

# PERFORMANCE REQUIREMENT ANALYSIS AND EVALUATION MODEL OF WEB APPLICATIONS

Pankaj N. Moharil<sup>1</sup>, Ch.Sanjay<sup>2</sup>, Sudarson Jena<sup>3</sup>

<sup>1</sup>Research Scholar, SGV Amravati University, Amravati

<sup>2</sup>GITAM University, Hyderabad

<sup>3</sup>GITAM University, Hyderabad

## Abstract

One of the principal characteristic of a software system is software Performance. Software Performance analysis helps to mitigate the risk of redesign. So the software development people pay more attention one software performance. Development People has to state accurate software performance requirements, proper plan to test cases, and utilize appropriate skills to examine the software performance characteristic which helps in developing a software system of admissible performance. But insufficiency of a management schema for software performance engineering may reduce efficacy of the software performance skills. In this paper, it proposes the Performance Requirement Analysis and Evaluation Model (PRAEM) as a software performance mainframe schema settled on the particularization of quantitative analysis and loads; PREM classifies performance requirements into three steps. At each step, it shows that which information should be included in the requirement, and which techniques can be used to estimate the performance.

**Keywords:** Requirement analysis, Qualitative, Quantitative, Load estimation, Target specification, Performance Testing.

\*\*\*

## 1. INTRODUCTION

Software Requirements analysis is the first phase in the systems engineering process and software development procedure. Software Requirement analysis in software engineering encircles the particular tasks that go into actuate the needs or redesign product, evaluating at all collision of requirements of the various stakeholders, such as beneficiaries or users [1].

Software Requirements analysis is interpretative to the favorable outcome of a development project. It should be actionable, measurable, and testable, associated to recognized business requirement or opportunities, and propound to a level of detail enough for system design. Requirements can be functional or practical and non-functional or non practical. A software system will be build with satisfactory performance; the development people or developers have to take software performance into contemplation through the entire software development cycle. Software performance issues are inscribed by many models, techniques and with different methodologies. Development peoples require applying different software performance engineering skills or techniques to achieve the pine performance [2]. But insufficiency of a management schema for software performance engineering may reduce efficacy of the software performance skills. As illustration, In case unessential sophisticated software performance model is utilized all along the initial phases, the software development peoples may require making presumption as anonymous yet compulsory component to the model [3]. These presumptions can distribute the software performance

model unreliable or alike abortive. The development peoples have to choose accurate or best performance skills or methods, which compassionate performance attributes of the system. This paper proposes the Performance Requirement analysis and Evaluation Model (PRAEM) just as schema for execution of software performance development process. Software Requirement analysis involves many graceful conceptual skills; it can be prolonged & difficult process. Before performance requirement analysis it is primary to recognize all the stakeholders, pay attention on their requirements and shelter that they understands the suggestions of the new suggested system. Analysts may appoint several different techniques to evoke the requirements from the customer. To understand the exact & accurate requirements of the stakeholder prototyping and use cases techniques are introduced. These techniques help to produce the useful system which encounters the entire business requisite. Methodical examination of software requirements is called as requirement engineering. It is also mentioned as requirements gathering, or requirements specification [4]. After requirement specification it is important that performance target should be derived. Web application performance target gets derived from relationship between transactions, batch jobs and reports depend on business workload. Habitually customers give number of simultaneous users and respond time objectives, yet not output targets. So the Performance target validation becomes an important factor in specifying the performance targets requirements. Through validation of performance requirements system load will be get calculated with number of users, throughput or output and response time. It is important to know the connection between the number of

uses, throughput or output and response time. Network traffic load and release times should be validated [5]. The Workload estimation includes no of functionalities of software system with estimated response time. At each phase, the model narrates how the software performance requirements and its related test cases are framed and sets actual perspective to analyze the performance of web applications.

Section 2 of this paper presents performance requirement analysis and evaluation model. In section 3 result analysis is discussed. Finally section 4 of this paper presents the concluding remarks.

