

Visualization, Monitoring and Control Techniques for Use in Scrum Software Development: An Analytic Hierarchy Process Approach

Nesib Tekin¹, Mehmet Kosa², Murat Yilmaz^{3,4}, Paul Clarke^{3,4}, Vahid Garousi⁵

¹ Turkish Defense Industries Research and Development Institute – TUBİTAK
SAGE, Ankara, Turkey.

nesib.tekin@tubitak.gov.tr

² Cognitive Science & Artificial Intelligence (CS&AI) Department, Tilburg
University, Netherlands.

m.kosa@tilburguniversity.edu

³ Lero, the Science Foundation Ireland Research Centre for Software.

murat.yilmaz@dcu.ie

⁴ Dublin City University, Dublin, Ireland.

Paul.M.Clarke@dcu.ie

⁵ Queen's University Belfast, Belfast, UK.

V.Garousi@qub.ac.uk

Abstract. Scrum is the most widely used agile development framework that guides the development process with its ability to create customer-valued software artifacts iteratively and incrementally, whilst seeking best practices to provide continuous measurement during production. However, measuring success in Scrum can be a challenging endeavor. In particular, it is hard to select the best fitting agile metrics during consecutive Scrum sprints. The goal of this industrial case study was to utilize a systematic selection process for identifying the appropriate scrum metrics tools add-on component within the TÜBİTAK SAGE software development group. Moreover, the distribution of software developers' preferences of process metrics were analyzed according to their characteristic features and defense industry structure, and are presented using various distribution charts. Finally, alternatives to the software development process measurement component, which was integrated into the agile software process tool employed by the TÜBİTAK SAGE software development group, were efficiently determined by using the Analytic Hierarchy Process approach. Among the options discussed, our results suggest that the Actionable Agile Addon scored the highest followed by the Screenful Addon. The present study presents a rigorous approach that ultimately have improved community participation in metric planning, implementation and monitoring, thus moving towards sustainable software development goals.

Keywords: Software Measurement Component, Software Process Metrics Tool, Scrum, AHP, Software Component Selection, industrial case study.

1 Introduction

The notion of quality has become crucial across all engineering disciplines, and can be assessed through various forms of measurement, which is a crucial element to all scientific and engineering activity. Therefore, software engineering is no exception as it is essentially an engineering discipline. According to Pfleeger [1], software measurement is an inseparable part of software development and maintenance. Throughout the software development lifecycle, measurement processes should be used effectively in order to evaluate the quality, as well as the improvement [2] and performance of the product [3]. Today, performance measurement has become a key feature in the development of successful software engineering applications [4], and productivity [5].

The SPI manifesto suggests creating a learning organization [6]. This promotes the fact that metrics within the software development process should provide valuable information about the product under development in such organizations. One of the essential goals of the software metric is the aim to eliminate human-factor uncertainties within the software measurement process [7]. Since the collected metric data reflects the problem in the development process, a company can use such data to formulate regulatory actions and to improve the software development process [8]. There are many different software metrics (e.g., lines of code, code complexity, cycle time and velocity, mean time to recover) within the software development domain. In the literature, different approaches have been applied in classifying these metrics. Lee and Chang's [3] classification related to the metrics necessary for software measurement based on software quality. They divided software quality metrics into five groups: product quality; in-process quality; testing quality; maintenance quality; and, customer satisfaction quality. Another grouping can be expressed as: commercial perspective; significance perspective; observation perspective; measurement perspective; and, software development perspective. Commercial perspective includes technical metrics, defect metrics, end-user satisfaction metrics, warranty metrics, and reputation metrics. Significance perspective includes core and non-core metrics. Observation perspective contains primitive and computed metrics. Measurement perspective involves direct and indirect metrics [9]. Software development perspective includes process metrics, product metrics, test metrics, maintenance metrics, and subjective metrics [4]. However, Lee and Chang [3] pointed out that a metric can be included in one or more of these categories. Scrum is one such modern agile software development framework that is widely used and known. The most important change in the Scrum approach is the transfer of responsibilities and decision mechanisms that previously sat with administrative staff to the actual software development teams [10]. Therefore, an important purpose of Scrum metrics is to help the software team and their managers to monitor the business development process, as well as business quality, productivity, predictability, the health of the product and the team [11].

