

DOI <https://doi.org/10.30525/978-9934-26-597-6-91>

USING SURVEY RESULTS AS A TOOL FOR DEVELOPING SOFTWARE MAINTENANCE REQUIREMENTS

Sunnatjon Akhmedjanov, Nikita Petkuns, Anastasija Vostrikova

Latvia, ISMA

*e-pass: ssakhmedjanov010@gmail.com, npetkuns@gmail.com,
vos.anastasija@gmail.com*

Abstract

This research investigates the use of survey results as a means for establishing software maintenance requirements. At its essence, the survey serves not only to evaluate the suitability of candidates for intricate software development positions but also as a document that summarizes employer expectations and technical priorities. By conducting a thorough analysis, the responses from the survey are converted into high-level specifications that inform software maintenance planning.

Keywords: provisions, whole, circumstances, model, program, result, document

1 Introduction

By definition, a questionnaire is used as a means of identifying a worthy candidate for a vacant position. In this regard, it is necessary to highlight the main places of the questionnaire and, based on them, determine the needs of the employer. The following important provisions were developed during the analysis

- Programming languages – a professional skill in the field of software development [1].
- Specific languages, such as management, design, system, engineer, determine the ability to organize and manage the development process.
- Conference – the ability to share skills with colleagues.
- Project – skills that demonstrate not only the ability to apply the results of development in practice, but also to formulate a proposal for the investor.
- Key Word – immersion is assumed (fundamental knowledge to effectively support the system).
- ROI – a comprehensive indicator of evaluation and determination of the effective distribution of invested assets [2].

- Reicheld's Three Loyalties Effect – makes it possible to ensure a connection of interests between three participants in the process; clients, investors, project managers [3].

- Implementation time – the ability to determine the labor intensity and ensure the implementation of the project on time within the agreed funding.

- System – the ability to support the system and retain the right to manage the development for a long time.

- The share of management in the process – a high percentage indicates the importance of the role of the system developer.

Based on the provisions, two key points were identified.

- Firstly, the authors of the questionnaire took into account the requirements of the employer who is interested in a developer of complex software.

- Secondly, such a developer must not only organize the activity, but also exercise control over it at the entire stage of the support life cycle

The listed circumstances characterize the system of four platforms. In order to implement such a system in practice, it is necessary to use the DIKW model [4]. The ROI model is used as an element of DIKW purposefulness.

2 Approaches

The study revealed a contradiction that boiled down to the fact that as new levers for the effective use of ROI technology are revealed, uncertainty about its effective use arises. This made it possible to identify the research problem. Its content is as follows: "It is impossible to resolve the phenomenon of the ROI methodology in the absence of a complete set of recommendations for its effective use." The object of the study is the result of combining two models, DIKW and ROI. During the combination, a synergistic result is achieved. In other words, generalizing the unique properties of the object, the position of its study is strengthened. The presence of methodological provisions allows not only to get answers to all questions, but also to form a document reduced to the form of an expert opinion.

The purpose of the study is focused on developing a procedure that ensures the use of the DIKW approach to establish the degree of consistency of all elements of the ROI system. In accordance with the stated goal, four tasks were formulated:

1. Study the DIKW model to identify signs of effective use of ROI.
2. Debug the program for linking the source data and output results.
3. Develop an algorithm for resolving the commoditization effect [5].
4. Suggest an approach to coordinated impact on key business parameters.

This approach makes it possible to ensure the consistency of all elements. After all provisions have been agreed upon, the applicant develops the skill of identifying the main link, knowledge of which allows revealing the content of the hidden plan, and on its basis, linking together all elements of the system.

3 Conclusions

In summary, this research illustrates how surveys, when strategically interpreted, transform into effective tools for identifying software maintenance requirements – bridging the divide between vague employer expectations and specific technical specifications.

The results are categorized into two sections: the first part concentrates on technical skills—including programming languages, project experience, and knowledge sharing – while the second part highlights strategic metrics such as ROI, implementation timelines, and system sustainability. The analysis reveals twelve critical indicators that collectively outline the profile of a developer suited to both lead and maintain software throughout its lifecycle.

References

- [1] Kossiakoff A., Swee N., Seymor S. Bier S., (2011). *Systems Engineering Principles and Practice*. John Willey & Sons.
- [2] Kelly, N. (2013). *How to Measure Social Media: A Step-by-Step Guide to Developing and Assessing Social Media ROI*. Boston: Pearson Education.
- [3] Reichheld, F. (2006), *The Ultimate Question: Driving Good Profits and True Growth*, Harvard Business School Press.
- [4] Ackoff, R. L. (1989). From data to wisdom. *Journal of Applied Systems Analysis*, 16, 3–9.
- [5] Aulet B. (2017) *Disciplined Entrepreneurship Workbook*, 288p.