

# Performance Requirements Identification and Refinement: A Research

Gopichand Merugu

**Abstract**—Identification and refinement of performance requirements is important for successful development and deployment of the software product. The acceptance of the software product by the customer depends on the performance requirements which are incorporated in the software. For this, we need to identify all possible performance requirements required by all stakeholders. This paper presents the literature review about different approaches proposed for performance requirements identification and refinement

**Keywords** : Performance requirements, Identification, Refinement

## I. INTRODUCTION

Performance requirements are one of the thirteen nonfunctional requirements listed by 'IEEE-Std 830-1993' and most important to consider in software development[1]. A performance requirement is a statement that describes how well a function is to be executed or accomplished, or how well it is to be achieved. Performance requirements generally expressed in terms of timeliness, rate, quantity, degree and quality. Performance requirements have to be identified and specified completely in system requirements specification to avoid the problems that may occur at the end of the development of the product with respect to performance. If the product is developed with improper performance requirements specification definitely it leads to poor performance of the product. In these situations, the performance requirements have to be refined for next version of the product. So identification and refinement of performance requirements is an important task in software development. This thesis focuses on performance requirements identification and refinement.

## II. PERFORMANCE REQUIREMENTS IDENTIFICATION

Performance requirements identification is the process of identifying the necessary performance requirements for the software system according to customer needs. This is an important activity in requirements analysis[2] particularly in nonfunctional requirements analysis[3]. The related work about performance requirements can be found in [4, 5, 6, 7, 8, 9, 10,11, 12, 13,14, 15, 16, 17, 18, 19]. In [4] Edward presents the research work about the technology which can be used for specification of performance requirements of data processing. The initial research results indicate that a

technology based on petri nets, formal logic, and simulation can be used to describe and analyze some important aspects of data processing performance requirements. The work presented in [4] is only about the technology which supports the specification of performance requirements for a specific application called ballistic missile defense system software. The intent of the proposed approach is to avoid the introduction and propagation error at design of a complicated software system not at requirements level. The work presented in [5] describes the requirements exclusively for Ada compilers and their associated tools which are used for development of real time embedded weapons system. The work presented in this approach is all about the technical and performance requirements of Ada compilers and linkers. It is particularly about compilers and linkers specification not about complete system performance requirements specification which is one limitation of the proposed approach.

In [6, 7, 8, 9, 10, 11, 12] Brain A. Nixon proposed a framework for managing the performance requirements for information systems. The Performance Requirements Framework (PeRF) integrates and catalogues various types of knowledge of performance and information systems. This framework includes the concepts and principles of performance. These can be used to build the performance into information systems. The frame work proposed for information systems to manage performance requirements is developer oriented approach rather than user oriented. This is the one drawback because user is the most important stakeholder for any system.

In [13] an approach is proposed for analyzing the software requirements specifications for performance. In this approach the scenarios of the use cases are executed with data operations and responsibilities. Use case maps are used in specification and then analyzed for performance. The model presented in [13] is validated only indirectly. As the model is intended to for insight into potential performance fit falls it should be validate for structure, completeness and parameter accuracy. These are the drawbacks from the proposed approach. In [14] an approach is proposed. This is named as "PeGU". This approach is used for modeling the performance by using Goal oriented Requirement Language (GRL) and Use Case Map (UCM). This is used to evaluate the architectures qualitatively with each technology at early stages. Performance requirements are treated as soft goals and goals of GRL by PeGU. Use case maps are used for representation of scenarios.

**Revised Version Manuscript Received on 10 September, 2019.**

GopichandMerugu, Professor& Head, Department of Information Technology, Vardhaman College of Engineering, Hyderabad, Telangana, India.

(Email: Gopi\_merugu@yahoo.com)

The work presented in [14] uses goal and scenario oriented approach for addressing and refining the performance requirements. The performance requirements addressed are so abstract. They are not detailed descriptions. For example the performance requirements identified by using this approach are response time and space, which are represented as high level performance requirements. This may cause for difficulty in understanding the performance requirements. This approach cannot effectively support the performance requirements identification process. The work presented in [15] is describes about expressing the soft bounds of performance for software systems using stochastic probes. This approach uses regular expressions to specify stochastic probes. The work presented in [15] uses stochastic probes as means of measuring soft performance characteristics over software systems. In this work an analysis is made to identify the performance bounds and reliability bounds. Performance analysis is made on system specific operations. There is no approach proposed particularly which can be used at requirements level.

There is an approach [16] for gathering performance requirements which bases on three views like deployment view, operational workload view and persistent data view. Deployment view is used to acquire service-level agreements and the hardware and software environment constraints for the deployment. Operational workload view is used to acquire several types of workloads of the software system, the intensity and pattern of requests during the various workloads, the growth projections, intervals and expected responsiveness. Persistent data view is used to capture the growth of data over a time period and the current data size. The main attributes from technological and business point can be captured by these three views. But these three views will not present all the performance requirements as many other views are there which still need to be considered. For example different stakeholders' views have to be considered during the performance requirements identification. Use case view has to be considered to know the number of use cases and corresponding constraints with respect to performance.