Organizations generally choose software-based development process tools in order to more easily control comprehensive system developments. Various software development process tools are widely used by companies to manage the

Scrum process. In fact, tools have been developed for specific market areas that contain several add-ons. Various developers have created adaptive applications for these software development process tools, and companies can purchase and integrate these applications into their own software development process tools as an add-on. The variety of software process metrics increases daily. Commonly used process metrics have mostly been integrated into process tools as default. However, companies can also expand the scope of these process metrics by purchasing add-ons based on their specific business needs. The most commonly used Scrum problem and project tracking software tool contains a total of nine software process metrics by default. However, there are also 66 metrics add-ons available on the market. Their content, functionality, technical specifications, websites, usage training, and demonstrations are also available in the market.

The aim of the current study is to select the software development process measurement component which includes the appropriate software development process metrics by using the AHP method. Also, software project requirements and software developers' contributions are also considered in the selection process. The remaining part of the paper proceeds as follows: The next section of this paper reviews the literature. Third section elaborates on the research methodology. The fourth section discusses and summarizes the results. Finally, conclusion section concludes the study.

2 Background

Agile development practices enable self-development and ultimately embrace the human experience [12]. Scrum methodology consists of iterative and incremental sprint structures, and software development roles [13] with the target determined prior to a sprint being started. Scrum metrics indicate whether or not the target of the sprint can be accomplished. The most fundamental example is where the result of a sprint presents new functionality of a product. The Scrum team can monitor quantitative evaluation of the work, success rates of the sprint, and the maturity level of the team by using Scrum metrics. Sprint's success rate is an important starting point for adaptation and inspection [14].

2.1 Importance of Choosing the Right Metrics

Pfleeger and Fitzgerald [15] conducted a detailed study about selecting the right software metrics toolkit. They underlined that requirements, process, and maintainability are important factors which should be considered during the software metric selection process. Additionally, they defined that customization and coordination of the toolkit can be time-consuming activities. Card and Glass [16] stated that data collection and analysis in the calculation of metrics increased the project cost by 7-8% in a software engineering laboratory at the University of Maryland. While the cost of metrics in the project and data collection are important, accurate data collection is vital in order to generate accurate metrics. Spending both in terms of time and cost for unnecessary metrics should be avoided.

Research on software measurements continues to be a hot topic today. Although the benefits of using of software measurement are well known, problems in their practice are still ongoing. More than 80% of software measurement attempts fail within the first 18 months [17]. The most common explanation for this is the difficulty in understanding and using metrics. According to Fenton and Neil [18], metrics are not used effectively within the decision-making process, because they do not have reliability in terms of validity.

2.2 Importance of Software Process Metrics

Kitchenham et al. [19] mentioned that more sensitive and considerate behavior is required in the measurement process. Ptleeger [1] emphasized a very strong motto that better decisions can be made on various subjects by measuring the past and changing the future. This situation requires the recruitment, training and building of a team based on software quality. In particular, to improve coordination and collaboration between software development teams, Huawei Turkey Research and Development Center proposed a version Hoshin Matrix and mapped software metrics and organizational goals [20].

Measurement and competition are factors that encourage people to improve on their previous performance in a situational context [21]. Software measurements help to understand how a project performs according to its goals, and provides information about the state of an organization compared to some previous period [22]. Pfleeger and Fitzgerald [15] explained that a software metrics toolkit should be determined related to the needs of the software being developed. Also, when choosing metrics for a software project development process, three concerns should be taken into consideration: (i) maturity of the software development process; (ii) availability of measurement data; and, (iii) the project management requirements. However, several drawbacks were revealed when the selection process for a metrics toolkit were determined by only the project managers. These disadvantages were that metric customization is both time-consuming and difficult in terms of coordinating the usage of the metrics. Paulish and Cartelon [8] suggested that an evaluation process should be realized prior to starting a software process improvement program. The researchers indicated that the evaluation can be a powerful method that addresses priorities and consensus within an organization that focuses on improvement. According to Ebert et al. [22], metric creators should not act independently from metrics users. They highlighted that users of metrics should comprehend the fundamental information about the measurement process of the software.

2.3 Analytic Hierarchy Process (AHP) Framework

The Analytic Hierarchy Process (AHP) approach, which is one of the multi-criteria decision-making techniques, integrates different types of criteria into a hierarchical structure, enabling the evaluation of each alternative [23]. There have been a limited number of studies using multi-criteria decision-making methodologies such as AHP related to the selection of software metrics and their tools.

