Software Engineering a Web Based Performance Evaluation System and Devising Mathematical Model forPerformance Measurement

Bekim Fetaji [1], Majlinda Fetaji [1], Mirlinda Ebibi [2], Dzavid Begovic [3]

[1]*South East European University, Computer Sciences, Ilindenska bb, 1200 Tetovo, Macedonia*
[2]*International Balkan University, Information Technology, Skopje, Macedonia*
[3]*Department for Computer Sciences, University of Novi Pazar, Serbia*

Abstract. **This research study tries to foster a research on software engineering of a web based solution for performance evaluation. The contributions of the research study are development of a methodology perspective, mathematical model of estimation calculationand performance measurements with its analyses, insights and recommendations. The methodology includes applying different patterns, structural definitions and creating five-layered application that will follow the cyclomatic code analyses recommendations. Another contribution that the study tries to make is to become a good reference point for further researches in developing performance evaluation. Finally, the research results, insights and user evaluations have been stated and recommendations are provided.**

Keywords.**Performance measurements, web software solution, design patterns, evaluation processes.**

1. Introduction

The focus of this research study is the investigation into devising conceptually as well as development approaches and analyses of Web Based Performance and Research Evaluation System for Universities.

Performance evaluation systems are part of all contemporary institutions and companies and all these performance evaluation, or sometimes called appraisal systems, may be found in different forms serving sometimes general, other times more specific needs for the institution, with one and main goal: to help the institution increase the overall performance of its employees and provide an equitable measurement of employee's contribution to the institution.

This research study examines the performance evaluation systems and applications that help in evaluating employees with a company or academic staff in a university. The main focus of this research is to briefly explore existing performance evaluation systems and build academic staff evaluation system for Universities and implemented and evaluated in South East European University (SEEU).

2. Research methodology

The main research questions addressed in this research study are as follows:

1. What are the main benefits of having performance evaluation system?
2. What are the techniques of measuring performance evaluation of employees using electronic performance evaluation systems?
3. What are the necessary requirements for developing performance evaluation system that can satisfy the needs of an educational organization such as a University?

The research goals of the research study are as follows:

1. Researching existing performance evaluation systems
2. Analyzing the features, functionalities and results that one performance evaluation system can bring to an organization
3. Design and develop complete architecture of an performance evaluation system
4. Development of web-based application that will serve as a performance evaluation tool
5. Providing insights and recommendations for future further development of such systems

The research study is divided into two main parts:

- Theoretical approach – research existing performance evaluation systems and summarize features, advantages and disadvantages of such systems and then devising the conceptual design based on the background research and evaluation analyses.
- Practical implementation – development of web-based academic staff research performance evaluation system for SEE University

The theoretical part has been used for gaining knowledge for such systems with accent on finding best practices in developing such systems by summarizing features, advantages, disadvantages and possible issues that may arise while developing such types of applications [7].

Practical part is mainly divided into three development phases:

1. Design and Modeling of the entire solution
2. Database Analysis, Design and Implementation
3. Web Application development

The end result is the implemented developed software solution at the South East European University that serves for academic staff evaluation processes and will be easily extensible to become large general performance evaluation application for the entire University.
There are several analyses performed:

- Architectural Visualization
- Dependency Matrix
- Coding Standards
- Coding Approach
- Code Quality
- Other coding metrics

3. Review of Performance Evaluation Approaches

There are different performance evaluation approaches [5] that are suitable for different scenarios, types of organizations and the goal the organization wants to achieve. Sometimes performance evaluation systems may include multiple sub-systems which implement different performance evaluation approaches for different purposes.

In general, there are three performance evaluation approaches [4]:

- Behavior-based approaches
- Results focused approaches
- Evaluations of team performances

Behavior-based approaches use specific performance evaluation factors for evaluating employees. These factors can be quantitative or qualitative.

Results focused approaches are goal-oriented and objective-oriented which are considered as defensive for the organization because these approaches encourage a high level of participation, however since these are results-oriented, the same are considered as not very flexible.

Evaluations of team performances' approaches are those focused more on the team rather than the individual. These types of approaches can sometimes be problematic. To avoid team-related problems, it's generally accepted that such evaluations should be conducted by creating specific team-based matrix where each employee (vertically) will be evaluated for a given task (horizontally).
In general, there are four measured metrics [4]:

- Quality
- Quantity
- Timeliness
- Cost-Effectiveness

In the information technology era, all processes are becoming more and more automatic. Since performance evaluation is a process (not a form) [4], most of its parts can be completely automatic and gathering data through the entire process may be much faster, easier and provide much higher level of accuracy when performing complex measurements.

