International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Modeling The Framework for Evaluating the Non-Functional Requirements Affecting Application Efficacy in Cloud Computing Domain

Brijesh Pandey, S.S.Soam, D.K.Yadav

Abstract: ***Predicting application performance concerning the parameters such as processor, network, memory, and disk needs to be analyzed and the model is to be built up for various non-functional requirements in cloud computing. The NFR include availability, portability, security, reliability, etc. Cloud computing provides a way for provisioning of resources in one of the three ways. The first could be cloud infrastructure, the second could be provider infrastructure or the third could be a combination of both. Specifying NFR and ascertaining the required resources requires a lot of knowledge and therefore the various models are built up in this context. To exploit the cloud computing domain software engineer's needs to quantify various requirements of the application and based on that framework needs to be designed and developed for controlling cloud services at the multiple levels of abstraction. In this paper, we will analyze models evaluating various NFR affecting the efficacy of application in a cloud computing domain. The model could be both working for static as well as dynamic implementation.***

Keyword: ***NFR--Non-Functional Requirement, MTTF, MTTR***

I. INTRODUCTION

Requirement engineering is the method or process to discover, properly document and manage the requirements for a computer-based system that in itself is complete, relevant and follows consistency [1]. It acts as a foundation for any type of system or application software. The Requirement elicitation, analysis, specification, and verification are the four building blocks of the requirement engineering process. Although negotiation, documentation is also an integral part of the requirement engineering process. At project inception, some questions are asked from the customer to set up an essential comprehension of the issue and the nature of the solution that is desired is analyzed.

Requirement elicitation is the phase where the customer is asked about the objectives of the system that is to be accomplished on a per-day basis. The data acquired from the client during inception and elicitation is extended and refined during elaboration.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Brijesh Pandey*, M.Tech, Motilal Nehru National Institute of Technology Allahabad (up) India.

Dr. Sudhir Singh Soam, PhD, Dr. A.P.J. Abdul Kalam Technical University, Lucknow, (Uttar Pradesh) India.

Dr. D. K. Yadav, PhD, Indian Institutes of Technology, Allahabad (Uttar Pradesh) India.

This phase emphasizes on building up a refined specialized model of software features, functions, and imperatives.

Different users of the software may process conflicting requirements. In this manner, the Requirement engineer must accommodate these contentions through a procedure of negotiation. The specification is the rearmost task delivered from the side of requirement engineer. It fills in as the establishment for ensuing software engineering activities. Requirement validation examines the determination to guarantee that all product necessities have been expressed unambiguously.

NFR manage the qualities of the framework that can't be communicated as functions. These requirements address aspects concerning maintainability, portability, usability, concurrent users, timing, throughput, etc. It might likewise incorporate unwavering quality issues, the precision of results, human PC interface issues and imperatives on the framework execution. The non-functional requirements are a non-negotiable obligation that must be supported by the system. IEEE 870 standard lists four types of NFR: EI requirements, performance requirements, limitation, and software system attributes. NFR can be abstract since they can be seen, deciphered and assessed contrastingly by various individuals. The Non-functional necessity can likewise be relative since the understanding and significance of it might shift contingent on the framework being figured it out. Cloud computing implies controlling, planning and getting to the application on the web. Two models make cloud computing progressively plausible and

open to end clients. The first one is the deployment model which determines the kind of access to the cloud. The access can be of four kinds: Public,private, hybrid and community. The public cloud permits access to services and systems publically which could lead to compromise in security. The private cloud permits systems and administrations to be open inside an association. The community cloud permits access to services and system only to a certain group of the organization. The hybrid cloud is amalgamation of private and public clouds. Essential exercises are performed utilizing private cloud and non-fundamental activities are performed utilizing open cloud.The second one is the service model and this can be categorized into three primitive service models i.e. Software as a service (IaaS), Platform as a service (PaaS) and Infrastructure as a service (SaaS).