Sharma et al. [24] used the AHP method, which is one of the multi-criteria decision-making method, for selecting software related to the project management process. Sagar et al. [25] proposed the fuzzy AHP method for selecting the most reusable software component such as plug-ins. They determined criteria that pointed to reusability as adaptability, availability, interface complexity, customizability, and understandability by reviewing the related literature. Ömürbek et al. [26], the aim was to select a project management program which could be used in software development based on using AHP and the TOPSIS methods.

In a study by Al-Qutaish et al. [27], the AHP method was used to manage the selection process of open-source software according to the ISO 9126-1 standard related to a set of six quality characteristics: functionality; reliability (R); usability; efficiency; maintainability; and, portability. Zaidan et al. [28] presented open-source electronic medical record software packages selection approach based on the AHP and TOPSIS methods. The results of the study showed that GNUmed and OpenEMR software had the most high-ranking scores when compared to other open-source EMR software packages.

All in all, the software process metrics in improving software quality is important to understand, as well as the cost of applying unnecessary metrics. Moreover, end-users themselves should be included in the metric selection, efficiency improvement and related activities [29]. However, software development process metrics are presented as a generic structure by the software process tracking tool. Also, there are several metrics add-ons on offer for the Scrum process metrics as sub-group kit products. It is a difficult and complex issue deciding which add-ons to use because the process is known to be affected by multiple criteria such as the type of software being developed, the development process model, the experience of the software developers, the domain of the project, and its duration. Therefore, a systematic selection process is needed as a solution to this multi-criteria problem.

In the current study, the selection of add-ons which include the software process metrics was performed using the AHP method. The purpose of the current study is to examine the selection of components that offer software development process metrics based on the AHP method. In this context, the weighting of selection criteria and the evaluation of alternatives were conducted in order to select add-ons that include software process metrics for an industrial case study in the TÜBİTAK SAGE software development group.

3 Research Method

First, the issue of appropriate software development process measurement component selection was determined. Then, a survey was applied to the study group. After that, the AHP method was implemented with the contribution of the expert group. The priorities of both criteria and alternatives were then evaluated. Finally, the highest priority alternative was selected as the final decision. Figure 1 illustrates each step of the case study process.

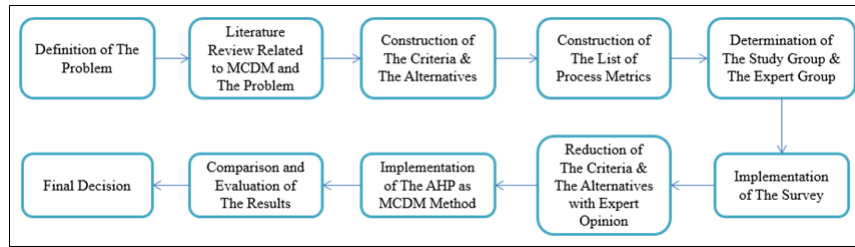


Fig. 1. Case study process steps

A list of software process metrics that are deemed appropriate to the Scrum software development process was constructed based on:

- The tool of the Scrum issue and the project tracking software, which has nine software process metrics by default.
- Various Scrum process metrics that can be integrated into the tool as an add-on to the Scrum issue and project tracking software.
- How users can customize the Scrum process metrics with the help of any available add-ons.

Study group: The study group consists of project software developers who take responsibility as developers of the software during the development process. This group was selected because they understand the corporate culture, the software development process, are aware of the things they are doing well or any problems, and have significant experience in the software development domain. **Expert group:** The expert group consists of employees who have at least 5 years' experience in software development. Group members have titles such as Team Leader, Unit Manager, Coordinator, and/or Chief Scientist. Also, this group has the primary decision-making responsibility for the software development process.

A software development process meeting was organized in the unit where the study was conducted. The participants were informed about the process metrics survey. Both the study group and the expert group participated in the meeting, and were asked to review the list of software development process metrics and to report their preferences through completion of the survey. Customized process metrics could be suggested by the participants when completing the survey as indicated at the first meeting with the study group. In addition, the participants were informed that demographic data about their years of work experience, experience domains, graduation departments, and their working units would be collected within the survey.