Nowadays, different types of organizations have implemented different types of systems. Some have implemented partial performance evaluation systems where only specific areas of employee's contribution are automatic by using technology, while other areas are still performed quite manually and both results are merged to give the end result of the performed measurement. Other organizations have implemented complete performance evaluation systems where the entire process is done semi-automatically.

In different types of organizations, there are managers and employees who are participants in the performance evaluation process.

Employee rates his/her own capacity or knowledge in the following areas in terms of current position requirements. If appropriate, the employee provides evidence at the appraisal discussion to support own rating. The filled in rating scores are submitted to the manager prior to the appraisal discussion.

Manager makes his/her own rating. In case of differences, both discuss and reach agreement. The final rating scores are filled in by the manager prior to both parties signing the form.

The research has shown that there are five mostly used types of ratings that employees and managers use while performing performance evaluation.

4. Conceptual Design

ASPES has been named the acronym for Academic Staff Performance Evaluation System.

ASPES represents a partial performance evaluation system that focuses on accomplishing the process of academic staff evaluation by gathering data through web application.

ASPES is designed following the usability recommendations [3] for University academic staff and is focused on the university needs to accomplish certain scenarios as part of the academic staff evaluation process.

The purpose of this system is to evaluate individual staff performance on an annual basis.

The aim is to recognize achievement, support continuous development and provide information for the processes of contract renewal and promotion.

In ASPES, Academic Staff members can fill their profiles by themselves and create professional curriculum that can be visible publically. Moreover, the system is focused on the researches accomplished by the academic staff members, which can be added in the system by filling up form suitable for this purpose.

Besides the main scenarios, ASPES also includes approval workflows as described in [2], so that faculty deans can approve/disapprove certain research items or published books by checking them prior to approval.

The system uses points-based criteria of assessment of scientific-research work of the academic staff at the South East European University.

Development of such systems is not an easy task, it requires a lot of analysis, proper planning and modeling.

There are various technologies used, different design patterns integrated into one whole system, complex structure that integrates with other services and other important elements that make the entire development of the application not an easy task.

Therefore, before starting the project, the complete software requirements specification (SRS) and features were clearly identified and listed.The process of approving scientific research items should be done by faculty deans or other representative that will be given special access to do this operation by application administrators.

Approving scientific research item should be very straightforward and easy. Mainly, the approved should see list of research items and have three main operational options to do:

- Approve research item
- Disapprove research item
- Make pending

The initial state of research items is 'Pending'. Approve research item means the item will become approved. This should be available only if item was previously Disapproved or in Pending state. Disapprove research item will be done when approved considers that this research is not valid or incomplete. If approved disapproves research item, it will not be editable anymore by the user unless it's made pending again, which means, the staff member who has added the item will be only able to see it but not edit. However, the staff member will be able to delete the item.

Make pending is in case when approved performs an action (approve or disapprove) but then considers there is need to make the item to Pending state whether for the staff member to be able to edit some parts of the scientific research item or in case to make some other checks.

Total points are calculated only for Approved research items, which represents the performance of the academic staff member for its completed researches.

5. Development Methodology

Since this is conceptually thought as web based application, it's obvious that there will be many technologies used to build and develop the entire application.

Two main researches have been performed:

- User Interface Usability Design
- Technology architecture and visualization

The user interface design research consists of steps to find current trends in creating user experience and user interface that users can find it intuitive and easy to use [2].

This research consists of few main parts:

- Structuring website layout
- Colour nuances
- UI technologies

To implement the UI researches in practice following recommendations from [1], there is need of defining important set of UI technologies to be used. Some of these technologies are used in 99% of the web sites development nowadays, some other are more rarely used.

To build strong, robust and a system that is easily extensible it is very important to research and define the technology platform and list of technologies that you can easily integrate all together to achieve the main goal, build the end-user software application [6].

The development methodology includes application of MVP, decorater, repository and other structural patterns, structural definition and creation of five-layered application that will follow particular code standards. Application architecture defines the application layers, design patterns used to accomplish different tasks and ways that the problem of developing the application is solved.

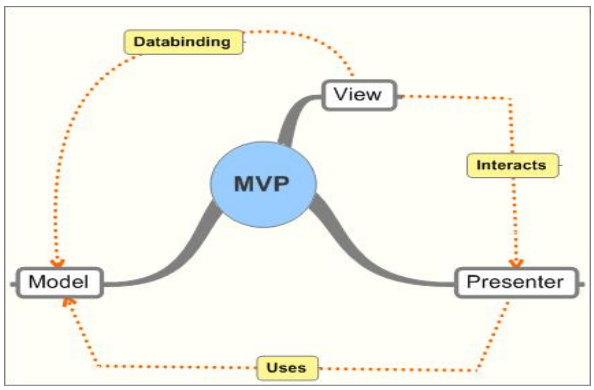

Figure 1. The development methodology

6. Cyclomatic Complexity Analyses

Cyclomatic Complexity or also known as 'Conditional Complexity' is measurement of complexity of a program code. The cyclomatic complexity is calculated using the control flow graph of the program, so this is based on the graph theory.[5]

Code Cyclomatic Complexity = Number of graph edges – Numer of Graph Nodes + Number of connected components.