A model captures aspects important for some applications and omitting the rest. About software development, the model can be textual, graphical, mathematical or program code based. For modeling the different nonfunctional requirements in cloud computing, we must evaluate the various issues and those can be said to be the major inputs required. Like for portability, we need to look for OS images, hardware, etc. For the reliability, the MTTF (Mean time to fail) and MTTR (Mean time to repair) are evaluated. It also requires resources for application backup. For availability uptime percentage, user location, load balance and contract time are the inputs. For efficiency throughput, workload and response time could be the features that need to be asked by the user. Likewise, for fault tolerance, cost estimation, and security the other features or factors could be the major inputs of the system.

II. FACETS OF SURVEY

In this section of the paper, we shall discuss our aspect of surveying the vast and early research of requirement engineering to the importance of NFR in the new scenario. We shall discuss the emergence of cloud computing and its role in today's generation. Then we shall go for the NFR usefullness in cloud computing and perform the survey of related works. We will discuss various NFR models in cloud computing that are built up for predicting and analyzing the application performance. We will discuss various cloud monitoring services and application resource estimation techniques.

A. Requirement Engineering

An overview of 8000 tasks attempted by 350 US organizations uncovered that 33% of the activities were rarely finished and 50% only get partially completed with incomplete functionalities, real cost overwhelms, and huge deferrals [5]. At the point when gotten some information about the reasons for such disappointment official chiefs distinguished poor prerequisites as the real wellspring of issues (about portion of the reactions) - all the more explicitly, the absence of client involvement(13%), necessities deficiency (12%), evolving necessities (11%), unreasonable desires (6%), and hazy destinations (5%). On the European side, an ongoing review of more than 3800 associations in 17 nations likewise inferred that the greater part of the apparent programming issues are in the territory of necessities determination (>50%) and prerequisites the executives (half) [6].

The Quality of the product is unexpected to necessity elicitation, prerequisite investigation, and necessity of the executives [9]. The specialists have ejected a technique 'Abstraction based requirement management (AbstRM) to overcome elicitation's difficulties in necessity building. The data ends up opposing and inconsistent as it has been obtained from various sources. Also, manual prerequisite examination, the revelation of significant procedures and location of deliberations from situations have been preeminent difficulties for necessity elicitor [10]. The specialists ejected a device known as AbstFinder [11] which records significant terms known as 'abstraction identifiers.' The idea is utilized to group identifiers arrays into various classes, for example, operators, elements, activities, objectives. The Explanation for every abstraction identifier is recovered from situations.

Table -I: Views on requirement engineering

Sl.No	Researchers	Views on Requirement Engineering
1.	Bell and Thayer [2]	They saw that insufficient, conflicting, inadequate or questionable prerequisites are various and critically affect the nature of coming about programming. They presumed that the prerequisite for a framework doesn't emerge normally; rather they should be built and have proceeded with audit and amendment.
2.	Boehm [3]	He assessed that the late revision of prerequisites blunders could cost up to 200 fold the amount of as amendment during such necessity building.
3.	Brooks [4]	He expressed that the main task of developing a software in whole is to realize what is to be build..Therefore, the software developer must extract and refine its prerequisites to a significant capacity.
4.	Ross and Schuman [7]	They expressed that prerequisites' definition is a cautious evaluation of the necessities that a framework is to satisfy. It must state why a framework is required, in light of current or anticipated conditions, which might be inward tasks or an outside market. It must state the features which will serve in various context.
5.	Goldin and Finkelstein's [8]	They expressed that it has been an incredible test to understand the partner's needs and deal with the startling development of prerequisites.

- *Modeling the Requirement:* In all the effort we did to study requirement engineering it was understood that modeling is the kernel part of requirement engineering. The multiple intertwined activities of requirement engineering can be performed only with the help of modeling. The models serve as the basis for a common interface to such activities. They also provide the platform for documentation and evolution.
- *Functional Requirement:* Functional Requirement determine a capacity that a software framework or part of it should have the option to perform. It very well may be reported in different ways. The commonly used ones being are composed depictions in archives and use cases. During the development of use cases, the set of actors is defined. The actors are nothing but the set of users who utilize the framework inside the setting of functions that needs to be delineated. They speak with the framework and are outside to the framework itself. Each use case must define the set of primary and secondary actors identify their goals, task or function to be performed by them, the information that the system will acquire, produce or change, the precondition that must exist before the scenario begins or the exception that can be dealt with dealing with the description of scenario. The elements of the analysis model can be scenario-based, class-based, behavioral or flow-oriented.