In [17] a new method was proposed to predict performance requirements of mobile devices' software tasks using system models describing the hardware and software. With the help of clustering algorithms and linear regression, behavioral models of software tasks are generated automatically. These models are used to project the runtime of representative parts of the software tasks. The runtime of representative execution parts is determined with instruction-accurate simulations which are not feasible for whole executions. The inputs for the projection task are a model of the hardware platform and input data parameters, especially the data size. In this way the method helps to seamlessly integrate the performance analysis process into the development process. The method presented in [20] is an automated method which models the performance requirements automatically based on the behavior models which are not part of performance analysis. This method was applied for only mobile devices not generic software systems. Moreover, this is not a generic approach for identification of performance requirements.

In [21] a Five layered approach is proposed for elicitation of performance requirements. As part of this approach some conceptual rules and metrics are proposed. Conceptual rules

are used in the identification process and metrics is used to check the completeness of identified performance requirements. By this approach all possible performance requirements can be elicited.

### III. PERFORMANCE REQUIREMENTS REFINEMENT& RESULTS

Sometimes, it may be difficult to specify performance requirements completely at the early stages of the development process. If the product is developed without having an appropriate performance requirements specification, definitely it leads to performance problems at the end of the development. To produce the product with acceptable performance for next version, the performance requirements have to be refined. For this purpose, a very few methods are proposed in the literature such as performance requirements improvement model [20,22 23], improving with field failures [24] and improving performance requirements with the "defect not a problem reports" [25]. In [24] author has proposed a framework called Performance Refinement and Evolution Model (PREM). This can be used to manage performance software development. Workload specifications and quantitative requirements are used to distinguish the levels in model. Performance properties are specified qualitatively in test cases and requirements at higher levels. This model is used in performance improvement in [24, 25].

In [24] a procedure is proposed for improvement of performance requirements in future product releases. The information obtained from field failure reports is used in this procedure and requirements specification. This procedure uses the UML Profile for Schedulability, Performance, and Time (UML-SPT) [26] to extract the information. The information includes the subject of the requirement, performance measure, computation resource, and workload. By using the information extracted new performance requirements can be constructed. PREM is used in specifying the new requirements.

The approach proposed in [24] for performance requirements improvement uses field failure reports from customers. Some rules are proposed for estimation of performance requirements in this approach. These rules are so abstract rules rather than detailed. Individual measures for refinement of performance requirements are not specified in the rules. The team size used to test the product is 15-20 and the members of the team all are professional testers. If the team size increases, then more number of chances will be there for uncovering the errors. If the product is tested by the common users, then the results obtained projects more of the user opinion about performance requirements rather than developers. These are the drawbacks from the proposed approach. The approach "defect not a problem" proposed in [25] to improve the performance requirements uses defect reports which are considered as "not a problem" reports of defects. Professional testers have submitted the maximum number of defect reports. Additional information was added by the development team, which is gained during the

investigation to the defect. "Defect not a problem" is designed based on the information obtained from four elements. These are workload, subject, environment and measure. These elements are used for only the work described in [25]. In some cases these elements may not sufficient, where other types of requirements has to be considered. This is one aspect that has to be studied. Another problem is, after some releases this not a problem defect report may become actual problem. There should be strong reasons to say the defect found is not a problem.

#### IV. CONCLUSION

The acceptance of any software product by the customer depends on how well we identify the Performance requirements and incorporates in the software. There is a great need for sophisticated methods and supporting tools in order to successfully carry out performance requirements engineering activities. In this paper several useful methods available for performance requirements identification and refinement are presented such as Five Layered approach for performance requirements identification, Seven Layered model for performance requirements refinement, PeGU, approach is used for modeling the performance by using Goal oriented Requirement Language (GRL) and Use Case Map (UCM), Performance Refinement and Evolution Model (PREM), and "defect not a problem".

#### REFERENCES

1. IEEE Std 830-1993 IEEE Recommended Practice for Software Requirements Specification-Description.
2. M.Gopichand, A.AnandaRao, K.Narender Reddy and J.Kiran Kumar, "An approach to requirements elicitation and analysis using goal" ICSTE 2010: 2<sup>nd</sup> IEEE international conference on Software Technology and Engineering, October 3-5, 2010 San Juan, Puerto Rico, pp.218-221.
3. Gopichand.Merugu, A.AnandaRao, "Four Layered Approach to Non-Functional Requirements Analysis" International Journal of Computer Science and Issues (IJCSI), ISSN: 1694-0814, Volume 8, Issue 6, pp 371-379, November 2011.
4. Edward E.Balkovich "Research towards a technology to support the specification of data processing system performance requirements", ICSE 76 proceedings of the second international conference on software engineering 1976, pp. 110-115.
5. Donald G.Krantz "Ada development system technical and performance requirements", TRI-Ada 90 Proceeding of the conference on TRI-Ada 1990, pp.581-595.
6. B.A. Nixon, "Implementation of Information System Design Specifications: A Performance Perspective", Proc. Third Int'l Workshop Database Programming Languages: Bulk Types and Persistent Data, Aug 1991, pp. 149-168.
7. Brain A.Nixon "Dealing with performance requirements during the development of information systems", proceeding of IEEE international symposium on requirements engineering 1993, pp. 42-49.
8. B.A. Nixon, "Representing and Using Performance Requirements during the Development of Information Systems", Proc. Fourth Int'l Conf. Extending Database Technology, EDBT '94, March 1994, pp. 187-200.
9. B.A.Nixon, "Performance Requirements for Information Systems", doctoral thesis, Dept. of Computer Science, Univ. of Toronto, 1997.
10. B.A.Nixon, "An Approach to Dealing with Performance Requirements during Information System Development",

