of software products like reliability performance modeling and operational products, the user is monitored according to how they use the product.

2) *Manufacturing View:* This viewpoint looks at the production aspect of the software product. It basically stresses on enforcing building the product without any defects and getting it right the first time rather than subsequently making a defective product and spending valuable project time and more importantly costs ironing out the defects or bugs at a later stage. Being process based, this viewpoint focuses on conformity to the process, which will eventually lead to a better product. Models such as ISO 9001 as well as the Capability Maturity Model do encompass this viewpoint that stress on following the process as opposed to going by specification. However, that being said, the theory that following the best and high quality manufacturing process will automatically lead to a better product cannot be inferred. The critic's viewpoint is that following an optimized and high quality product manufacturing method can also lead to the standardization of a product making it more of a commodity rather than a standout product. That being said, there have been a lot of industry example where the philosophy of "doing it right" the first time been profitable. Also both the models CMM as well as the ISO, indirectly do imply by following the principle of "Documenting what you do and doing what you say" helps in improving the product quality.

3) *Product View:* The product viewpoint looks at the internal features as well as the characteristics of the product. The idea behind this viewpoint is that in case a product is sound in terms of the features and functionality it offers, and then it will also be favorable when viewed from a user viewpoint in terms of software quality. The idea is that controlling the internal product quality indicators will influence positively the external product behavior (user quality) There are models trying to link both the views of software quality but more work is needed is this area.

4) *Value based view:* The value-based view becomes important when there are lots of contrasting views, which are held by different departments in an organization. For example, the marketing department generally takes a user view and the technical department will generally take a product-based view. Though initially these contrasting viewpoints help to develop a 360- degree product with the different viewpoints complementing each other, the later stages of the software product development might have issues

The issues arise when there might be a set of change proposed to a certain view that can end up throwing a conflict in the other view. For example, say the marketing department (user view) wants changes to the user interface that are not technically feasible (product view).

This is where a value-based view comes into play helping resolve such conflicts so that the software product is not delayed indefinitely. The value-based viewpoint looks the conflict with a cost to benefit angle. It help in resolving such issues by looking at the issue in relation to terms like costing, constraints, resources, time. Using this viewpoint, it's possible to resolve interview conflicts helping to keep the software product on track and within initial cost and timeline estimates.

Making a web site does not end with putting all the media and software together. Actually, web site work never ends. When all the design is done, you have to test the site first before sending it to the World Wide Web for the world to see. There is site management software that can do this for you. These software can help reconnect graphics that may have been accidentally moved, change the name of a file and re-link it and so many other things.

Aside from the site management software, you also have to the quality of your website. Your site has to be tested, fixed, retested and fully documented.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 4, Issue 7, July 2015)

If any there is any software running on your website (which you most probably will), this has to be tested.

Some of the things that have to be checked for quality assurance are multiple browser compatibility, download time of graphics, Flash components, or streaming media, hardware requirements, memory size requirements, connection speed of user, and load (number of users your website can accommodate). There are many companies now that are developing software specifically for quality assurance. But this software is expensive. Usually, e-commerce sites are load tested as they have a great deal of software running on their sites.

Some of the other test types are functional tests (makes sure features work), stress tests (site is tested on computers with different hardware specifications), regression tests (defines how site will be tested in the next phases), boundary analysis (tests the limitations of site such as entering information in forms) [04]. Sometimes the best way to test your site is by having an actual person go through your site and let him tell you the problems he encountered. These people are usually called testers. Some of these software run test that simulate testers.

## II. Motivation

Existing web-application performance-testing tools offer a broad variety of functionality. However, none of them combines all the functionality we expected to use in our projects. This resulted in my decision to define and implement such a tool. License fees for existing performance testing tools vary from zero to thousands of U.S. dollars. For projects that are tight on budget, the tool must be free. The only expenses allowed are for test preparation, execution, and reporting. To keep these expenses down, a test definition must be easy to maintain not only during the project, but it must be available for reuse on future projects. The tool must be able to perform as much of the common activities as possible. Such activities are request-header generation or automatic retrieval of images included in the page.

Most web applications this tool would test depend heavily on user input; therefore, the tool must offer easy manipulation of user data and requests that are sent to the application server. User input is not only data a user enters into web forms, but the selection of functionality as well. In web applications, this means clicking on links displayed on page. Not all functionality is used equally; thus, it must be possible to branch in test execution randomly, having different weights for individual actions.

Based on user input, applications generate responses, and under very high load, they can fail either by collapsing and sending nothing or only an error code or by responding with a page that has an error message inside. The desired tool must be able to parse the response and check for user-defined data. In fact, it must incorporate some functionality that covers functional testing to be able to detect the application errors. According to its finding, it must be able to branch in test execution. Target systems running our web applications are always off-site, either at the customer's location or in hosting centers. Because clients for performance testing in such installations are mostly only servers running on Linux or UNIX, the performance-testing tool must run on such systems.

In an ideal world, everything works perfectly when the test is run. In reality, first runs often show problems in server configuration or in application itself. To identify such flaws, data gathered during the test can be helpful. In such a case, pre-defined reports and graphs might be not enough. If the tool were able to generate data for statistics software, it would be possible to mine the data and help to find the application bottleneck.

## III. Problem Description

Software developed with interlinking on many executable programs. These programs are developed by many members of a team with good co-ordination. When software is ready to perform than before put it in market the quality of that software tested. We all use many software for many purposes, all software perform well in field for which they developed. So testing of quality is a step that is necessary to perform. Testing of complete software is very difficult because it is so difficult and time consuming to find error line in whole code and it may occur that to correct that error due to which a new take place. So we propose a modularization metric that perform well as compared to existing.

