Analysis and Comparison of Web Development Platforms Based on Software Quality Attributes in Network Management System

Muhammad Nadeem, Shahnawaz Qureshi, Syed Muhammad Zeeshan Iqbal, Naveeda Bibi, and Lars Lundberg

***Abstract*—Network Management is an emerging field and progressively more popular in telecommunication industry. Software platform has become an essential part in development of Network Management System (NMS). Modern NMS must provide high performance in terms of response time, and reduced maintenance cost. In this paper, we investigate the significance of the software quality attributes for development of three NMS prototypes in PHP, JSP and Perl. Each of these prototypes was evaluated with respect to maintainability, performance and implementation cost. JSP based NMS is on the top having low maintainability effort of 79.4 with 95% confidence interval. The prototype developed in PHP has high performance and it is ranked no 1. With the lowest implementation cost of 58 hours, prototype implementing using Perl is on the top. Based on the results, we conclude that there are tradeoffs when considering different quality attributes. The main conclusion is that the choice of software platform for NMS is dependent on the industrial preference of software quality attribute.**

***Index Terms*—Performance, implementation cost, maintainability, network management system, software platforms.**

I. Introduction

Network Management System (NMS) monitors a collection of wireless network systems. In communication industry, its importance is increasing due to an increase in data operation and network activities. Therefore, the development need for NMS has been more esteemed by business planners, investors, and higher management, etc. One of the challenges is to show the stored network monitoring data in user friendly GUI (Graphical User Interface) and timely manner which then helps the higher management in decision making.

Other challenge in its development is to use different software platforms for evaluation of software architecture quality attributes. Software architecture constrains the achievement of quality attributes such as performance, usability and maintainability of a system [1], [2]. In both industry and academia several components are used in software architecture that are generally recognized and has led to better control over the software design [3].

A software platform is a concrete model of an architecture that allows executing software with detailed hardware-software interaction based on quality attributes [4] [5]. An evaluation of software platform has a great impact on the quality attributes. The evaluation process iterates over design of platform and fulfils all the required quality attributes. Changed scenario is an important part of practicing architecture in order to gain more information about maintainability of the software system [6]-[8]. In general, software architecture can be defined as "a decomposition and structure of software components and how these system parts are interconnected to each other" [9]-[11]. The design method of software is an activity to identifying the sub-systems of a software system. A prototype is a basic model/structure of a software system [12].

A mixed telecom network management application using J2EE is developed in [13], which interoperates with existing support systems. A method for NMS performance testing by using commercial cloud computing infrastructure is proposed in [14] and evaluated through a case study(it testifies the method applicability and effectiveness). A web services based NMS framework is proposed in [15], which covers system-structure, an architecture and deployment method. The system was implemented using LAMP (Linux Apache MySQL PHP) platform. All three prototypes have the same functionality but the quality attributes such as performance, maintain-ability and implementation cost is different in each prototype with different popular software platforms [16].

In a smart grid, for data collection and monitoring purpose, SCADA (Supervisory Control and Data Acquisition) system is used. Its security is one of the key issues which are addressed in [17] and [18]. A smart grid distribution and communication network architecture is suggested in [19] by considering technical and business requirements as well as financial cost. A prototype of Software-Defined Networking (SDN) is developed in [20]. It communicate and cooperate with SDN controller in order to measure Quality of Service (QoS) based on collected statistics.

To the author's best knowledge, there is a research gap for network monitoring using GUI by using three different software platforms and considering the software quality attributes (i.e., Maintainability, Implementation Cost and

Manuscript received January 12, 2017; revised April 19, 2017.

Muhammad Nadeem and Lars Lundberg are with the School of Computer Science and Engineering Blekinge Institute of Technology 371 79 Karlskrona, Sweden (e-mail: nadeemntcpk@gmail.com, Lars.Lundberg@bth.se).

Shahnawaz Qureshi is with the Department of Computer Science, Prince of Songkla University, Hat Yai, Thailand (e-mail: shahnawazqureshi86@gmail.com).