Cyclomatic Complexity (CC): (defined for types, methods) (Only available for C# code, a VB.NET version is currently under development) Cyclomatic complexity is a popular procedural software metric equal to the number of decisions that can be taken in a procedure. Concretely, in C# the CC of a method is 1 + {the number of following expressions found in the body of the method}:

if | while | for | foreach | case | default | continue | goto | && | || | catch | ternary operator ?: | ??
Following expressions are not counted for CC computation:

else | do | switch | try | using | throw | finally | return | object creation | method call | field access
Adapted to the OO world, this metric is defined both on methods and classes/structures (as the sum of its methods CC). Notice that the CC of an anonymous method is not counted when computing the CC of its outer method.

Recommendations: Methods where CC is higher than 15 are hard to understand and maintain. Methods where CC is higher than 30 are extremely complex and should be split in smaller methods, except if they are automatically generated by a tool.

If we look at the standards, methods that have total Cyclomatic Complexity of 15 or higher, these are considered as complex.

Methods that have Cyclomatic Complexity 30 or higher, these are considered as critical and hard to maintain/change. There are 17 methods with small complexity which have Cyclomatic Complexity less than 15 higher than 5. Method Month(Int32) is the most complex method in this list. If we open it, the code looks like in the figure below.

What we can notice is that this is not very complex method, but only maps number values with month names for some specific purpose. Thus, refactoring is not needed.

When detailed Cyclomatic Complexity analyses were performed, surprisingly but true, there are no methods that are considered as complex with Cyclomatic Complexity 15 or more.

7. Estimation Calculation

The estimation calculation approach has been improved by modification of the Newton-Raphson iterative method for solving equations method outlined by Lord. The estimation equation used is shown below:

$$\theta_{n+1} = \theta_n + \frac{\sum_{i=1}^{n} S_i(\theta_n)}{\sum_{i=1}^{n-1} I_i(\theta_{n-1})}$$

where θ is the skill level after n questions, and ui = 1 if the response is correct and ui = 0 for the incorrect response.

$$S_i(\theta) = [0.9 - P_i(\theta)] \frac{P_i'(\theta)}{P_i(\theta)[u_i - P_i(\theta)]}$$

The standard error associated with a given ability is calculated by summing the values of the item information functions (IIF) at the candidate's ability level to obtain the test information. Test information, TI(θ), is given by the equation:

$$TI(\theta) = \sum_{i=1}^{N} I_i(\theta) \quad \textbf{8.} \quad SE(\theta) = \frac{1}{\sqrt{\mathrm{TI}(\theta)}}$$

Results analyses

Using jQuery Visualize, the results have been visualized in the following pie chart given below.

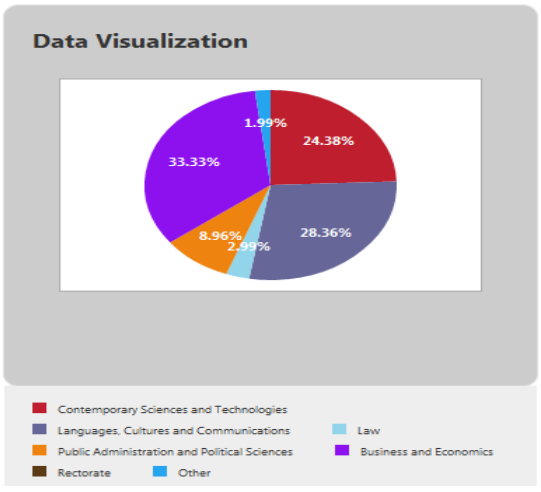

Figure 2. Data Visualization

Besides Data Visualization, on each two-dimensional table preview and chart preview, there is functionality to export the table data to Excel.

Architecture Visualization aim is to show the architectural overview of the entire project dependencies. In other words, this is called Dependency Graph.

Figure 3. Exporting data

As we can see from figure 5, the intended layers-based structure is satisfied from ASPES.Web (View page) perspective. ASPES.Web communicates only with ASPES. Presenter, ASPES.Presenter.BusinessObjects, ASPES

Reports and System libraries, but never ever communicates with ASPES.Task or ASPES.Model, what is interesting is that no one communicates with ASPES.Web, which is perfectly good because ASPES.Web does not know to serve anyone, but it only asks for data and visual representation.

