UDC: 055.332.1:004.3/.4
Professional paper

THE MANAGEMENT SYSTEMS CHALLENGES FOR PROCESSES AND COMPUTERS PERFORMANCE ANALYSIS

Nderim Zeqiri[1], Eip Rufati[2], Bestar Zeqiri[1], Denis Sinani[2]

1 University of Tetova, Faculty of Applied Sciences, n.n. 1200, Tetovo*
2 University of Tetova, Faculty of Natyral Sciences and Mathematics n.n. 1200, Tetovo

**Corresponding author e-mail: nderim.zeqiri@unite.edu.mk*

Abstract

In this paper, it is envisaged to make assessments regarding the aspect of various challenges that appear in computer equipment and the aspect of finding the relevant software. Also an integral part of this research is the process of challenges in creating a relevant structure with the interconnection of computer equipment. Another characteristic of this paper is the presentation of some formulas for software evaluation in distribution. The main goal is to create a sustainable and long-term system, as well as to assess the maintenance process that lasts as long as the life of the computer structure with certain practical destinations. Apparently, most publications agree with the fact that object-oriented testing is a challenging aspect of the software development process. Object-oriented technology is evolving and has not yet achieved sufficient development. Many articles have been written on the subject of object-based software development, especially in the field of testing. Also, in the paper are given some characteristics of the metric for determining the quality of the software, before the sale and after the sale, and here are found the respective values, which are important data for the maintenance process.

Keywords: Management, challenges, performance, total time execution.

1. Introduction

The whole research process is related to the aspect of finding the best performance, and determining the characteristic equipment for the system to be stable and long-term in use. For the realization of this research, computer equipment, database system and formulas for metrics-determining their user validity as key elements that are part of the job challenge in general are in support and analysis. Also, changing the system, which offers us the opportunity to evaluate metrics through characteristic forums, allows us to find the best solutions, in software evaluation, and evaluation of devices that are an integrated part of the system. The most important challenge is the preoccupation with finding the most adequate and long-term solution in use Object-oriented technology is evolving and has not yet achieved sufficient development[1][2]. Many articles have been written on the subject of object-based software development, especially in the field of testing. Apparently, most publications agree with the fact that object-oriented testing is a challenging aspect of the software development process. The main reason for this view stems from the fact that objects, code and heritage are inseparable. Despite these views, all publications agree on one aspect, that object orientation when successfully tested leaves behind an easier maintenance product compared to traditional software that was not object-oriented. Object-oriented software creates easier maintenance software; it also has an added benefit to traditional software development because in the final analysis, it will cost less by shortening the development time as well as reducing maintenance costs.

2. Performance and computer systems

To calculate the performance of a computer system, we first need to know the equipment of the system, and loading with certain tasks to assess which of the computers, performs the best processing speed and what are the factors [3][4]. To assess the state of the system, we need to create workgroups, and upload with tasks. For example: In the next table is given the considered computer systems.

Table.1-Computer system and average time for tasks and processing

Tasks	Computer1	Computer2	Computer3	Computer4	Computer5	Computer6	Total Task Time
T1	0.5	0.4	0.6	0.8	0.6	0.5	3.4
T2	0.4	0.5	0.3	0.2	0.6	0.5	2.5
T3	0.3	0.6	0.6	0.8	0.6	0.5	3.4
T4	0.2	0.7	0.6	0.8	0.6	0.5	3.4
T5	0.5	0.6	0.7	0.8	0.9	0.7	4.2
T6	0.7	0.4	0.6	0.8	0.6	0.5	3.6
Average Time	**0.43**	**0.53**	**0.57**	**0.70**	**0.65**	**0.53**	**3.42**