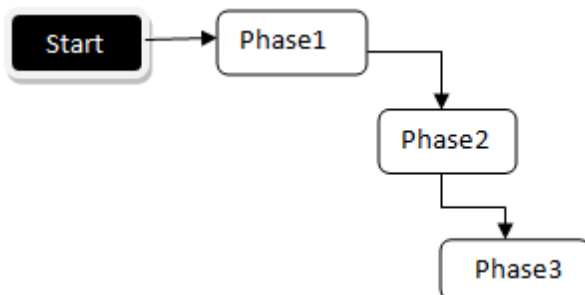
## 2. PERFORMANCE REQUIREMENT ANALYSIS AND EVALUATION MODEL

In PRAEM, three phases are organized according to performance requirement specifications, Performance testing, and activities. A requirement within PRAEM enumerates more performance characteristics of the software system with giant accuracy.

**Phase1.** Qualitative and Quantitative performance requirements are specified.

**Phase2.** Estimate the average workload and average Time.

**Phase3.** Average workloads are collected and Target or Goals are specified.



**Fig.1** Performance Requirement Analysis and Evaluation Model

Basically Software Requirements are categories in two categories *Functional* and *Non-functional*.

### Functional Requirement

On what the system is supposed to do are mention as Functional requirements. It extends with the explanation of how the system will run from the user's point of view. Business stakeholders and technical people should be allowed by the functional requirements to go through the system, to observe each and every feature of the system how it should work – before it is built [7].

### Non-Functional Requirements

Non-functional requirements can be clustered into the following groups:

1. Performance requirements: Performance requirements typically concern with the speed of the operation of a system.
2. Reliability requirements: Reliability requirements usually associate with the reliability of the system.
3. System interface requirements: Interface requirements mention to the software interfaces provided by the system.
4. Security requirements: Security requirements associate with the access limitations to the system.
5. Standard requirements: all the requirements associated with standards should be grouped into this category.
6. User interface requirements: Number of simultaneous user interfaces given by the system.
7. Documentation requirements: Documentation requirements capture those requirements that are related to the software documentation.

System performance is calculated with an attribute Response time and Throughput. Response time is the time required to complete an operation and within assign time period how much quantity on an operation get completed is the output or throughput. Mostly Performance requirements get calculated regarding quantity, quality, response time coverage and readiness. [8]. Depending on the system life cycle feature functionalities performance requirements will be coactively evolved; and performance requirements will be describe the concerning assurance standards in their estimates, the number of importunity to system success, and their affiliation with other requirements [9].

Performance Testing: software Performance testing identifies whether the software System accomplish the expected performance objectives including response time & workload. PRAEM model performance testing approach describes that how the performance test cases should be design to reflect the performance requirement of a PRAEM phases.

In this section, PRAEM Model shows the phases to specify the performance requirement in efficient manner and achieve the required goals.

### 2.1 Qualitative and Quantitative Performance Requirements

In phase 1 PRAEM model specifies two i) Qualitative and ii) Quantitative performance requirements. By specifying PRAEM phase1 requirements, a customer estimates the operations for which performance affairs. Phase1 initial points help developers and end users to specify their higher requirements.

#### 2.1.1 Qualitative Approach

A Technique is introduced to specify Qualitative performance requirement. Through PRAEM phase1- The qualitative performance requirements analysis technique is provided as follows:

1. Conduct the thorough workload analysis.

2. Set performance targets or goals.
3. Check the performance targets in light of technology.
4. Analyze transactions/interactions by considering the factor transaction frequency, Complexity and workload which results in a bombastic display of information & appraise strain opportunity considering the same.
5. Analyze transactions, reports, batch jobs by considering the factor transaction frequency, complexity and workload that can account boundless consumption of system resources.
6. Evaluate transactions, reports, and batch jobs by considering the factor transaction frequency, complexity and workload that require workload management.
7. Adopt and Analyze change in policies in business processing by considering the factor transaction frequency, complexity and workload.
8. Provide space to calculate performance requirements of all types of transactions that design performance issues in production.
9. Performance testing with qualitative measurement specifies the appropriate test cases, run the scenario and then catches the performance measurement.

PRAEM performance models cleft the performance scenario into a few small phases. To Estimate the response time for each small phase. It describes a PRAEM phase-1 with response time requirement of an operation with three parts:

*Preparation:* preconditions should be described before the operation can be carried out.

