Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

ARTICLE TYPE

A Novel Approach for Visualization, Monitoring and Control Techniques for Scrum Metric Planning using the Analytic Hierarchy Process

Nesib Tekin[1] | Murat Yilmaz*[2] | Paul Clarke[3,4]

[1]Turkish Defense Industries Research and Development Institute – TUBİTAK SAGE, Ankara, Turkey.
[2]Department of Computer Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey.
[3]Lero, the Science Foundation Ireland Research Centre for Software.
[4]Dublin City University, Dublin, Ireland.

Correspondence
*Corresponding author Murat Yılmaz, Email: my@gazi.edu.tr

Present Address
Department of Computer Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey.

Summary

Although Scrum is one of the most preferred agile development frameworks that guide the development process, measuring sprint productivity is still challenging. In fact, it is hard to provide a continuous measurement during consecutive Scrum sprints, especially selecting the optimal metrics that fit better for real industrial applications. To bridge this gap, we conducted an industrial case study within the TÜBİTAK SAGE software development group to demonstrate the performance and applicability of a systematic selection process for fitting addon components using various scrum metrics tools. Next, we analyzed the combination of software developers' preferences of process metrics concerning their characteristics and their defense industry compatibility. Consequently, we assessed the metrics that might likely integrate into the Scrum development using the Analytic Hierarchy Process. The results indicated that the Actionable Agile Addon scored the highest, followed by the Screenful Addon. Ultimately, this contemporary study presented a novel approach that has increased individuals' participation in metric planning, implementation, and monitoring, therefore moving towards more achievable software development goals.

KEYWORDS:
Software Measurement Component, Software Process Metrics Tool, Scrum, AHP, Software Component Selection, Industrial Case Study.

1 | INTRODUCTION

Recently, the notion of quality has become crucial across all engineering disciplines, and can be assessed through various forms of measurement. Measurement is a process of objective association by assigning elements from number or symbol sets to the real-world properties of an entity [1]. In other words, measurement is the objective representation of the experimental knowledge of a real-world being. Fenton and Bieman [2] defined measurement as "*the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way so as to describe them according to clearly defined rules*." (p. 5). Measurement is a crucial element to all scientific and engineering activity. Therefore, software engineering is no exception as it is essentially an engineering-based discipline. According to Pfleeger [3], 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 [4] and performance of the product [5]. Today, performance measurement has become a key feature in the development of successful software engineering applications [6], and productivity [7].

The software development process has a more abstract structure than other engineering activities. It can be considered as a social activity with software measurement defined as an approach used to control, manage, monitor and improve the software development process [8]. Software measurement can be divided into direct and indirect measurements. Direct measurement refers to values of the internal attributes of a software product such as its cost, speed, and its requirements for both effort and memory. Indirect measurement, however, refers to values of the external attributes of a software product such as its functionality, complexity, and reliability. These measurement parameters can be used in order to create meaningful metrics. In this way, a software metrics ecosystem can be structured so as to provide determination, prediction and the improvement of quality of the related product or process.

Pfleeger and Fitzgerald [9] suggested that measurement data of the middle product can be applied in order to better understand the quality of the final product. In addition, certain measurements are known to be associated with the software development process. Thus, they are expressed as controlling activities related to the process and quality of a product that evaluates the maturity of the process.

The combination of data obtained by the measurement with the useful information reveals the metric. In this way, objective measurements are converted to an interpretable form. In practice, the terms of *metric* and *measure* can be utilized interchangeably. Although their definitions may overlap in places, a metric is attained by one or more measurements together with other information. On the other hand, a measurement is a value which is assigned to an entity [10]. However, metrics can be much more complex, as they provide a greater level of information about the entity being measured [3].

The SPI manifesto suggests creating a learning organization [11]. 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 [12]. 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 [13]. 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 [5] 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 [14]. Software development perspective includes process metrics, product metrics, test metrics, maintenance metrics, and subjective metrics [6]. However, Lee and Chang [5] pointed out that a metric can be included in one or more of these categories.