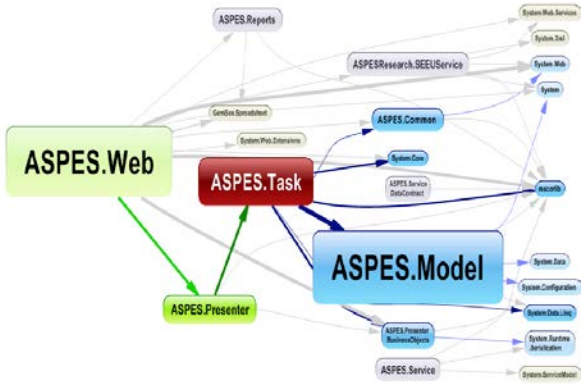

Figure 4. Software Architecture Visualization

There are many other coding metrics that have been analyzed for this study, including:

- Abstractness and Instability
- Treemap Metric View
- Biggest instances

Abstractness and Instability diagram should help detecting assemblies that are potentially painful to maintain (concrete and stable) and assemblies that are potentially useless (abstract and instable). Abstractness is big when assembly has many abstract types (interfaces, abstract classes, etc.) and few concrete types. Instability if assembly's types are used by few types of tier assemblies. It means more assembly is stable, means it is harder to maintain.

With the visualized code metrics, it is clear that most of the application code is optimized to satisfy the worldwide known standards.

The architectural visualization helps in viewing all dependencies visually and easily observe if the architectural design is broken or not. This graph is very useful when there are larger teams working on the project, where it is very hard to maintain clean architectural dependencies between libraries and layers, thus by using this graph we can confirm that our architecture still follows the rule that the architect has set up.

Zones of abstraction and instability can show us how much our code is tightly or loosely-coupled. This way, we can visually see whether we have stable assemblies or not. If it goes to the zone of instability, it means it is more stable but lacks abstraction. There is no perfect case and silver bullet when seeing design things from this side, therefore your assemblies would be good if these aren't going too much to the Zone of Uselessness or Zone of Pain.

The code metrics help confirm that we have good job done and have good architecture, but these cannot confirm that your code runs correctly regarding the logic of implementation.

9. Conclusion

The research study contributes with new proposed methodology for developing Performance Evaluation systems, cyclomatic complexity analyses, estimation calculation and performance measurements with its analyses, insights and recommendations.

The methodology includes applying different patterns, structural definitions and creating multi-layered application that will follow particular code standards.

The focus of this research study was to use the gained theoretical knowledge to apply in practical scenario and build robust, flexible and performance evaluation systems that will serve to accomplish specific scenarios in performance evaluation process of staff members in University and similarly in other organizations as well.

The entire design of the system is easily extensible. The database scheme is already designed

to support extensibility.By completing this workflow and having all other features at hand, ASPES can be counted as fully functional system that will help automatically evaluate staff members based on their submitted research papers. Data Visualization and Reporting features are additional functionalities of ASPES, which can be easily extended and new reports can be added pretty much easily into the existing system. All performed analysis have improved the overall design of the system, application architecture, structure, code quality and application usability from user perspective.

By having such fundamental ground, achieved insights and results can be used for further future development of such systems, that will help institutions and organizations automate performance evaluation processes and achieve better results in the overall evaluation process.

References

[1] Fixsen, D. L., &Blase, K. A. (2009). Implementation: The missing link between research and practice. *NIRN implementation brief, 1*.

[2] Ebibi M, Fetaji B. Fetaji, M. (2012), "Combining and suporting Expert Based Learning and Academic Learning in Developing Mobile Learning Knowledge Management System", Journal TEM, Technology Education Management Informatics, 1(1), 16-23.

[3] Fetaji, M., Dika, Z., &Fetaji, B. (2008, June). Usability testing and evaluation of a mobile software solution: a case study. In *Information Technology Interfaces, 2008. ITI 2008. 30th International Conference on* (pp. 501-506). IEEE.

[4] Behn, R. D. (2003). Why measure performance? Different purposes require different measures. *Public administration review*, *63*(5), 586-606.

[5] Camil Sukic, Mirlinda Ebibi, Bekim Fetaji, (2011) Education in the digital environment and improving e-learning“, WCIT2011, World Conference on Information Technology, Antalya, Turkey.

[6] Lovrekovic, Z., &Lovrekovic, T. (2012). Software Application for Managing Multidisciplinary and Interdisciplinary Projects as a Part of Educational Process. TEM Journal, 1(1), 38-45.

[7] Tan, W. L., Lau, W. C., Yue, O., &Hui, T. H. (2011). Analytical models and performance evaluation of drive-thru internet systems. *Selected Areas in Communications, IEEE Journal on*, *29*(1), 207-222.

Corresponding author: Bekim Fetaji
*Institution:*South East European University, Computer Sciences, Tetovo, Macedonia
E-mail: b.fetaji@seeu.mk