The survey was prepared using Google Forms as a template to create the survey. Additionally, data collected via Google Forms could later be converted into graphical format. The survey schedule was announced to the participants 7 days in advance of its application. The results were grouped and associated with the demographic data collected from the participants. The findings obtained

from the survey were then presented graphically to the expert group. Reduction of the criteria and the alternatives was conducted by the expert group according to the results of the survey.

In determining the software development process metric add-ons, consideration was given to the ability that add-ons could be integrated to the Scrum software development process tool used by the company at which this industrial case study was applied. The market for the Scrum issue and project tracking software was then examined, and a list of 66 products was identified as potential add-on applications. Each product was then examined according to the following:

- Technical specifications and capabilities (types and number of process metrics);
- Usage rate in the market;
- Scoring rate in the market.

The alternative add-on products were compared and their functionalities considered based on requirements suggested for software developers' process metrics. Each alternative product was then examined according to the following:

- Technical specifications that meet the requirements;
- Accessibility and availability;
- Includes visualization and/or report mechanisms.

The literature included several criteria for metrics tools. It was observed that software process metrics tools aim to increase market sales by highlighting these criteria, and that important criteria were emphasized within the technical documents of each metric tool (see Table 1)

While evaluating the software process metrics, robustness, simplicity and cost-effectiveness were seen as the key parameters. This is because the processes are defined according to systematic and powerful procedures. Each process was found to have been described quite simply. Finally, it was emphasized that the processes should be cost-effective. The following criterion group was then used in evaluating the tools offered for the software development process metrics.

The expert group performed the reduction process by considering the results of the survey. After that, the criteria and alternatives list in the AHP method were finalized. Reliability of the reduction operation was assessed by using Cronbach's alpha method. It is represented by a numerical value between 0 and 1 in terms of a reliability scale.

4 Results

The study group consists of 28 software developers. Figure 1 shows the working units of the study group members. The study group members work in four different units: Embedded Software, Software Simulation, Software Testing, and Software Architectural Design. There most common working group for the study

| Criteria | Meaning |
|--------------------------|---|
| Relevance | (to collect process metrics) Reflects the relationship between process metrics and the metric tool. |
| Experience | Degree to which the metric tool has been used and recognized. |
| Correctness | Assessment based on a tool's objectivity, justness and precision. Objectivity: input and results cannot be easily influenced. Justness: no specific result should form part of the metric tool. Precision: tool measures to a precise degree. |
| Practicality | Required and relevant to both development and improvement. |
| Feasibility Usability | Assessment of a tool based on three conditions: Understandability of all formulae in the tool should be high; data collection should be easy; and, evaluation of the metric's results should be convenient. |
| Functionality | The metric tool should meet technical requirements, with a high number of essential process metrics, advanced level strength of visualization and reporting mechanism. |
| Adaptability Portability | The metric tool can be integrated to the process methodology. It should be considered as portable and easy to integrate in a minimal time to the software development process management tool. |

Table 1. Software process metrics add-ons evaluation criteria

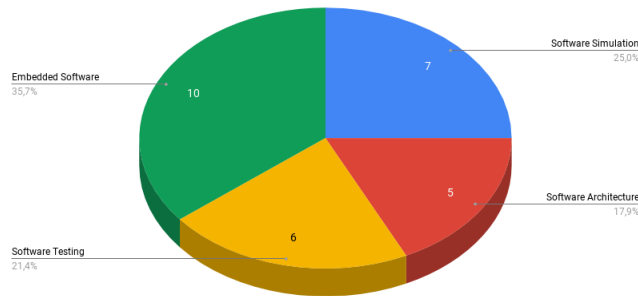


Fig. 2. Study group by working unit

group members was Embedded Software ($n = 10, 35.7\%$), and the least common was Software Architectural Design ($n = 5, 17.9\%$).

Cumulative Flow Diagram was the most selected software development process metric, and was selected by 15 (53.6%) of the study group members. Burndown Chart, Velocity Chart, and Burnup Chart were the options selected by more than 10 of the participants. The Blocked Issues Chart and Contribution Chart are marked with an asterisk symbol (“*”) because they were additional options added by the participants. Potentially Deliverable Scope and Feature and Epic Progress were each selected by only one study group member. Most software development process measurement components were observed to contain the Cumulative Flow Diagram and Burndown Chart metric options. In other

words, the majority of the selections made by the study group can be covered by the software development process metric providers.