## IV. Objective

*Step 1:* Study about metrics of website user for quality measure in load.

*Step2:* Source code of software required on which we implement testing algorithm and measure quality.

*Step3:* Use existing metrics to measure quality of that website.

*Step4:* The parameters of metrics generate result and we save that results for further use in our research work.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 4, Issue 7, July 2015)

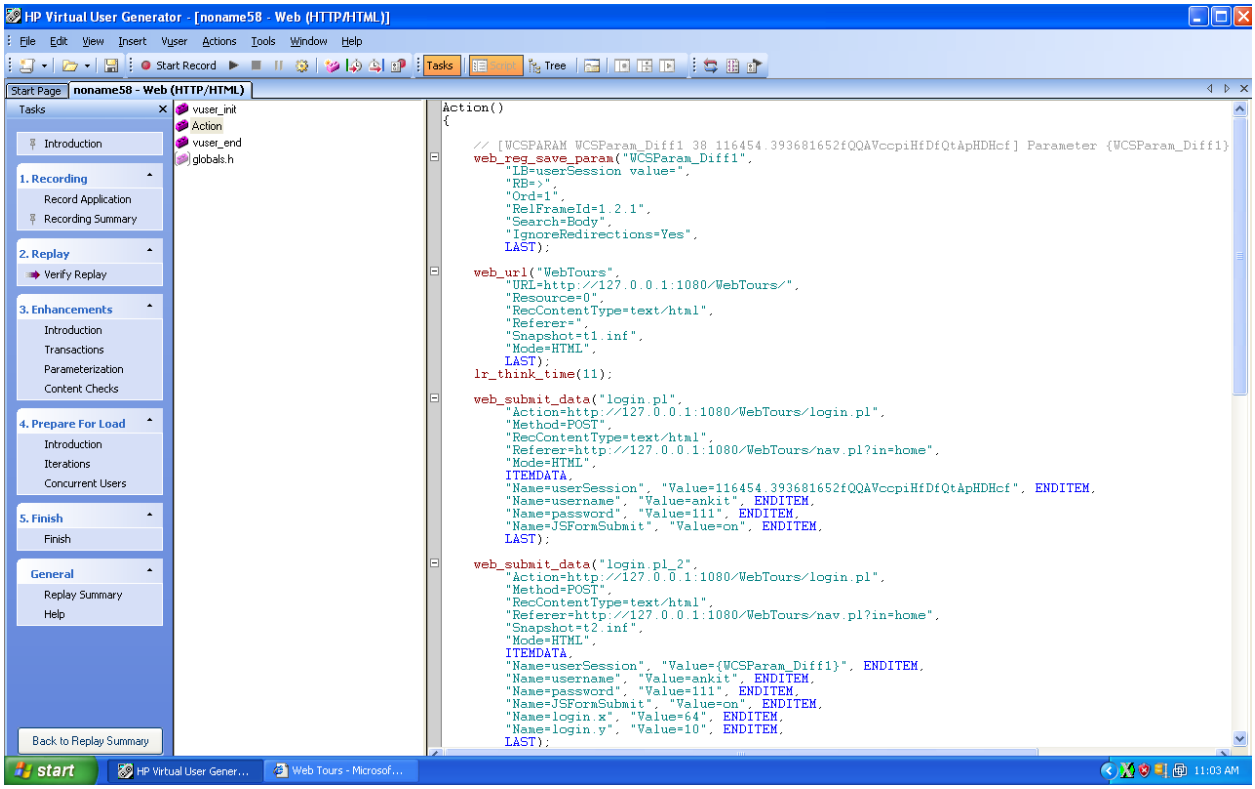
*Step5:* we propose a HP Load Runner on website for improvement in performance.

*Step6:* Perform this metric on that same code on which existing applied and get results using new metric.

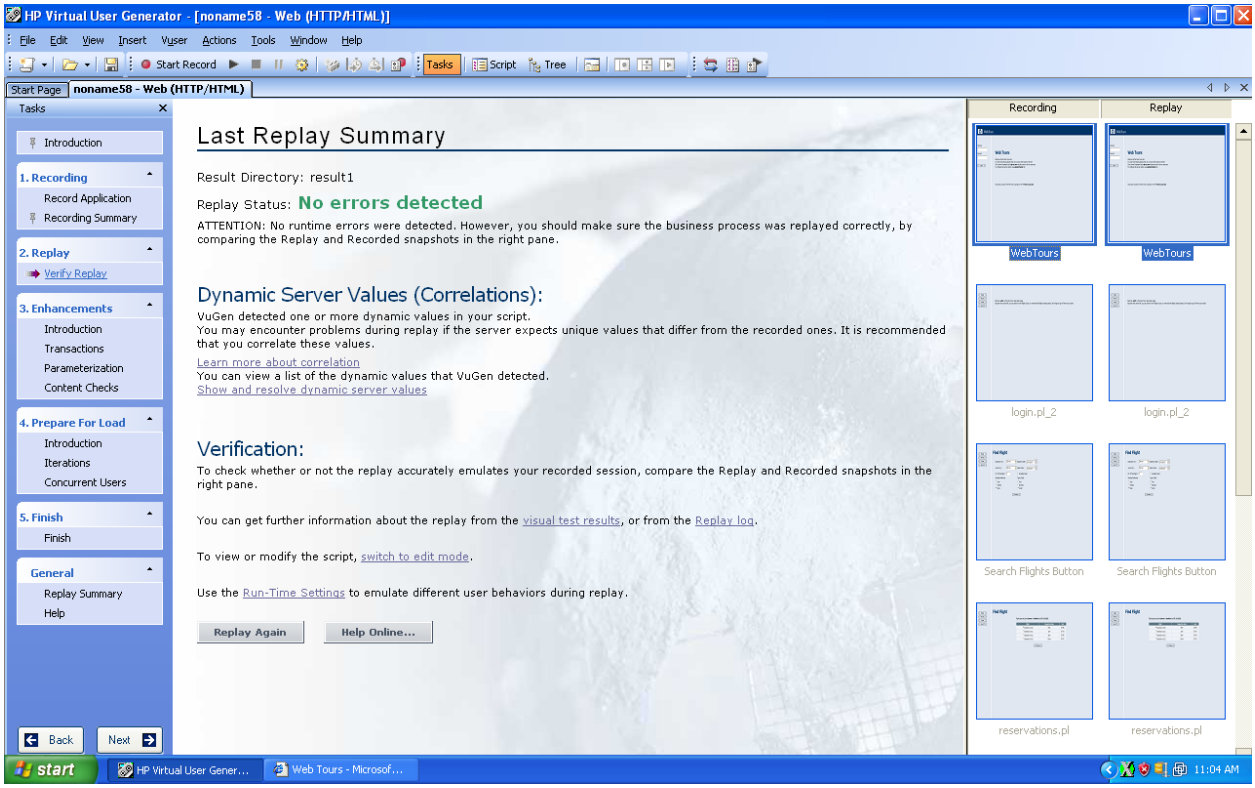
*Step7:* Results of existing is compared with proposed metric and we will get conclusion.