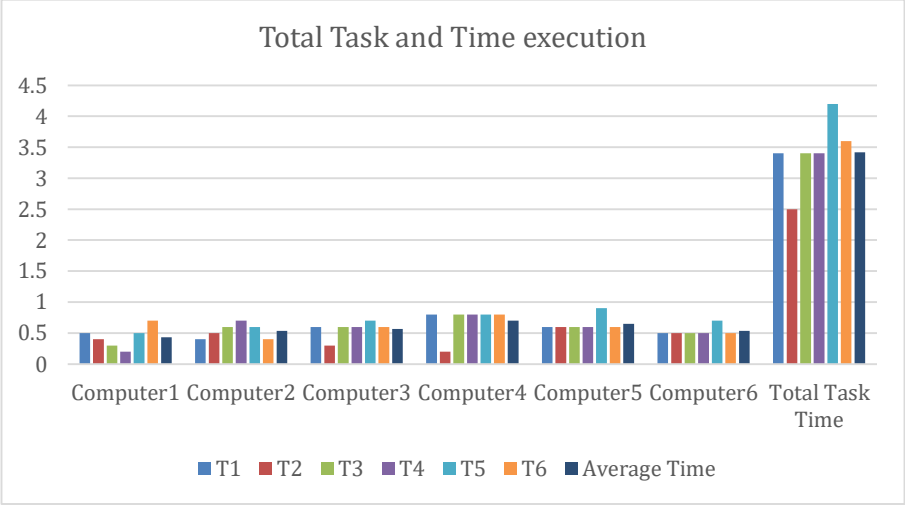

Figure 1. Graphical representation, Total Task and Time execution

Table 2. Performance of the computer system

Computer	Performance=1/Total Time
PC1	0.38
PC2	0.31
PC3	0.29
PC4	0.24
PC5	0.26
PC6	0.31

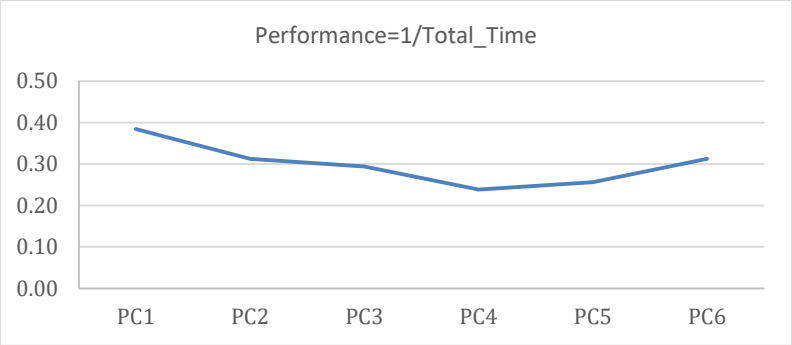

Figure 2. Graphical representation of the respective performance

The better performance has the computer PC1, and Low performance has the Computer PC4.

$$PC1 > PC2 = PC6 > PC3 > PC5 > PC4 \qquad (1)$$

If the conclusion is from lab analysis, this is really, and the next step is how to increase the performance for other computer systems [5][6]. The first attempt is to analyze, the type of computers, interconnection, computer network (if there any). Also, the finally context observation is to change all the resources and to install new system, which can support the charge/or uploads in the time of processing.

3 Clusters and Class Testing

Clusters are a collection of classes that are interconnected through the way it works. All of these clusters are based on system behavior. Cluster testing is cheaper than class testing. Library clusters flow as a result of implementation problems. Most cluster classes have already been tested. Whenever changes are made to pre-tested software, the cluster test needs to be run. Therefore, it tends to accumulate in the cluster. And in the end, they became so large that there would not be enough time to review the code. When this happens, the goal is possible because it was supposed to cost less; but since it is possible to detect errors in the final product, fixing those errors will be expensive. However, this problem can be eliminated. Elimination is done through classroom testing and combination testing of classrooms and clusters which can be automated. Cluster testing consists of three stages. Initial testing is documented. The next stage is to review the test plan. The final stage is the execution and recording of test results [12]. Within the cluster testing plan, testing is required. Test functions were to be marked by the object-oriented analysis stage and the design stage.