Syed Muhammad Zeeshan Iqbal was with the Blekinge Institute of Technology, Karlskrona, Sweden (e-mail: mzeeshan01@gmail.com).

Naveeda Bibi is with the IT Department, of Abbottabad University of Science and Technology, Pakistan (e-mail: khan_jadun@yahoo.com).

doi: 10.18178/jacn.2017.5.1.234

Response Time). This gap is filled by developing three prototypes, analyzing tradeoffs among them based on the quality attributes.

We applied both theoretical and empirical research methods. During literature review, we have analyzed learning structure and workflow of prototypes, and then incorporated quality attributes by theoretical analysis. In the experiment part, three prototypes were built in PHP, JSP and Perl. Each of these prototypes's development was a challenging task. The prototypes were evaluated with respect to quality attributes: Maintainability using twenty five surveys from industrial experts, Implementation Cost in number of hours and Performance in terms of response time. Finally, a comparative analysis of prototypes was performed.

II. Design and Development of Prototypes

According to the J. Bosch [7], architecture design is the process of changing a set of requirements into software architecture which satisfies the requirements. The design method of software is an activity to identifying the sub-systems of a software system.

First prototype is developed by using PHP (Hypertext Pre-processor), Apache web server (XAMP) and MySQL. Second prototype is developed by using JSP (Java Server Pages), Tomcat v7.0 web server and MySQL. Third proto-type is based on Perl, MySQL and Apache web server.

The computer for all three proto-types was Intel core i3 processor 2.4 GHz and 2 GB RAM.

NMS (Network Management System) Prototype

Smart Grid Networks, a company which is located in Karlskrona, Sweden, provide advanced solutions for NMS and digital communication. It enables cost efficient operations for infrastructure network owners by providing fast and secure control over their network. It contributes to greater efficiency, monitoring and control of network solutions.

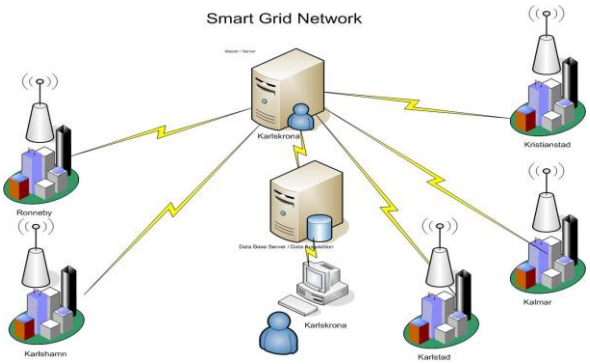

Fig. 1. High level diagram of smart grid network.

The company required a NMS (initially prototype) for monitoring and diagnostic of a radio network of 20 nodes (i.e., it represents an individual city) initially. The system manages connection to one or several different radio systems through both serial RS232 and TCP/IP communication. The system consists of three parts, (1) Data acquisition: probing network; acquiring, validating (according to protocol) and storing data (2) Data Manipulation: it stores the data of all nodes coming through data acquisition interface (3) Data Presentation: it shows the status, history, alarm, voltage level, etc., of nodes. Generally, the Fig. 1 shows the considered network.

This research may be helpful for the company: (1) to evaluate and select the suitable software platform for, (2) to estimate the size of NMS, its performance and usage pattern, (3) to recognize new areas for expansion, and (4) to set achievable targets for business growth.

III. Quantification of Software Quality Attribute

A. Implementation Cost (α)

The development process of prototype is divided into three phases: Design, Development, Implementation, and testing. The value 'α' is recorded in hours for all the phases.

B. Response Time (β)

The Performance in terms of response time (i.e., the waiting time spent in receiving the response from server) is measured. In each prototype the response time is measured (i.e., in millisecond) by using user request. In order to make the results statistical significant, we have executed the prototype several times (i.e., 30 executions). Average and standard deviation is taken as a reference to compare the performance and maintainability in all three prototypes.