In the scenario-based, the framework is depicted from the client perspective by utilizing activity and use cases diagrams. In class-based , the realm classes of the objects exploited by the actors are identified. They use class diagrams to depict the attributes of the classes and their interaction. The behavioral element uses a state diagram to depict the state of the system and the event that causes alteration of the event. For flow-oriented, DFD_s are used to manifest the input data that approach towards the system, the functions that are used to do the transformation and the output data that is generated. [12], [13], [14], [15], [16] and [17] have contributed much in this field.

- *Non Functional Requirement:* In the context of software system engineering IEEE defines NFR as the software requirement which not only explains what is to be done but how it will be done. Lawrence Chung views NFR as "Informally stated, often contradictory, difficult to enforce during development and
  - **Table -II: Comparision of NFR Architecture**

SL.NO	RESEARCHERS	APPROACH	METHODOLOGY	CASE STUDY	NOTATION	TOOL	NF ATTRIBUTES
1.	Nelson et al [22]	N/A	Configuration model (CL)	An appointment system	N/A	No	Yes
2.	Nelson et al [23]	N/A	Mapping strategy	N/A	Block diagram	No	Yes
3.	Lopez and Heman [24]	Integrating NFRs with use-case	Scenario-based Approach	Stock behavior system	Use case diagram	No	No
4.	Barbara et al [25]	Experience-based	Experience based	Navigation System of rocket	Block diagram	No	No
5.	Barbara et al [26]	Experience based	N/A	N/A	Class diagram	No	No
6.	Lin Liu Eric Yu [27]	Scenario based Approach	Use Case Maps (UCM	Mobile telecom system	N/A	No	No
7.	Cortellesa et al [28]	XML-based Approach	Integrating NFRs & FRs	Set & counter application	N/A	No	No
8.	Jane Cleland-H Uang [29]	Goal-oriented	GCT model	A telephony system	Block diagram	No	Yes
9.	Lin Liao [30]	Survey-based	N/A	N/A	Goal graph	No	No

evaluate to the customer before delivery". There are many NFR which can be stated as:Security, portability, reliability,agility availability and etc. These many requirements can be classified under various classification schemes. Such can be requirements for performance,interface,operation, and life cycle, economical and political. The other classification is done considering the product, process and external interfaces. There are even some approaches to non-functional requirements such as process vs product approach, quantitative vs qualitative approach, etc. Thomas [18] stated that meeting the non-functional requirement is very critical but if elicited properly the challenge can be met. The network diagram stated by Golden and Finkelstein [19] traces the impact it has on other system requirements by changing or appending some of the other non-functional requirements. The use case diagrams are not employed to specify the constraints of non-functional requirements instead of the graphical notations given by Firesmith [20] and mathematical terms are used. Cortesi and Logozzo [21] stated that non-functional prerequisite can be approved by creating models or devices and applying the unique elucidation based static investigation of source program and choice of theoretical area. The objective arranged methodology gives necessity designers to comprehend the non-useful prerequisite and investigate them with maximum capacity.

B. Cloud Computing

The National Institute of Standards and Technology (NIST) says "*Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources for example networks, servers, storage, applications and services that can be rapidly provisioned and released with minimal management effort or service provider interaction*." NIST also categorizes cloud computing "*as a service*" of three offerings namely infrastructure, platforms and software. These services can be analyzed in the following ways:

- *Cloud IaaS (Infrastructure as a Service):*
  A Cloud Infrastructure comprises of a pool of exceptionally preoccupied and versatile framework gadgets existing in different supplier server farms associated over private web associations where a believed outsider is accused of upkeep and portion.
- *Cloud PaaS(Platform as a Service):*
  A Cloud stage executes a virtual occasion of a center arrangement of usefulness with basic highlights, for example, client sign up, security, revealing and so forth. It enables designers to expand over the case the customization for explicit needs and to also construct includes over the stage, for example, making of structures, information section gathering, report composing and so forth without composing the programs.
- *Cloud SaaS(Software as a Service):* Cloud programming is a product part of distributed computing. It is the most obvious part since it faces the end clients. It gives Application Programming Interface (API) which enables the engineer to build up a redid application. SaaS has demonstrated to be advantageous as far as execution, versatility, productivity and substantially more.