*Event:* Response time measured of an operation when an event performed within operation. As the event happened response time measurement starts.

*Time:* The time desired to complete the operation.

### 2.1.2 Quantitative Approach

A Technique is introduced to specify Quantitative performance requirement. Through PRAEM phase1- The quantitative performance requirements analysis technique is provided which uses three scenarios:

1. Acquiring performance targets
2. Authenticating performance targets
3. Analyzing interference crosswise transactions specified by the targets.

Performance testing with quantitative measurement specifies average workload & estimated time. The estimation of workloads has specification of performance requirements and test cases. [10].The process of estimating average workload specification describes number of events.

## 2.2 Estimate the Average Workload and Average Time

PRAEM phase2 Estimating the average workload requirement of a system involves load assign to the system. The Inputs assign within workload are to be collected by using following five areas:

- a) Online processing Action
- b) Batch processing

- c) Reports generation process
- d) Data volumes
- e) User interaction

a) *Online processing Action:* By workload it means that the number of actions that need to be processed simultaneously. Online processing action is categories as

1. Transaction necessity,
2. Concurrent Processing
3. Work breeze Processing

b) *Batch processing:* A technique of collecting process with input parameter and data in huge quantity, either at the assign time period of the day or time period for a week or month. With Batch processing the huge data processing compellation in assign time period.

c) *Reports Generation process:* Many business and management departments required reports. Reports can be planed systematically, On the other hand, there some improvised reports that are requested on demand.

d) *Data volumes:* Growth in business data bangs performance greatly by the growth in repository estimated amount and unfavorable results of data volumes on the performance of transactions & reports. It becomes compulsory to have a business data model. It is compulsory to seize the sizes and volumetric figures of business operations.

e) *User interaction:* user interaction regarding user details and usage provides appropriate understanding of business workload and crosscheck against transaction and report workload data.

External interfaces for the business applications Performance testing with quantitative measurement specifies average workload & estimated time. With the estimated workloads performance requirements and test cases are specified. [11]. Workload identification reports the frequency of the requests for the functionalities of software system with estimated response time. The process of estimating average workload specification describes number of events. Load level specifies the average workload used. [12].

## 2.3 Average Workloads are Collected and Target or Goals are specified

PRAEM phase3 specifies the average workload with performance Target, introduces technique to gather the performance requirement in collecting inputs on performance Targets. PRAEM phase-3 helps developer to make study of metrics of performance targets. Performance metrics are expressed as follows:

1. *Response time to each online interaction:* Communication time between screens is accomplished as response time. Commonly Response time is calculated in the act of mainstream amount or proportional amount.
- i). Positioned on type of interaction.

- ii). Positioned on network bandwidth.
- iii). Positioned on overall completion time of transaction.

2. *Completion time to each online transaction*: Time to require completing an online transaction. Completion time is comprehensive of the data entry time, time to response the activities and time required to collect information from end users.

3. *Delivery time for asynchronous transaction*: Time required to deliver the data through on line transaction to the other end and to maintain workflow environments. For example time require doing for an online correspondence operation where the time required submitting message to the message server and time required by the receiver to though which the message will reach to the receiver, which is the delivery time. There are some points to be considered while setting up assigning completion time goals:

1. Positioned on type of transaction and network Bandwidths
2. Positioned on number of stages or number of hops generated in application before reaching final destination.
4. *Transaction output*: Number of transactions processed per unit time accredited to the output.
5. *Batch completion time*: completion time require by the batch program along with the backups.
6. *Report throughput and completion time*: Reports might be evaluated as subsistence to batch operation and transaction processing. Presented reports have to be classified into anticipated and improvised reports. With an every category of report, output should be stated as mainstream and apex loads. With each improvised report, the completion time should be stated; however it should have addicted values to control the submission of ad hoc reports. Improvised reporting system is better than the other reporting systems.
7. *Resource consumption*: Resource consumption at each level of an operating system calls with the following factors:
  1. CPU usage time: Percentage of the time required by the CPU of the system when it is active.
  2. Memory utilization: How much memory utilized by the system RAM in the number of MB or GB.
  3. Network bandwidth: Complete Network utilization in terms of Kbps or Mbps.