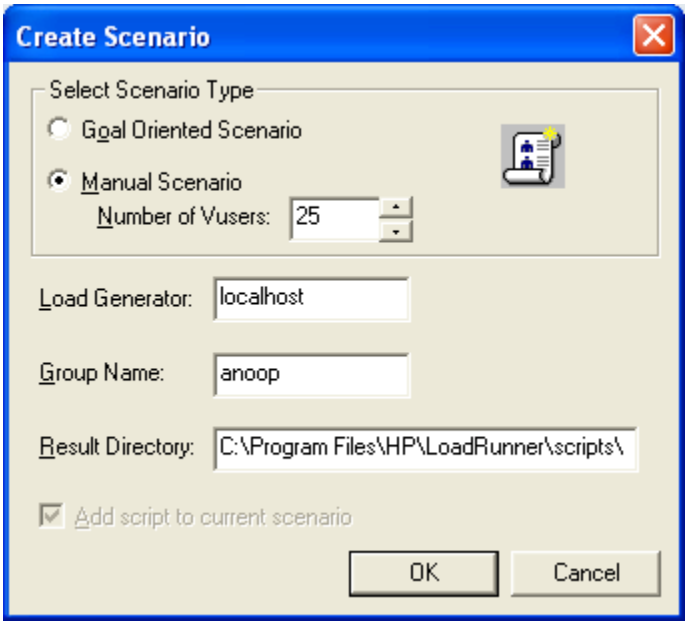
*Step8:* Our objective is to make new metric better than existing.