Moreover, in order for software process metrics to exist, methodologies concerning the software development lifecycle should be both defined and applied. Traditional methodologies have been replaced by modern methodologies in today's software development processes [15]. Agile methodologies pay special attention to quality because the ultimate purpose is to deliver high quality software to the end-user [16]. Research over the past decade has shown that organizations which develop software need to regulate agile approaches according to their requirements. Scrum is one such modern agile software development framework that is widely used and known [17]. An 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 [18]. 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 [19].

In addition to the aforementioned explanation, Scrum metrics can be defined as a "focus on the predictable delivery of working software to customers". Scrum metrics can help to observe the efforts expended for the development of quality software. In addition, these metrics can be used to evaluate and visualize the results of these efforts. There are several defect and bug measurements within Scrum such as the escaped defects metric measure and how many bugs were discovered in production. Scrum methodology consists of iterative and incremental sprint structures, 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 "done" definition of a Scrum team is that the new feature of the product has been developed, tested, integrated, and documented [20]. 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 [21]. Sprint's success rate is an important starting point for adaptation and inspection.

The Scrum process evaluation metrics are listed and explained in Table 1.

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. Various software process metrics have been proposed over time. Commonly used process metrics may be integrated into supporting process tools and later adapted for use in specific organizations. 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 various metrics add-ons available on the market. Their content, functionality, technical specifications, websites, usage training, and demonstrations are also available in the market.

Scrum Process Evaluation Metrics	Definition
Burnup Chart	Monitors the progress of the team. The Burnup Chart shows a comparison between completed works and the targeted finish.
Velocity Chart	The total number of values transferred in each sprint are shown in the Velocity Chart. This allows conjecture as to the amount of a job the team can accomplish in future sprints. The chart also helps decisions to be made regarding how much to feasibly commit, and can therefore be useful during sprint planning meetings.
Burndown Chart	Total work progress is traced by the Burndown Chart, enabling assessment of whether or not a sprint was successful. Teams can self-manage their progress, and then respond accordingly.
Sprint Report	Issues in each sprint are shown in the Sprint Report. Checking mid-sprint progress can be made easier, and can also be used in sprint retrospective meetings.
Control Chart	The cycle or lead time is represented for the product, version, or sprint by the Control Chart. Creating cycle time needs time spent by each topic in a certain status. The chart includes the average, rolling average, and standard deviation for this data.
Cumulative Flow Diagram	Stability of the flow is shown in the Cumulative Flow Diagram. Also, the focus point which makes the process more deductive can be made more comprehensible. Quantitative and qualitative insight into both past and present problems can be easily seen, and almost limitless data can be visualized.
Epic Report	The list of completed, incomplete, and imponderable issues is shown by the Epic Report, which can be beneficial for the planning of sprints.
Epic Burndown Chart	The team's progress against the work for an epic is expressed by the Epic Burndown Chart. Largescale user stories can be broken down into a number of smaller stories and defined as an epic. Data based on the estimation statistic utilized in the board is expressed in the chart.
Version Report	The team's progress directed according to completion of a version is shown in the Version Report. The report can also provide the predicted release date based on the team's average rate of progress.
Release Burndown Report	The team's progress towards work for a release can be shown by the Release Burndown Report.
Feature and Epic Progress	Shows the relative progress and size of properties within a project.
Multi-dimensional Backlog	Backlog of the project can be recognized clearly through the Multi-dimensional Backlog, which also attempts to engage stakeholders in the backlog.
Backlog Map Estimate Accuracy	The Backlog Map attempts to engage stakeholders in the backlog, and can be utilized to review spent effort and future focus. The team's estimating performance can be seen using the Estimated Accuracy; helping them to improve it where necessary.
Requirements Readiness	The focus on requirements can be improved by the Requirements Readiness. It can also help the team to evaluate the distribution of work packets (past, current, or those still in the backlog).
Potentially Deliverable Scope	The time needed for a team to complete the backlog by a certain date can be shown using the Potentially Deliverable Scope.
WIP (Work in Progress) by Team Members	Team members' workloads are shown by the WIP, which indicates any overburdening of team members.
Landing Zone Story	Movements of the end dates over time can be easily shown using the Landing Zone Chart as animation.
Time Between Events	The average spent time for a combination of events can be tracked using the Time Between Events. Supplementary details regarding WIP distribution can also be revealed.
Team Status	Team members' current work and what has been completed can be tracked with the help of the Team Status.
Track Lead and Cycle Times	This allows for tracking of the lead and cycle times. It also shows the task's spent times for each individual state in the workflow.
Task Status (Bird's-Eye View)	The open and completed tasks can be better viewed using the Task Status Screen. The slicing and dicing of tasks is also easily achievable using this screen.