### 3. RESULTS AND DISCUSSION

The performance requirements analysis has been illustrated through phases in PRAEM model, both qualitatively and quantitatively. PREAM Model assists to authenticate direct requirements and additionally actuate bang of adverse requirements. In this model each phase requirement specification is used to increase the efficiency or performance of the web applications. The required accuracy for performance requirements particularization anticipates on the kind of the project. How to maximize the web application throughput or output? This is the important question of the web application performance test system. Resource management is accommodated with some strategies to increase activities of the software system. Advance and updated technology skills with the design of system construction are utilized in assistance with resource allocation and lateral test. The approach presented in this

paper is to gather the performance requirements by looking into multiple ornaments to execute a performance test schema. Complexity of the performance tested web application systems is increasing rapidly with the appeal of added countenance, better features and less response time. The numbers of different performance testing tools are utilized for web application performance testing as the amount of user increases. An increase in performance requirement creates an issue of how to design a proper system to nourish the accomplishment of number of users by using a cost acceptable hardware to achieve the performance targets and the load balancing mechanisms. Requirement gathering through PRAEM model helps in some to achieve the load balancing and specifying the performance targets.

### 4. CONCLUSION

This paper shows how performance requirements can be stated by recognizing the essential segments in the requirements. Rule of deduction can be activated on the particular segments to accomplish the different arrangements of requirements. It uses PRAEM a model for software performance development, to determine the requirement particular segments and analogous characters. The model shows how PRAEM principles and related techniques can be applied. This paper model authenticates the particular segments through phases. The model phases are acceptable to all performance requirements. Performance requirements described with the phases are less inconclusive than particularization without using the phases. Performance requirements described within the phases are more complete than particularizations without using the phases. This believes that the required precision for performance requirements particularization or specification anticipates the type of project.

### REFERENCES

- [1]. V. Jain and J. Murty, *Performance Analysis of a Large Centralized Tax Processing Solution*, ROSETEA, Boston, 2007.
- [2]. R. Mansharamani and M. Nambiar, "*Performance Engineering of a Trading Exchange Risk Management System*", CMG, Orlando, 2010.
- [3]. Rajesh Mansharamani, *Performance Requirements Gathering & Analysis*, November 2011.
- [4]. Scott Moore, *Centralized Performance Testing*, software test Professional Spring, Springer, March. 27. 2012.
- [5]. Khaled M. Mustafa, Rafa E. Al-Qutaish, Mohammad I. Muhairat, "Classification of Software Testing Tools Based on the Software Testing Methods", *IEEE Second International Conference on Computer and Electrical Engineering*, 10/ 9. 2009.
- [6]. Dr.Liam Murphy, Performance Engineering, at university college, Dublin & Dublincity , *ACM 978-1-4403-1105-March.2.2011*.
- [7]. Xiang Gan , Software Performance Testing, University of Helesinki ,elesinki, *ACM Sep.26.2009*.

- [8]. H.sarojadevi DCSE , Performance Testing, *Journal of Information Engineering and Applications*, ISSN, Vol2, No.3, 2012.
- [9]. Sreedevi Sampath , Composing a Framework To Automate Testing of Operational Web Based Software , *CIS university of Delaware Newark, ACM* 978-4304-1105-8/9/12.2011.
- [10]. Melynda Eden, A Survey of Performance Modeling And Analysis Issues in Resource Management Across x86 based Hypervisors in Enterprise Cloud Computing Deployment, *INFORMS* Vol. 56, N0.10, Nov. 2010.
- [11]. Minzhi Yan, Hailong Sun, Xu Wang, Xudong Liu, Building a TaaS Platform for Web Service Load Testing , *Proc. of IEEE International Conference on Cluster Computing*, , 978-0-7695-4807-4/ 2012.
- [12]. Leslie Cheung, Leana Golubchik, Fei Sha, A Study of Web Services Performance Prediction: A Client's Perspective, *19th Annual IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* 1526-7539/2011.
- [13]. Luiz H. Nunes , Luis H. V. Nakamura , Heitor de F. Vieira, Rafael M. de O. Libardi, A Study Case of Restful Frameworks in Raspberry Pi: A Performance and Energy Overview, *IEEE International Conference on WebServices, 2014*