## V. RESULTS

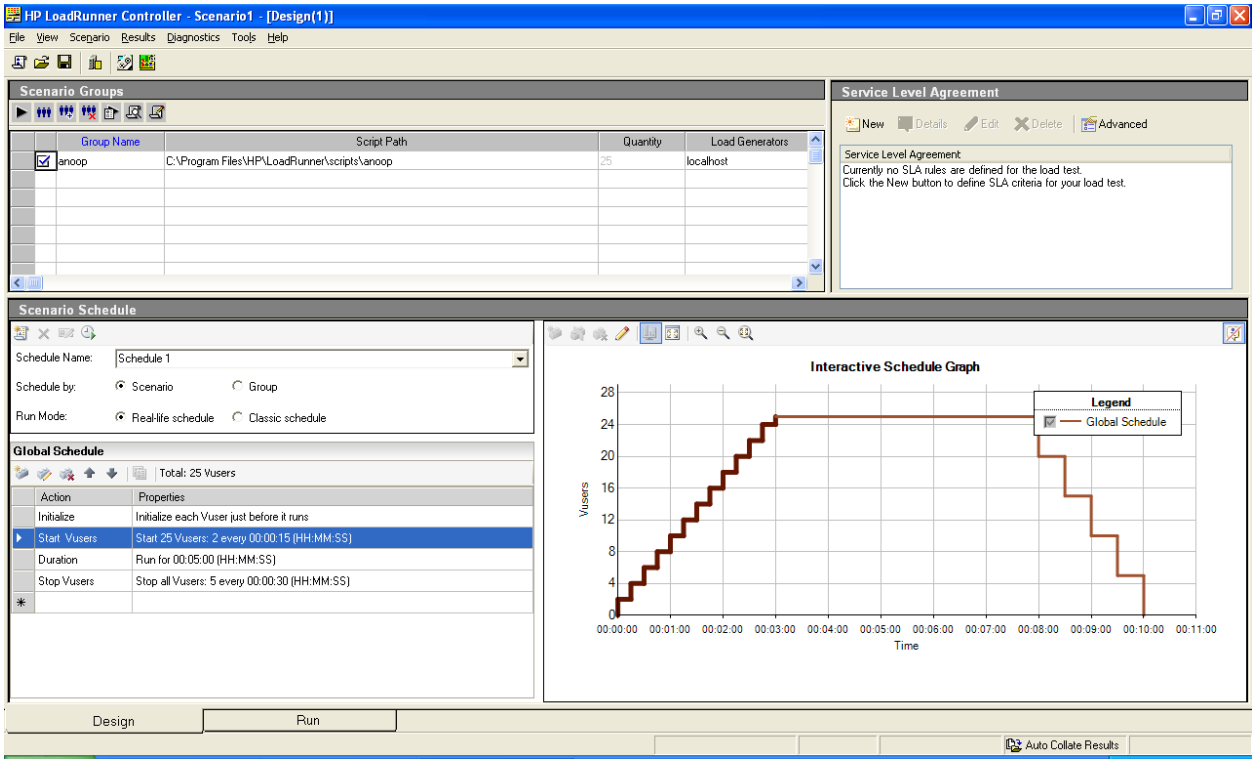


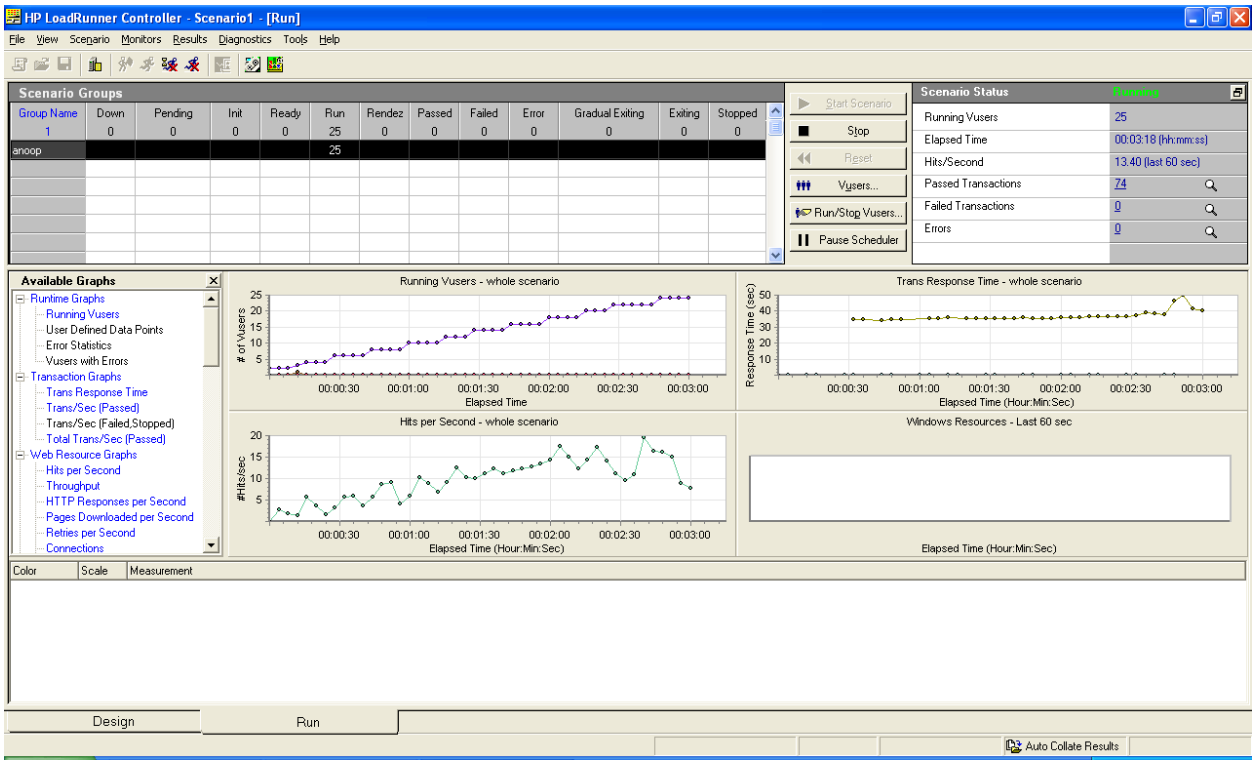
International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 4, Issue 7, July 2015)