C. Maintainability Effort (γ)

Scenario-based analysis method (change scenario) is used to evaluate the maintainability of the software prototype. Further, survey based research is used to distinguish the knowledge, attributes and behaviour of large group of people and support effective research results. The collection of survey results are taken from twenty-five industrial experts for the maintainability evaluation in a highly efficient way. The maintainability survey is based on following steps:

- Description of prototypes and scenarios
- Identify the target audience from both industry and academia
- Designing the survey form
- Approval of survey
- Statistical analysis of results

Maintainability for P_n as shown in equation 1.

$$P_n = \sum_{i=1}^{Ks} [Effort(P_n, i) * \Pr(i) \quad (1)$$

Equation1. Maintainability Effort Formula

where

P_n = Prototype number

K_s = Number of scenarios

i = Scenario number

Pr (i) = Probability of change scenario (i) occurrence Effort (P_n, i) = Effort to implement change scenario (i) in prototype P_n

IV. Results and Comparative Analysis

In this section, the detail of quality attribute for each prototype is given. The value of 'α' for following phases: design, development and testing are calculated. The value of

'β' shows the response time of the server and results were averaged over 30 executions. To calculate maintainability effort (γ), the survey results from 25 industrial experts were taken, the avg. value and standard deviation is given in Table 1 and in Fig. 2-Fig. 4 respectively.

TABLE I: VALUES FOR THE QUALITY ATTRIBUTE

Description		P1 (PHP)	P2 (JSP)	P3 (PERL)
Implementation Cost for each Phase (α)				
Design		10	8	6
Development		60	64	45
Testing		8	8	7
Total		78	80	58
Kilo Lines of code (KLOC)		50	1.20	1.35
Avg. Response time in milliseconds (β)				
View Network		6.74	13.80	6.90
Status of Network		7.63	14.03	8.24
Search Network		6.86	10.83	9.04
Maintainability effort (γ)				
AVG		80.92	79.40	81.40
Std Dev		19.35	16.26	19.33
Std. Error		3.870	3.252	3.947
95% Confidence interval for Mean	Lower Bound	72.93	72.69	73.25
	Upper Bound	88.91	86.11	89.55

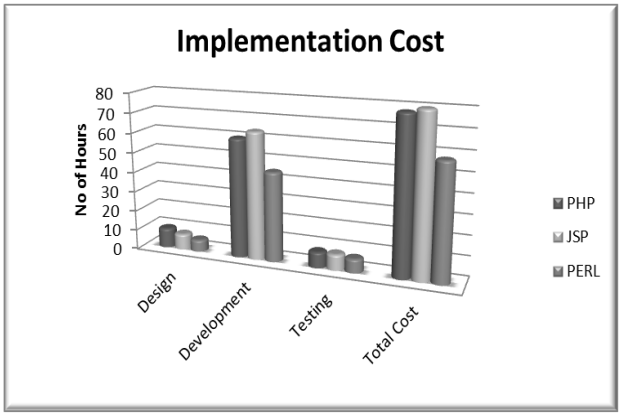

Fig. 2. Implementation cost for each phase (α).

Prototype developed in Perl has the highest maintainability effort when compared with other two prototypes. Among all three prototypes, it is observed that JSP prototype has lowest maintainability effort.

We have observed that the Perl prototype has the lowest implementation cost as compared to PHP and JSP prototype, whereas implementation cost of PHP & JSP prototype is approximately the same in hours but different in term of lines of code.

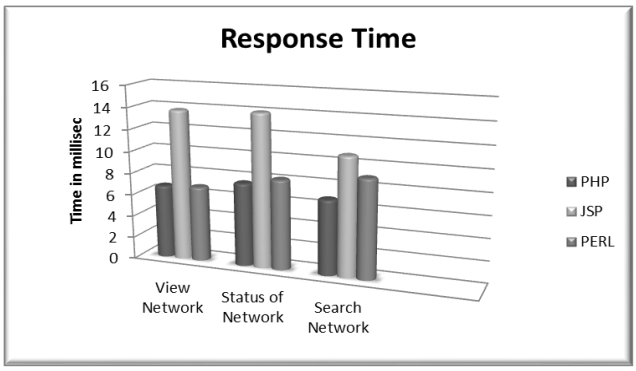

Fig. 3. Avg. Response time in milliseconds (β).