- *Cloud Monitoring Services:*
  - *Cloud Harmony [32]:* This service reports benchmarks for the public cloud providers that endorse comparisons related to performance, network, uptime, etc. To gather the measurements observing administrations is situated inside and outside of the cloud supplier. Some standard applications are likewise executed to process the equivalent.

- *Cloud Sleuth [33]:* This service provides a tool called cloud provider view. This device screens the apparent reaction time and the level of accessibility of cloud suppliers in various urban communities around the globe.
- *Cloud Status [34]:* This instrument is incorporated into the Hyperic and it gathers ongoing perception of cloud supplier measurements, for example, accessibility, reaction time, idleness and throughput.

- *Application Resources Estimation Techniques:* Stewart and shen [35] have proposed the model to estimate the performance of the online application. They have obtained a profile of each component, communication channels and overhead of remote invocation to give the input in a linear model which gives the workloads, estimates throughput and response time to forecast the finest placement strategy for components. Shimizu [36] also presents an approach to predict several resources that an application would require to be executed. This work presents a solution that acts independently of the platform where applications are deployed. They have proposed to make a profile of the application with a specific workload in different platforms taking estimates of computing, communication and storage parameters. These parameters are inputted into the linear mathematical model that allows estimation of resources, response and execution time for static workloads in different conditions of hardware. Wood [37] also estimates the resources in an application deployed in a virtualized environment. This has used a linear regression model to estimate the resources. Kundu [38] proposes a neural network model that iteratively increases the accuracy of the model.

III. NON-FUNCTIONAL REQUIREMENT CLOUD COMPUTING

Table-III: NFR models

SL.NO	RESEARCHERS	MODEL	MEASUREMENTS
1.	Stewart and shen [35]	Linear Model	Throughput, Response time
2.	Shimizu [36]	Linear Mathematical Model	Measures of computing, communication and storage parameters
3.	Wood [37]	Linear Regression Model	Resources in an application
4.	Kundu [38]	Neural Network Model	Accuracy of the model

NFR are used to take care of the issue for resource estimation in a cloud computing environment. They are used to classify from the administrator the constraints and performance required from the application. They can be used to identify the most suitable cloud provider. The non-functional requirement can be prioritized to select the cloud provider. Hence we can say that cloud providers cannot be only selected based on price but the priority for the need in an application.

A. General nfr in cloud computing

Some non-functional requirement needs to be discussed to propose the model in the future perspective and they are as follows:

- *Portability:* It is the effectiveness of the application to be executed on various stages.
- *Reliability*: It is related to the resources required to run an application backup. Although Mean time to failure (MTTF) and Mean time to repair (MTTR) should be asked from the administrator.
- *Agility:* It is the ability to integrate new capabilities.
- *Availability*: The system can respond to the user request. The administrator must be asked about the minimum percentage of the uptime required.
- Fault tolerance: It is the property that specifies how the system can withstand errors. The errors can be software or hardware-based both.
- Security: The security of the system is mostly of utmost importance and that should not be compromised.

B. Tools to compare and estimate resources in cloud providers

Let us discuss some ideas pertaining to the task of resource estimation and finding the better among the cloud provider for a particular application.