- Proc. Doctoral Consortium, Third IEEE International Symposium on Requirements Eng. RE '97, January 1997, pp. 73-88.
11. Brain A.Nixon "Managing Performance Requirements for Information Systems", WOSP98 Proceeding of the first international workshop on software and performance 1998 pp. 131-144.
12. Brain A.Nixon "Management of performance requirements for information systems", IEEE transactions on Software Engineering vol.26, No.12 December 2000, pp. 1122-1146.
13. ZhimingCai and Eric Yu "Addressing performance requirements using goal and scenario-oriented approach", Advanced Information Systems Engineering, 14<sup>th</sup> International conference, CAiSE2002, Toronto Canada ,May 2002, pp.706-710.
14. D.B. Petriu, C.M. Woodside, "Analysing Software Requirements Specifications for Performance", Proceedings of 3rd International Workshop on Software and Performance, Rome, 2002, pp.1-9.
15. Ashok Argent-Katwala, Jeremy T.Bradely, Nicholas J.Dingle "Expressing performance requirements using regular expressions to specify stochastic probes over process algebra models", ACM SIGSOFT Software Engineering Lecture Notes vol.29, Issue.1, January 2004, pp.49-58.
16. [Manish Pande](#) and [Rajeshwari Ganesan](#), "Gathering Performance Requirements", CMG, June, 2005.
17. Chih-Wei Ho, Michael J. Johnson, Laurie Williams, and E. Michael Maximilien, "On Agile Performance Requirements Specification and Testing", In AGILE '06, Proceedings of the conference on AGILE 2006, pages 47-52. IEEE Computer Society, 2006.
18. Simon.S, Patrick.P, Lukas.P, Peter.M "Automatic estimation of performance requirements for software tasks of mobile devices", ICPE 11, proceeding of the 2<sup>nd</sup> ACM/SPEC international conference on performance engineering, 2011 pp. 347-358.
19. Andre B. Bondi, "Best Practices for Writing and Managing Performance requirements: A Tutorial", ICPE12 , International Conference on Performance Engineering 2012, pp. 1-7.
20. Ho, C.-W. and L. Williams, "Managing Software Performance Engineering Activities with Performance Refinement and Evolution Model (PREM)", Department of Computer Science, North Carolina State University Technical Report No. TR-2006-28, September 2006.
21. Gopichand.Merugu, A.AnandaRao "Five Layered model for identification of software performance requirements" International Journal of Software Engineering and Applications(IJSEA), ISSN: 0975-9018, Volume 3, No.5, pp 47-61, September 2012
22. Ho, C.-W. and L. Williams, "Deriving Performance Requirements and Test Cases with the Performance Refinement and Evolution Model (PREM)", Department of Computer Science, North Carolina State University Technical Report No. TR-2006-30, Nov 2006.
23. Ho, C.-W. and L. Williams, "Developing Software Performance with the Performance Refinement and Evolution Model", in Proceedings of the 6th International Workshop on Software and Performance, Buenos Aires, Argentina, Feb 2007, pp. 133-136.
24. Ho, C.-W., L. Williams, and A. I. Antón, "Improving Performance Requirements Specifications from Field Failure Reports", in Proceedings of the 15th International Requirements Engineering Conference, New Delhi, India, Oct 2007, pp. 79-88.

25. Ho, C.-W., L. Williams, and B. Robinson, "Examining the Relationships between Performance Requirements and "Not a Problem" Defect Reports", 16<sup>th</sup> IEEE International conference on Requirements Engineering, Barcelona, Spain, Sep 2008, pp. 135-144.
26. OMG, "UML Profile for Schedulability Performance and Time", Version 1.1, 2005

### V. AUTHORS PROFILE



GopiChand.Merugu received Ph.D. in Computer Science & Engineering from JNTUA, Anantapur, India and he received his M.Tech. in Software Engineering from the same university. He received B.Tech. degree in Information Science & Technology from Nagarjuna University, India. He is Professor & Head, department of

Information Technology Vardhaman College of Engineering, Hyderabad. He is a member of IEEE, ACM, IAENG, CSI and ISTE.