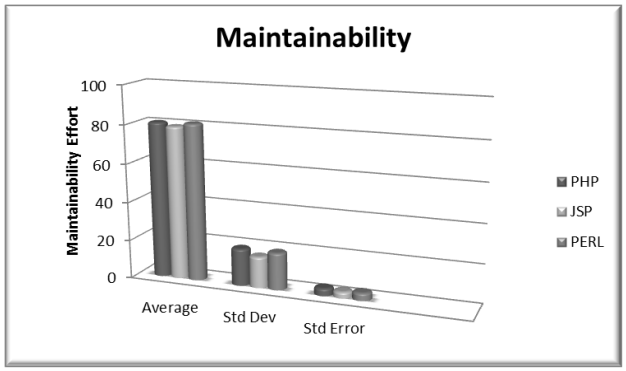

Fig. 4. Maintainability effort (γ).

PHP platform is the faster than other two prototypes with respect to all three major functionalities/test cases which were considered for evaluating the performance of each prototype. The lowest performance is on prototype 2 with JSP platform. Prototypes 3 with Perl platform has mediate performance with respect to other two prototypes. To put it in the other way, Prototype 3 is faster comparatively with prototype 2 but slower in contrast with (PHP) prototype 1.

V. SUMMARY AND CONCLUSION

This paper presents three software prototypes of Network Management System implemented using different software platforms (PHP, JSP and Perl). The purpose is to extract, represent and explore software quality attributes such as performance, maintainability and implementation cost. Further, use of extracted information is to identify challenges in software architecture and prototypes.

The significance level from low, average and high was assigned to each quality attribute. The first prototype (PHP) has high performance, average maintainability and average implementation cost. The second prototype (JSP) has low performance, high maintainability and low implementation cost. The implementation cost of both prototypes first and second is approximately the same with an average significance level. Finally the third prototype (Perl) has significance level as average performance, low maintainability and high implementation cost. It is not always possible to acquire each quality attribute in a software system. Consequently, there is trade-off between quality attributes of prototypes to-wards software platform.

Based on the finding, we give following recommendations. If main concern is maintainability industry has to select the second prototype. If performance is main concern, the first

prototype is the best alternative and the main advantage with third prototype its low implementation cost.

VI. Future Work

In our research, we studied the notion of design for quality attributes. In addition to that auxiliary research is required to realize, how to select suitable quality attributes based on certain problem.

This research has a great potential and can extended for future work by using the same software platforms to measure the different quality attributes such as utilization, throughput, availability, portability, re-usability and marketability. Furthermore, similar study is proposed as future work for different application fields like banking and telecommunication with the same quality attributes that we have used.

Acknowledgment

We would also be thankful for technical guidance, support, and encouragement given by Andres Grahn (Chief Technical Officer) and Peter Engström (Design Engineer) from Smart Grid Network AB Sweden. We would also like to thank all industry and academia survey participants, who contributed to survey part of this research.

References

[1] B. PerOlof and J. Bosch, "Haemo dialysis software architecture design experiences," *International Proceedings of Software Engineering*, 1999.

[2] D. Andrzej, "Visual modeling for scientific software architecture design. A practical approach," *Computer Physics Communications*, vol. 183, no. 2, pp. 213-230, 2012.

[3] B. Jan, "Software architecture: The next step," *European Workshop on Software Architecture*, 2004.

[4] D. J. Sung, "Platform maintenance process for software quality assurance in product line," *International Proceedings of Computer Science and Software Engineering*, vol. 2, 2008.

[5] P. Katalin, "Efficient software development platforms for multimedia applications at different abstraction levels," *International Proceedings of Rapid System Prototyping*, 2007.

[6] K. Rick, "SAAM: A method for analyzing the properties of software architectures," *International Proceedings of the IEEE 16th Conference on Software Engineering*, 1994.

[7] B. Jan, "Design and use of software architectures: Adopting and evolving a product-line approach," *Pearson Education*, 2000.