Table-IV: Tools for resource estimation

SL.NO	RESEARCHERS	WORK DONE
1.	DEVA [39]	A module was made to be included a cloud server farm toolbox administrator, for example, Open Nebula [31], Eucalyptus [32] or any internal resource manager. This module can interpret client asset necessities in cloud physical assets that fulfilled chiefly organize prerequisite given by the client. In this work, it was expected from the user to give CPU power, the quantity of RAM and Bandwidth.
2.	DEVA [40]	It was proposed to add a module that receives non-functional requirement which is later translated into a specific requirement. These non-useful prerequisites are additionally utilized for application observing and cloud deployment to keep fulfilling these non-functional requirements.
3.	Cloud Prophet [43] [44].	It performed the estimation in two phases i.e. tracing and replaying. The primary stage is utilized to gather information identified with the application outstanding task at hand in a situation outside the cloud supplier. In the second phase, an operator is set up with an application remaining task at hand to reenact the execution of the application in a cloud supplier and to evaluate the cloud execution.
4.	Li [45]	CloudCmp which looks at changed cloud suppliers by utilizing an instrument to perform orderly seat checking. This instrument assesses the administrations of flexible registering, constant stockpiling, intracloud and wide territory organizing. For playing out this sort of administration various lattices are picked and estimated by playing out similar undertakings in various cloud suppliers.
5.	Li [46]	It incorporated into his instrument the yearning application that permits the correlation of expense and execution of a web application that is sent in various cloud suppliers.
6.	SMICloud [47]	This was a product structure to rank cloud suppliers for a given application considering the administration measure files: responsibility, readiness, confirmation of administrations, cost, execution, security, protection, and ease of use. The proposition was to allocate distinctive key Performance Indicators (KPI) to assess these records in various cloud suppliers. These measures were utilized to think about, position and propose a cloud supplier. In the framework, the user could specify the attributes according to the application needs along with the weight to specify the importance of each attribute. The framework had the module to keep information about the cloud provider functionality and SMI calculator which gives a value to each one of the KPI. The ranking system module of this framework uses this information to resolve Multiple Criteria Decision Making (MCMD) using an Analytic Hierarchy Process (AHP).
7.	Zhang [48] [49]	It bolsters framework executives in the troublesome assignment of perusing cloud supplier determination to discover the parameters to think about lastly decide the best choice at the best cost. To achieve this goal they have used a genetic algorithm along with the Analytic Hierarchy Process (AHP).

V. RESULTS AND DISCUSSION

In this section of the paper, we would like to propose our model for the framework which determines the priority of NFR according to the application needs in the cloud computing environment.According to the data collected from science direct the following five NFR holds priority in the list whether it be any type of application. The Various graphs depict the work done in the field of particular NFR. We can classify NFR from a product to process-based, qualitative to quantitative-based, etc. Therefore different scenarios can be made according to the application needs and the different cloud providers can be determined based on that priority.