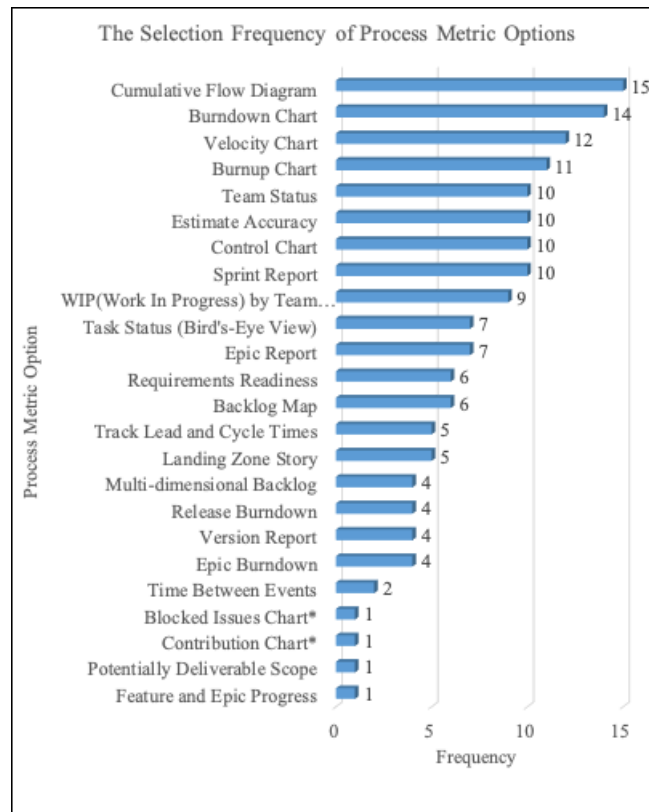


Fig. 3. Selection frequency of process metric options

The add-on products that can be integrated into the software development process management tool used by the TÜBİTAK SAGE software development group were evaluated. It is known that the global marketplace of the tool includes 66 add-on products for software process metrics. Some of these add-on products are considered primitive or are highly specific products that offer only a few metrics, whilst some offer a more comprehensive process metrics service. The tool currently used for software development at the company in this industrial case study provides nine common process metrics. Considering the varied suggestions made by the study group members, it is understood that the current tool contains an insufficient variety of process metrics. Moreover, it is seen that advanced add-on products that offer preferred process metrics were seen as useful by the study group members. As a result, user reviews in the market, user ratings, and

software development experts' opinions identified alternatives for four add-on products. The alternatives were determined from feedback of the expert group. Here are the four alternatives (i) Actionable Agile¹, (ii) SenseAdapt Agile², (iii) Screenful³, (iv) Predictable Agile⁴ with the determined criteria (1) Feasibility & Usability, (2) Functionality, (3) Relevance, and (4) Experience.

The hierarchical structure was established between the criteria and the alternatives. Figure 4 presents the AHP hierarchical structure of the software development process measurement component selection. It shows the hierarchical structure of the criteria and alternatives determined for the selection of software development process metric add-ons.

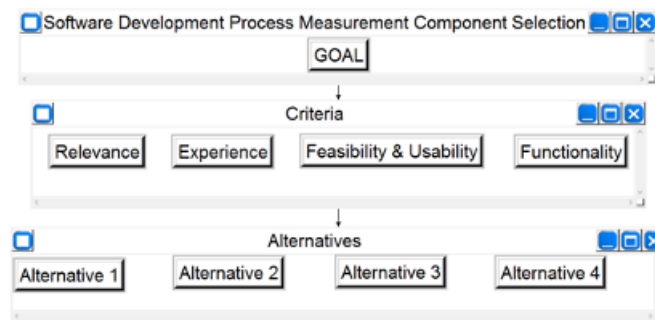


Fig. 4. AHP hierarchical structure of component selection

The decision matrix was created by using the Super Decisions program. The Relevance criterion had the highest weight value of .492, followed by the Experience criterion with a weight value of .225, the Feasibility and Usability criterion with a weight value of .170, and the Functionality criterion had the lowest weight value of .112. As a result of the matrix, weights for each of the alternatives were obtained. Alternative-1 had the highest weight with a normalized value of .402, Alternative-3 was second with a normalized value of .236, Alternative-2 was third with a normalized value of .188, and Alternative-4 had the lowest weight with a normalized value of .173.