[8] K. Rick, "Scenario-based analysis of software architecture," *IEEE Software*, vol. 13, no. 6, pp. 47-55, 1996.

[9] G. David and D. Perry, "Software architecture: Practice, potential, and pitfalls," *International Proceedings of Software Engineering*, 1994.

[10] L. Bass, P. Clements, and R. Kazman, "Software architecture in practice. 2nd addison-wesley," *Reading, MA*, 2003.

[11] B. Pierre, "The guide to the software engineering body of knowledge," *IEEE Software*, vol. 16, no. 6, 1999.

[12] O. Guadalupe, B. Bordbar, and J. Hernández, "Evaluating the use of AOP and MDA in web service development," *International Proceedings of Internet and Web Applications and Service*, 2008.

[13] X. Lu, "Construct heterogeneous telecommunication network management system based on J2EE," *International Proceedings of Power Electronics and Intelligent Transportation System*, vol. 1, 2009.

[14] G. Zohar and I. Zilbershtein, "Cloud-based performance testing of network management systems," *International Proceedings of Computer Aided Modeling and Design of Communication Links and Networks*, 2009.

[15] X. Wang, "Studies on network management system framework of campus network," *International Proceedings of Informatics in Control, Automation and Robotics*, 2010.

[16] N. Muhammad and M. Azharuddin, "Performance, maintainability and implementation cost for different software platforms in a network management system," M.S. thesis, Dept. Electrical Engineering, Blekinge Institute of Technology, Karlskrona, Sweden, 2012.

[17] Y. Abdulrahman and S. Rahman, "Smart grid networks: Promises and challenges," *Journal of Communications*, vol. 7, no. 6, pp. 409-417, 2012.

[18] M. Anthony and R. Ekl, "Security technology for smart grid networks," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 99-107, 2010.

[19] S. Sujie, X. Qiu, and L. Meng, "Smart gird distribution and consumption communication network architecture," *Journal of Communications*, vol. 8, no. 8, pp. 480-489, 2013.

[20] H. Tim, F. Glinka, and S. Gorlatch, "A northbound api for qos management in real-time interactive applications on software-defined networks," *Journal of Communications*, vol. 9, no. 8, 2014.

Muhammad Nadeem received the master degree in electrical engineering with emphasis on telecommunication from Blekinge Institute of Technology, Karlskrona, Sweden in 2012.

He joined national telecommunication corporation as an engineer O&M in 2013. He joined University of Sargodha (UCET), Pakistan as visiting lecturer in Electrical Department in 2015. He joined Hira College of Engineering Sargodha as a head of Department of Electrical Technology, Sargodha, Pakistan. His research interest includes network management, performance evaluation of network management systems, software engineering, software prototyping and software architecture

Shahnawaz Qureshi received the B.S in computer science from Hazara University Pakistan in 2008 and M.S from Blekinge Institute of Technology, Karlskrona, Sweden in 2012.

He is a Ph.D student in the Faculty of Science, Department of Computer Science at Prince of Songkla University Hat Yai Campus, Thailand. His current research includes brain computer interface, machine learning and artificial intelligence.

Syed Muhammad Zeeshan Iqbal received BCS in computer science from Punjab University, Lahore, Pakistan in 2003 and MCS from Punjab University, College of Information Technology in 2006. He received the M.S in computer science from Blekinge Institute of Technology, Karlskrona, Sweden in 2010. He is currently working as a private research consultant in Denmark. His current research includes parallel computing and computer networks.

Naveeda Bibi received the B.Sc and M.Sc in computer science from Hazara University Pakistan. She is a lecturer in Abbottabad University of Science and Technology, Pakistan. Her current research includes computer networks management system and software testing.

Lars Lundberg is a professor in computer systems engineering at the Department of Computer Science and Engineering at Blekinge Institute of Technology in Sweden. He has a M.Sc in computer science from Linköping University and a Ph.D. in computer engineering from Lund University. His research interests include parallel and cluster computing, real-time systems and software engineering. Lundberg's current work focuses on performance and availability aspects.