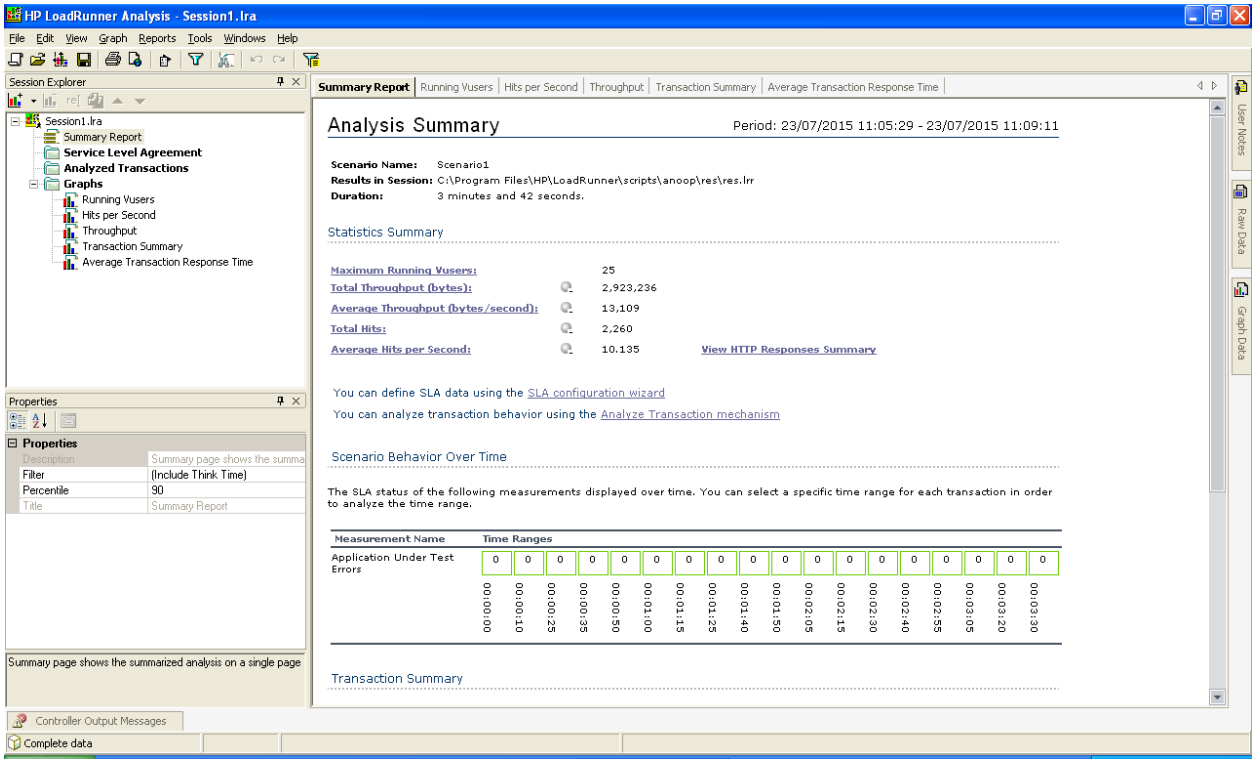



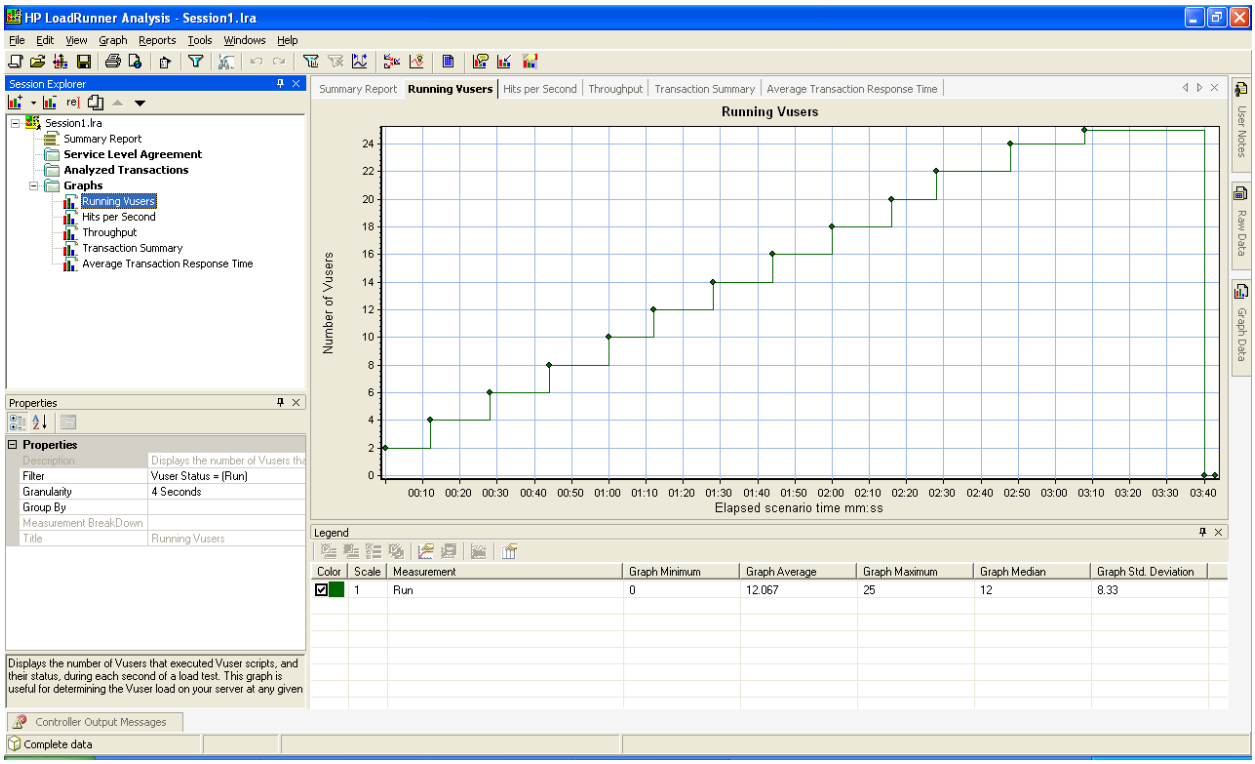
International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 4, Issue 7, July 2015)