In the current study, the software development process metrics preferred by the software development group of the company in this industrial case study were reflected. The preferred metrics were considered as being suitable for the characteristic structures of the study groups, and also to the company. Moreover, the most selected metrics such as Cumulative Flow Diagram and Burndown Chart were found to be commonly provided within all software process measurement

¹ <https://actionableagile.com/>

² <https://ripperrock.atlassian.net/wiki/spaces/SEN/overview>

³ <https://screenful.com/tour>

⁴ <http://agilemontecarlo.com/>

components. In other words, the study group was not considered to be contradictory in their selections. When evaluating the criteria by the expert group, it can be considered that the study group's preferences were taken into consideration as the *Relevance* criterion showed a significant value. It can therefore be said that the expert group were afforded sufficient time for the evaluation process so as to produce accurate and representative results, having reflected objective evaluations at the end of the process.

5 Conclusions

In the industrial case study conducted, the AHP method was applied, resulting in the *Relevance* criterion determined as the highest priority by the software development team. It was considered by the study group's members that the institution, the projects, and the software development team proposed metrics that were deemed suitable for the characteristic structures pertinent to the situation being assessed. Thus, the decision-making expert group examined the graphical results of the collected process metrics.

Here, it can be said that the expert group members were affected in accordance with the need for the *Relevance* criterion. In addition, it can be said that information such as comments, scores and the use of percentages from those with relevant experience may have been the reason for giving second priority to the *Experience* criterion. In addition, it was determined that the study group members gave almost equal priority to both the *Functionality* and the *Feasibility* and *Usability* criteria.

Expert-4, who considered the *Functionality* criterion to be of a low priority, verbally stated that "It is more important to use the proportion of features than the high functionality of the vehicle." An example of this would be that some of the tools used by the company have been upgraded with 100 new features; yet these new features were not being used, whilst the basic functions were continued to be used instead.

It was also determined that *Alternative-1* was prioritized as the most preferred with 40.259% following the systemic evaluative process of the available alternatives. It was observed that the expert group had difficulty in applying discrete thinking at the beginning of the evaluation process. Once this problem had been identified, it was determined that the expert group had successfully completed this process when an appropriate time and place were considered for them to ponder their decisions.

The industrial case study of the current study has proven that multi-criteria decision-making process, and in particular the AHP method, can be used effectively for selecting software metric components within the software domain. In the future, it is considered that the current study may serve as an example of the AHP method being preferred when systematic decision-making is required in the field of software development. In addition, it is considered that the contribution of the software development team to the decision-making process positively supports the use of the selected metric components. The use of appropriate software

development components increases the ability of companies to compete in the future.

Acknowledgments

This research work is supported, in part, by TÜBİTAK SAGE. During the research, we have received help and generous support from many people, to all of whom we would like to express our gratitude.

References

1. Pfleeger, S.L.: Software metrics: Progress after 25 years? *IEEE Software* **25** (2008) 32–34
2. Yilmaz, M.: Observed effects of software processes change in three software firms: Industrial exploratory case study. *Pamukkale Univ Muh Bilim Derg* **25** (2019) 240–246
3. Lee, M.C., Chang, T.: Software measurement and software metrics in software quality. *International Journal of Software Engineering and Its Applications* **7** (2013) 15–34
4. Farooq, S.U., Quadri, S., Ahmad, N.: Software measurements and metrics: Role in effective software testing. *International Journal of Engineering Science and Technology* **3** (2011) 671–680
5. Yilmaz, M., O’Connor, R.V., Clarke, P.: Effective social productivity measurements during software development—an empirical study. *International Journal of Software Engineering and Knowledge Engineering* **26** (2016) 457–490
6. Manifesto: Software Process Improvement (SPI). (Last accessed September 3, 2019) https://2019.eurospi.net/images/eurospi/spi_manifesto.pdf.
7. Yilmaz, M.: A software process engineering approach to understanding software productivity and team personality characteristics: an empirical investigation. PhD thesis, Dublin City University (2013)
8. Paulish, D.J., Carleton, A.D.: Case studies of software-process-improvement measurement. *Computer* **27** (1994) 50–57
9. Yilmaz, M., O’Connor, R.: Social capital as a determinant factor of software development productivity: An empirical study using structural equation modeling. *International Journal of Human Capital and Information Technology Professionals (IJHCITP)* **3** (2012) 40–62
10. Yilmaz, M., O’Connor, R.V.: A scrumban integrated gamification approach to guide software process improvement: a turkish case study. *Tehnički vjesnik* **23** (2016) 237–245
11. Jones, C.: A guide to selecting software measures and metrics. CRC Press (2017)
12. Yilmaz, M., Atasoy, B., O’Connor, R.V., Martens, J.B., Clarke, P.: Software developer’s journey. In: *European Conference on Software Process Improvement*. Volume 633., Springer (2016) 203–211
13. Yilmaz, M., O’Connor, R., Clarke, P.: A systematic approach to the comparison of roles in the software development processes. In: *Software Process Improvement and Capability Determination*. Volume 290., Springer-Verlag (2012) 198–209
14. Wan, J., Zhu, Y., Zeng, M., et al.: Case study on critical success factors of running scrum. *Journal of Software Engineering and Applications* **6** (2013) 59