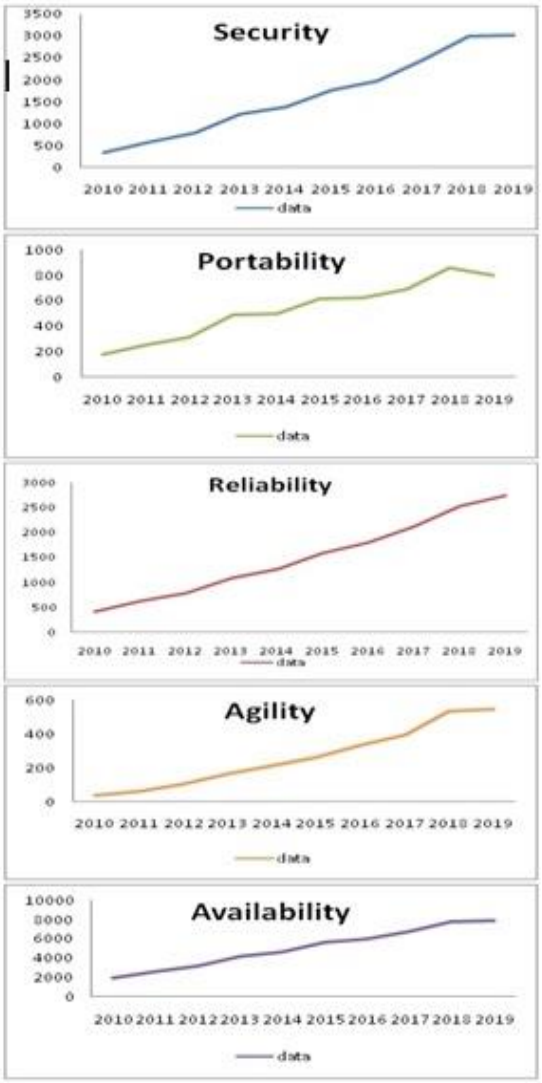

Fig.1. Statistical data from science direct

We are proposing a model in which the NFR priority can be set based on Binary codes. For any application, the two most prioritized NFR the Boolean code 0 and 1 can be assigned. For the next four prioritized NFR, the code could be 00, 01, 10 and 11. The process can go on as per the requirement of the application. Suppose if for the given application the sequence of priority set is {security, portability}, {reliability, agility, availability} … and so on then it can be depicted in the tabular form as follows:

Table-V: Prioritization of NFR

NFR	Priority	Binary Code
Security	Very High	0
Portability	High	1
Reliability	High	10
Agility	High	11
Availability	High	100

We propose to categorize the non-functional requirement into four quadrants of priority and that can be very high, high, low and very low. The higher the priority the lower is its binary code. In the static implementation where the priority of NFR will be fixed before the deployment of the cloud for the particular application the binary codes will remain the same throughout and hence the specification once written will not change. The specification of NFR will be written in the equation form and the same would be done for cloud providers. The third-party which is the cloud provider will be responsible for matching the equation and thereafter providing the service according to the NFR of the application.

In case of dynamic implementation where the NFR priority may vary from time to time and again the job of cloud provider will increase. The specification equations will change dynamically and hence the action has to be reciprocated. If the priority of NFR will vary in its set itself then no severe action needs to be taken. But if the priority of particular NFR changes the set then the service from that cloud needs to be discontinued. The equation will again be formed and matched with the cloud specification. The goal-oriented requirement engineering would be the inspiration for this work.

V. CONCLUSION

The work presented here is more of a survey starting from the late seventies to date in the field of requirement engineering to the significance of NFR in cloud computing. In the first section of the paper, we tried to elaborate on all the issues, aspects and even processes of requirement engineering, NFR, cloud computing, and formal methods. In the second section to specify our work, we started from the related work done in the field of requirement engineering to its modeling and then to NFR. In the same section, we discussed cloud monitoring services and application resource estimation techniques. In the third section, we have confined our research to the only Non-functional requirement in cloud computing. We have discussed some general NFR holding importance in a cloud computing environment. Then the tools to compare and estimate resources in cloud providers have been discussed. In the fourth section, we have proposed our model of specifying and verifying the NFR using binary codes in the equations for static as well as dynamic implementation.

The NFR is the main objective of intensify the performance of an application in a cloud computing environment in terms of processor, network, and memory. Therefore in the future perspective, we will take our model to its next level and present the deliverable which is more impactful to NFR.