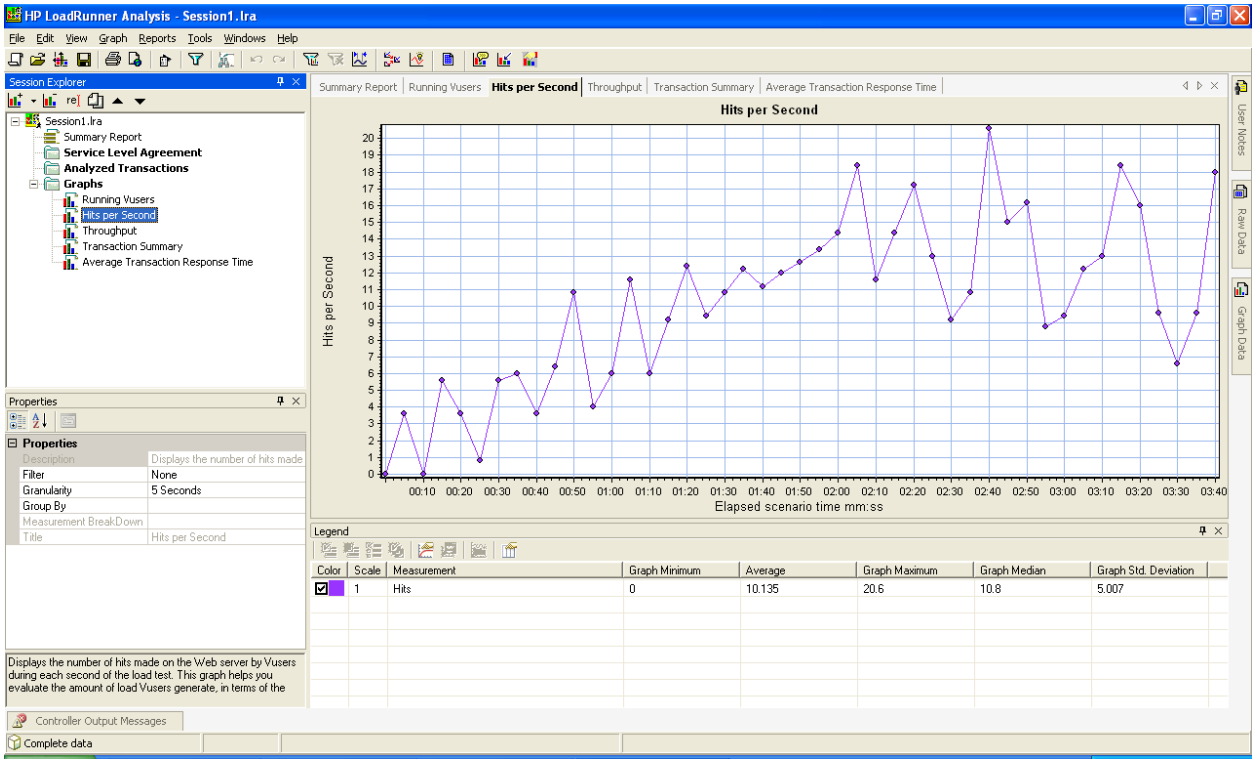



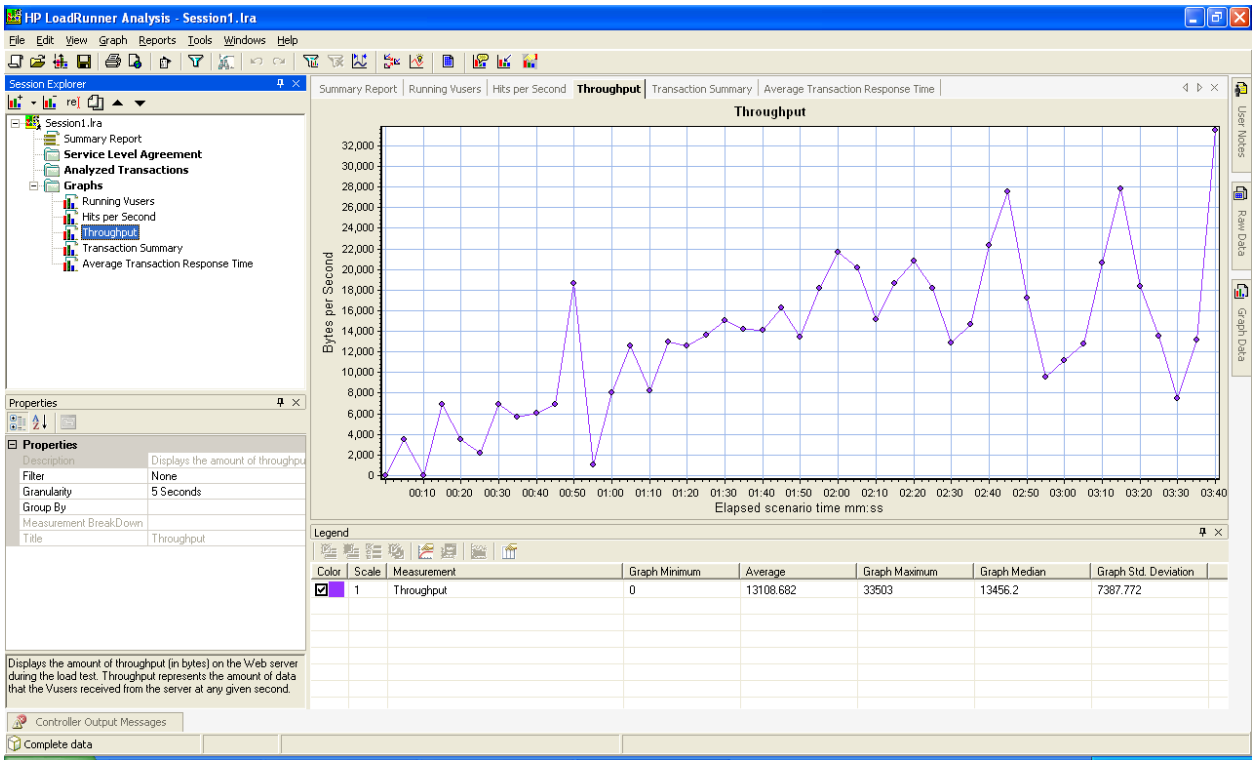
International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 4, Issue 7, July 2015)