15. Pfleeger, S.L., Fitzgerald Jr, J.: Software metrics tool kit: support for selection, collection and analysis. *Information and software technology* **33** (1991) 477–482
16. Card, D.N., Glass, R.L.: *Measuring software design quality*. Prentice-Hall, Inc. (1990)
17. Wallace, L.G., Sheetz, S.D.: The adoption of software measures: A technology acceptance model (tam) perspective. *Information & Management* **51** (2014) 249–259
18. Fenton, N.E., Neil, M.: Software metrics: roadmap. In: *Proceedings of the Conference on the Future of Software Engineering*. (2000) 357–370
19. Kitchenham, B., Pfleeger, S.L., Fenton, N.: Towards a framework for software measurement validation. *IEEE Transactions on software Engineering* **21** (1995) 929–944
20. Akarsu, Z., Ozgun, M., Kuru, Y., Yilmaz, M.: Using adapted version of hoshin matrix for selection of agile software development processes. In: *12th Turkish National Software Engineering Conference (UYMS 2018)*. (2018) 1–11
21. Marks, G., O, R.V., Yilmaz, M., Clarke, P.M.: An iso/iec 12207 perspective on software development process adaptation. *Software Quality Professional* **20** (2018) 48–58
22. Ebert, C., Bundschuh, M., Dumke, R., Schmietendorf, A.: Making metrics a success—the business perspective. *Best Practices in Software Measurement: How to use metrics to improve project and process performance* (2005) 9–34
23. Saaty, T.L.: What is the analytic hierarchy process? In: *Mathematical models for decision support*. Springer (1988) 109–121
24. Sharma, N.K., Gupta, S., Sharma, V.: A comparatively study for project management software selection using analytic hierarchy process method. *International Journal of Engineering and Management Research (IJEMR)* **5** (2015) 188–195
25. Sagar, S., Mathur, P., Sharma, A.: Multi-criteria selection of software components using fuzzy-ahp approach. *International Journal of Innovative Computing Information and Control* **11** (2015) 1045–1058
26. ÖMÜRBEK, N., MAKAS, Y., ÖMÜRBEK, V.: Ahp ve topsis yöntemleri ile kurumsal proje yönetim yazılımı seçimi. *Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü Dergisi* (2015) 59–83
27. Al-Qutaish, R.E., Muhairat, M.I., Al-Kasasbeh, B.M., Al-Kasasbeh, B.: The analytical hierarchy process as a tool to select open source software. In: *Proceedings of the 8th WSEAS International Conference on Software engineering, parallel and distributed systems, Cambridge, UK*. (2009) 39–44
28. Zaidan, A.A., Zaidan, B.B., Al-Haiqi, A., Kiah, M.L.M., Hussain, M., Abdulnabi, M.: Evaluation and selection of open-source emr software packages based on integrated ahp and topsis. *Journal of biomedical informatics* **53** (2015) 390–404
29. Orgun, P., Gungor, D., Kuru, Y., Metin, O., Yilmaz, M.: Software development overall efficiency improvement in a cmmi level 5 organization within the scope of a case study. In: *Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı (UBMK 2018)*. (2018) 1–12