3.1 *Qualitative Metrics and processes*

There has always been a problem with how to measure the quality of a software product. The methods used to measure the quality of software in the traditional way have been done by creating a starting point which had to be created during the process of object orientation [7][8]. This has happened due to the unique features of object-oriented technology. Software metrics can be used to identify the location of resource allocation which became a very important component in decision-making. Managers and developers need this reliable information about the decisions they will make. System testing is a time consuming activity that needs to be identified so that scarce resources can be used where they are most needed.
In the next step is given some characteristics formula for quality and metrics (defect distribution)

$$Defect\ ditribution = \left(\frac{Total\ number\ of\ defects}{Functional\ area\ (s)}\right) * Status * Phase \qquad (1)$$

Running a defect distribution report will give testers a rundown on the location and severity of all flaws discovered during the course of the project's run.
In the next step is given the "Defect open and Close Rate" as integrated part for Software Quality Assurance.

$$Defect\ open\ and\ Close\ Rate = \frac{Defect\ found\ before\ delivery}{\left(+\frac{Defect\ found\ before\ delivery}{Defect\ found\ before\ delivery}\right)} * 100 \quad (2)$$

For example,

Table 3. Defect open and close rate

	Defect found before delivery	Defect found after delivery	(Defect found before delivery + Defect found after delivery)	Defect Open and Close Rate (multiplied by *100)
1	4	3	7	57%
2	5	2	7	71%
3	4	1	5	80%
4	3	2	5	60%
5	4	1	5	80%
6	4	3	7	57%
7	3	1	4	75%
8	2	1	3	67%
9	..			

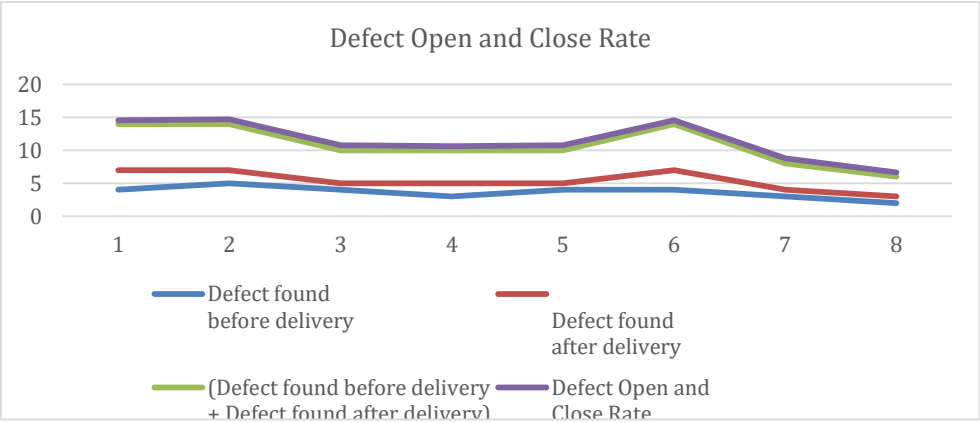

Figure 3. Graphical representations –Defect Open and Close Rate

In the metrics is given Execution Trend, these metrics identify which tests have been executed by a given member of the QA team as well as indicate trends related to the status of found defects.

$$Execution\ Trend = \left(\frac{Execution\ status}{Execution\ rate}\right) * \left(\frac{Functional\ area}{Iteration}\right) \quad (3)$$

Test execution reports can be created with information detailing these trends intact.

4 The part of the database management systems and interconnections

Creating a sustainable system for achieving software objectives also requires sustainability in terms of managing adequate measurement of software security parameters and system design. Part of the protection of the following network is realized in "Firewall". Also, the organization of the elements of this system is very important for the normal functioning, of the installation processes, and of the database management processes as central archiving [9][10][11]. Also, here we must emphasize the aspect of the performance requirement of computer equipment which we highlighted in the previous section.