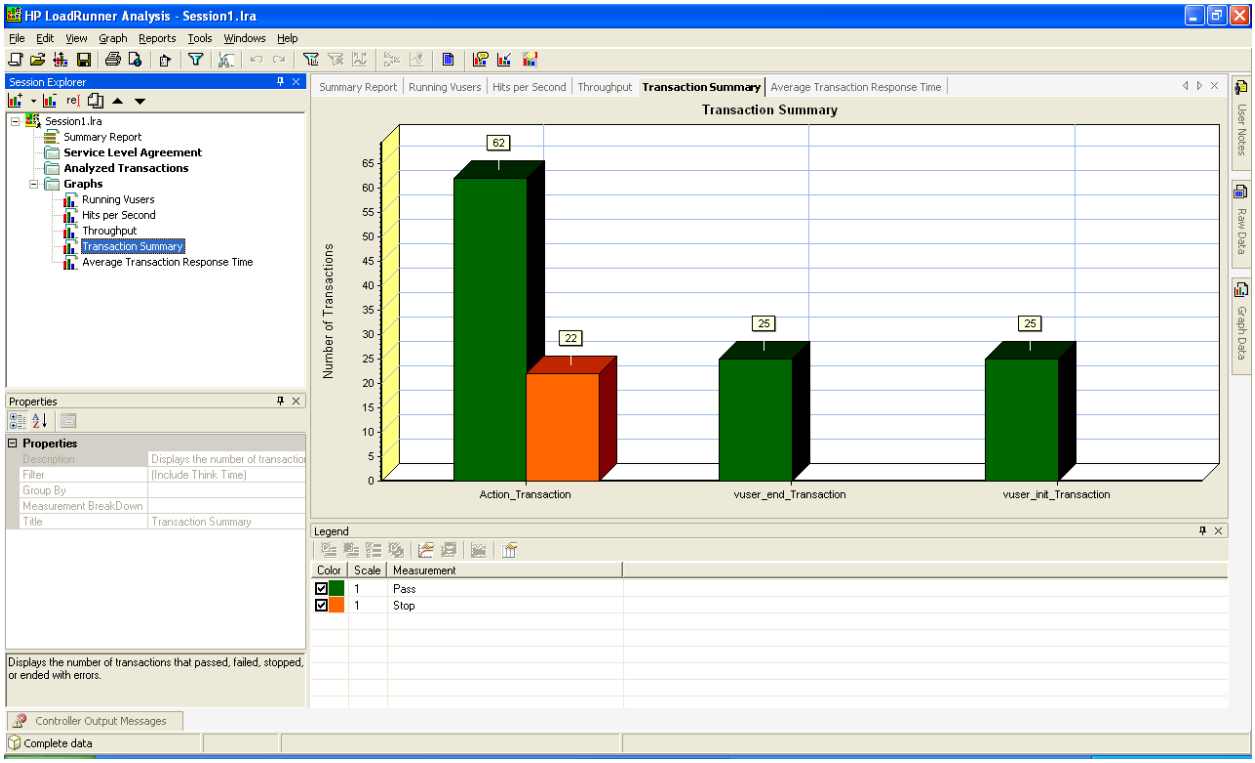



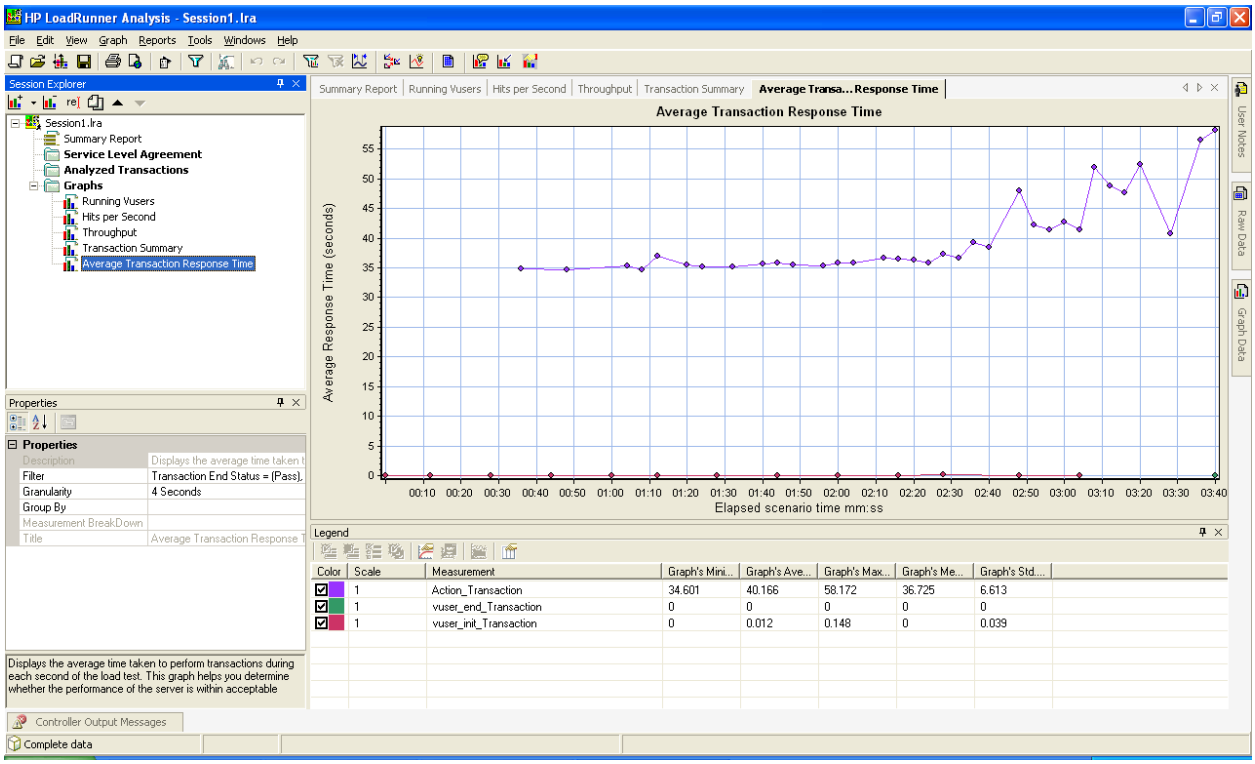
International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 4, Issue 7, July 2015)