REFERENCES

1. Requirements Engineering: A Good Practice Guide Ian Somerville, Pete Sawyer.
2. T.E. Bell and T.A. Thayer, "Software Requirements: Are They Really a Problem?" *Proc. ICSE-2: 2nd International Conference on Software Engineering*, San Francisco, 1976, 61-68.
3. B.W. Boehm, Software Engineering Economics. Prentice-Hall, 1981.
4. F.P. Brooks "No Silver Bullet: Essence and Accidents of Software Engineering". IEEE Computer, Vol. 20 No. 4, April 1987, pp. 10-19.
5. The Standish Group, "Software Chaos", http://www.standishgroup.com/chaos.html.
6. European Software Institute, "European User Survey Analysis", Report USV_EUR 2.1, ESPITI Project, January 1996.
7. D.T. Ross and K.E. Schoman, "Structured Analysis for Requirements Definition", IEEE Transactions on software engineering, Vol. 3, No. 1, 1977, 6-15.
8. L.Goldin and A.Finkelstein. "Abstraction-based requirements management." InProceedings of the international workshop on the Role of abstraction in software engineering. Shanghai, China, 2006.
9. A.V.Lamsweerde. "Requirements engineering in the year 00: a research perspective." In Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland.2000.
10. Endava: White paper on Requirements gathering and analysis, pp.7-10, 2007.
11. L.Goldin and D.Berry. "AbstFinder, A Prototype Natural Language Text Abstraction Finder for Use in Requirement Elicitation." In IEEE Requirements Engineering, 1994.
12. L.Goldin and A.Finkelstein. "Abstraction-based requirements management." In Proceedings of the international workshop on the Role of abstraction in software engineering. Shanghai, China, 2006.
13. D.G.Firesmith. "Modern Requirements Specification." Journal of Object Technology, 2(2):53-64, March-April 2003.
14. F.S. Juan. "Taxonomy of verification and validation of software requirement and Specifications.
15. Y.Chen, S.Chen, H.Sum, and S.C.Lin. "Functional requirements of metadata system: from user needs perspective." In Proceedings of the international conference on Dublin Core and metadata applications: supporting communities of discourse and practice---metadata research &applications. Seattle, Washington 2003.
16. B.Thomas. "Meeting the challenges of Requirement Engineering." News at Software Engineering Institute. 2009.
17. C.Alves and A.Finkelstein. "Challenges in COTS Decision-Making: A Goal-Driven Requirements Engineering Perspective."In Proceedings of the 14th international conference on software engineering and knowledge engineering. Ischia, Italy, 2002.
18. B.Thomas. "Meeting the challenges of Requirement Engineering." News at Software Engineering Institute. 2009. Website: http://www.sei.cmu.edu/library/abstracts/news-atsei/spotlightmar99pdf.cfm.
19. L.Goldin and A.Finkelstein. "Abstraction-based requirements management." In Proceedings of the international workshop on Role of abstraction in software engineering. Shanghai, China, 2006.
20. D.G.Firesmith. "Modern Requirements Specification." Journal of Object Technology, 2(2):53-64, March-April 2003.
21. A.Cortesi and F.Logozzo. "Abstract Interpretation-Based Verification of Non-functional requirements." Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 3454: 54-59, 2005.
22. Rosa, N. S., Justo, G. R., & Cunha, P. R. (2000). Incorporating non-functional requirements into software architectures. In Parallel and Distributed Processing (pp. 1009-1018). Springer Berlin Heidelberg.
23. Rosa, N. S., Justo, G. R., & Cunha, P. R. (2001, March). A framework for building non-functional software architectures. In Proceedings of the 2001 ACM symposium on Applied computing (pp.141-147). ACM.
24. López, C., & Astudillo, H. (2005). Use the case-and scenario-based approach to represent nfrs and architectural policies. In Proceedings of 2nd International Workshop on Use Case Modeling(WUsCaM-2005), Use Cases in Model-Driven Software Engineering Held in conjunction with Models.
25. B. Paech, et,al., (2002), Functional requirements, non-functional requirements and architecturespecification cannot be separated – A position paper".
26. Paech, B., von Knethen, A., Dörr, J., Bayer, J., Kerkow, D., Kolb, R., & München, T. U. (2003, May).
27. An Experience-Based Approach for Integrating Architecture and Requirements Engineering. In STRAW(pp. 142-149).
28. Liu, L., & Yu, E. (2001, May). From requirements to architectural design-using goals and scenarios. In ICSE-2001 Workshop: From Software Requirements to Architectures (STRAW 2001) May (pp. 22-30).Workshop (STRAW 2001), Toronto, Canada.