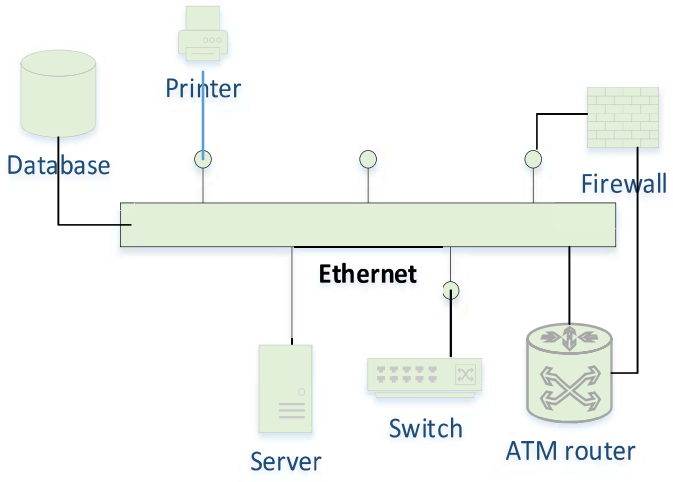

Figure 4. Database management systems and interconnections

For example, Figure 4, [3][8], shows the part of the block diagram of computer network operation. In this network, switch, router (and firewall), server etc., are deployed to interconnect across different networks. This segment shown above is an important part. So, this part of the network and the database are in function of access to elements of technological process, total resource management in the respective company.

5. Conclusion

We can conclude that finding good performance values is particular importance, because with this we create the functional security of various devices. Also, changing the system, which offers us the opportunity to evaluate metrics through characteristic forums, allows us to find the best solutions, in software evaluation, and evaluation of devices that are an integrated part of the system. The most important challenge is the preoccupation with finding the most adequate and long-term solution in use. This paper focus to use the computer systems, and respective software should be easy to use (user-friendly). In principle, it should be simple enough for the user to be able to use it with minimal problems. One of the advantages of object-oriented software is re-use. In general it can be said that object-oriented technology can improve the software design process as well as facilitate its modification and reuse of what the traditional way could never do. The economic advantage of re-use cannot be overstated. The ability to rely on reusable quality products accelerates and makes production cheaper. It also gives developers confidence in the fact that there are already tested products that can be used when needed. This is mainly the case with the design process, which can be applied to all object-oriented software products. Instructions can be used through which the developer can build quality software. The ability to rely on reusable quality products accelerates and makes production cheaper. It also gives developers confidence in the fact that there are already tested

products that can be used when needed. This is mainly the case with the design process, which can be applied to all object-oriented software products. The fact that some of these factors may be in conflict is why object-oriented testing is a difficult proposition. Instructions can be used through which the developer can build quality software.

References

[1]. Gilmour, R. & Cobus-Kuo, L. , (2011). Reference Management Software: a Comparative Analysis of Four Products. *Issues in Science and Technology Librarianship.*

[2]. Ackoff, R. (2006). Why few organizations adopt systems thinking. Systems Research and Behavioral Science, 23(5), 705-708. *Alter*, S.

[3]. A.S.Tannenbaum, (1996). Computer Networks. Prentice Hall, *Upper Saddle River*, NJ, 3nd edition.

[4]. Vess L. Johnson, Angelina Kiser & Leon Kappelman, (2018). Women in IT Management: Significant Gains and Continuing Challenges. Pages: 244-255.

[5]. N Zeqiri, A Luma. Multimedia learning systems and its implementation for the subject advanced it skills. *30th Jubilee International Convention MIPRO*, 2007

[6]. Desperately seeking systems thinking in the information systems discipline. *In Proceedings of Twenty-Fifth International Conference on Information System*s (pp. 757-769). Alter, S. (2006).

[7]. Aurum, A., & Wohlin, C. (Eds.), (2005).Organizational learning: A theory of action perspective. *AddisonWesley.*

[8]. Avison, D., (2000). Engineering and managing software requirements. Heidelberg, Germany: Springer.

[9]. Alfred Waller Ltd. Avison, D.E., & Fitzgerald, G. (2003). Multiview: An exploration in information systems development (2nd ed.).

[10]. Where now for development methodologies? *Communications of ACM*, 46(1), 79-82. Bajaj, A., Batra, D., Hevner, A., Parsons, J., & Siau, K. (2005).

[11]. Baresi, L., Di Nitto, & Ghezzi, C. (2006). Systems analysis and design: Should we be researching what we teach? *Communications of the AIS,* 15, 478-493.

[12]. Barton, J., Emery, M., Flood, R.L., Selsky, J., & Wolstenholm, E., (2004). Toward open-world software: Issues and challenges. *IEEE Computer*, 39(10), 36-43.