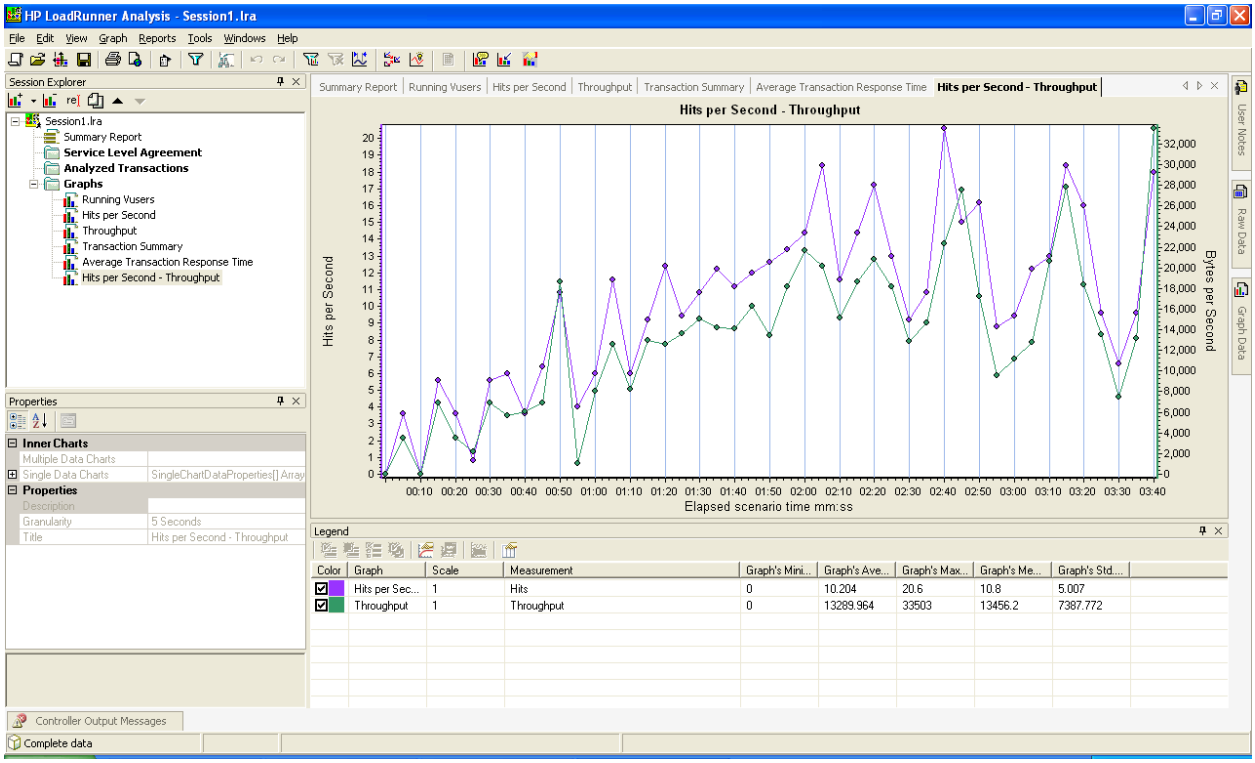



International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 4, Issue 7, July 2015)



International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 4, Issue 7, July 2015)




## VI. CONCLUSION

HP LoadRunner software, used by thousands of businesses around the world, is the industry standard software for performance validation. It enables an efficient and robust means to verifying that your application's architecture is built for more efficient performance and reliability, so you can deploy with high quality and confidence. We introduced performance testing and analysis tools HP LoadRunner that can be used to realize effective web application performance tests with minimal overhead. Analysis of test results is useful for system designers to make the best choice of the optimal platform in which to build web applications. Performance tools help software developers to find out bottleneck in system performance, as well as performance models to help them to develop web applications with high performance.

## REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lips Software Metrics SEI Curriculum Module", SEI-CM-12-1.1, December 1988

[2] Tu Honglei1, Sun Wei1, Zhang Yanan1, "The Research on Software Metrics and Software Complexity Metrics", International Forum on Computer Science-Technology and Applications, 2009

[3] http://www.pearsonhighered.com/samplechapter/0201729156.pdf.

[4] Burdman, J. 1999. Collaborative Web Development: Strategies and Best Practices for Web Teams. USA: Addison-Wesley

[5] Gerrard, P. (2000a). Risk-Based E-Business Testing: Part 1 – Risks and Test Strategy. Retrieved June 15, 2001, from the World Wide Web: http://www.evolutif.co.uk/articles/EBTestingPart1.pdf

[6] Crispin, L. (2001). Stranger in a Strange Land: Bringing QA to a Web Startup. Retrieved June 15, 2001, from the World Wide Web: http://www.stickyminds.com/docs_index/XUS247559file1.doc

[7] Miller, E. 2000. Website Testing Retrieved October 21, 2002 from the World Wide Web: http://www.soft.com/eValid/Technology/White.Papers/website.testing.html.