29. Cortellessa, V., Di Marco, A., Inverardi, P., Mancinelli, F., & Pelliccione, P. (2005). A framework for the integration of functional and non-functional analysis of software architectures. Electronic Notes in Theoretical Computer Science, 116, 31-44.
30. Cleland-Huang, J. (2005, November). Toward improved traceability of non-functional requirements. In Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering (pp. 14-19). ACM.
31. Liao, L. (2002). From Requirements to Architecture: The State of the Art in Software ArchitectureDesign. Department of Computer Science and Engineering, University of Washington,1-13.
32. CloudHarmony. CloudHarmony. http://cloudharmony.com/.
33. CloudSleuth. CloudSleuth. https://cloudsleuth.net/global-provider-view.
34. Hyperic. CloudStatus. http://www.hyperic.com/products/cloud-status-monitoring.
35. Christopher Stewart and Kai Shen. Performance modeling and system management for multi-component online services. In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2, pages 71–84. USENIXAssociation, 2005.
36. Shuichi Shimizu, Raju Rangaswami, Hector A Duran-Limon, and Manuel Corona-Perez. Platform-independent modeling and prediction of application resource usage characteristics. Journal of Systems and Software, 82(12):2117–2127, 2009.
37. Timothy Wood, Ludmila Cherkasova, Kivanc Ozonat, and Prashant Shenoy. Profiling and modeling resource usage of virtualized applications. In Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, Middleware '08, pages 366–387, New York, NY, USA, 2008. Springer-Verlag New York, Inc.
38. Sajib Kundu, Raju Rangaswami, Kaushik Dutta, and Ming Zhao. Application performance modeling in a virtualized environment. In High- Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on, pages 1–10. IEEE, 2010.
39. David Villegas and Seyed Masoud Sadjadi. DEVA: distributed ensembles of virtual appliances in the cloud. In Euro-Par 2011 Parallel Processing, pages 467–478. Springer, 2011.
40. David Villegas and Seyed Masoud Sadjadi. Mapping non-functional requirements to cloud applications. In International Conference on Software Engineering and KnowledgeEngineering, pages 527–532, 2011.
41. OpenNebula. OpenNebula. http://www.opennebula.org/.
42. Eucalyptus. Eucalyptus. http://www.eucalyptus.com//.
43. Ang Li, Xuanran Zong, Srikanth Kandula, Xiaowei Yang, and Ming Zhang. Cloud-Prophet: towards application performance prediction in the cloud. In ACM SIGCOMM Computer Communication Review, volume 41, pages 426–427. ACM, 2011.
44. Ang Li, Xuanran Zong, Srikanth Kandula, Xiaowei Yang, and Ming Zhang. CloudProphet: predicting web application performance in the cloud.http://www.cs.duke.edu/ angl/papers/cloudprophettr.pdf, 2011
45. Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. Cloudcmp: comparing public cloud providers. In Proceedings of the 10th ACM SIGCOMM conference on nternet measurement, pages 1–14. ACM, 2010.
46. Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. CloudCmp: shopping for a cloud made easy. USENIX HotCloud, 2010.
47. Saurabh Kumar Garg, Steven Versteeg, and Raj Kumar Buyya. SMICloud: A framework for comparing and ranking cloud services. In Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on, pages 210–218. IEEE, 2011.

48. Miranda Zhang, Rajiv Ranjan, Surya Nepal, Michael Menzel, and Armin Haller. A declarative recommender system for cloud infrastructure services selection. In Economics of Grids, Clouds, Systems, and Services pages 102–113. Springer, 2012.
49. Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and Research challenges. Journal of Internet Services and Applications, 1(1):7–18, 2010.

AUTHORS PROFILE

Brijesh Pandey has completed his M.Tech from MNNIT Allahabad. He is pursuing PhD from Dr. APJ AKTU Lucknow. He has published more than 10 research papers in various International Journals. He has been guided 5 M.Tech students.

Dr. Sudhir Singh Soam has received his PhD from Dr. APJ AKTU Lucknow. He is currently working in IET Lucknow. He has published many research papers and journals and delivered invite talks in reputed conferences. He is associated as a expert member of various universities. He is member of many regulatory bodies and panel member of RDC committees.

Dr. D. K. Yadav has done PhD from IIT Bombay. He is currently working in MNNIT Allahabad. He has published 25 international conference and journal papers. He has delivered many expert talks, invited talks and has been member of selection committee. He attended many short term courses and conferences. He has supervised 46 M.Tech students.