Source: https://www.atlassian.com/.

TABLE 1 Scrum Evaluation Metrics.

The selection of software metrics and the choice of tools to present metrics can be a significantly difficult and complex problem. Therefore, a systematic process is required for the selection of the metrics toolkit. Multi-criteria decision-making methods can be evaluated by taking into consideration various qualitative and quantitative criteria or purposes. They aim to achieve the best compromise solution by examining the existing alternatives according to the determined criteria. As a result of multi-criteria decision-making methods, the decision maker can sort, group, or make choices between existing alternatives.

The Analytic Hierarchy Process (AHP) is a multi-criteria decision-making method for complicated and unstructured problems [22]. In particular, it is a convenient approach to represent levels of goals, potential criteria (or sub-criteria) based on a set of alternatives. There are various reasons to select AHP as the primary process. What is essential for us to recognize here is that AHP can guide us to consider the alternatives and ultimately find the best choice. Therefore, it is important to stress that it certainly provides a very flexible model for a given problem with the preferred level of details [23]. Most importantly, however, a proposed approach based on AHP may rely on experts from different backgrounds. Perhaps one expected difference of software development is that it may expect contributions from practitioners from different knowledge levels and even domains or backgrounds. Therefore, it is crucial to assess the elasticity of the final decision and the consistency of the decision maker‘s judgments [24]. Finally, teams can benefit from available software to apply AHP fast and effectively.

The current study proposes an approach to select a process measurement component, which includes the appropriate software development process metrics using the AHP approach. In addition, we consider software project requirements and software developers' contributions during the selection process.

In the previous component of the research [25], a set of add-ons which include the software process metrics was already highlighted. The purpose of the current component of the study is to extend the expert pool of the study and rigorously revisit the add-on selection process that offer software development process metrics based on the AHP method. To this end, the weighting of selection criteria and the evaluation of alternatives were reassessed as a part of the industrial case study in the TÜBİTAK SAGE software development group.

The overall structure of the study takes the form of six sections. The next section has attempted to provide a review of the literature. Third section describes the procedures and methods used in this investigation. The fourth section discusses and summarizes the results. Finally, sixth section concludes the study by discussing recent findings.

2 | BACKGROUND AND RELATED WORKS

The notion of the Agile philosophy and its practices embrace the social aspects of software development [26]. While the Scrum methodology is based on iterative and incremental development, and concrete organizational roles [27], a suitable measurement approach reveals the level of productivity in a target sprint. In particular, agile development promises that its time-boxes present a new functionality. The goal of a metric is to assist software practitioners in monitoring the planned product's progression. In particular, it measures the sprint progress and the level of productivity of a development team. The research has also shown that Sprint's success rate is a crucial outlet for adaptation and inspection [28].

2.1 | Importance of Choosing the Right Metrics

Previous research has established that software requirements, the software development process, and maintainability are important factors that should be focused on during the software metric selection process [9]. Consequently, tailoring and management of a metric toolkit can be exhausting. In a previous study [29] on the cost of metrics in the project, different variables have been found to be related to data collection and analysis in the calculation of metrics increased the project cost by 7-8%. Therefore, accurate data collection is vitally important to avoid the cost of unnecessary metrics.

Although exploring software measurement at the organizational level provides important contributions to understanding the concepts and terminology used in this field, it has been argued that more than 80% of software measurement was found unsuccessful within the eighteen months [30]. The reason for this is likely to be considered as the difficulty in understanding and utilizing software metrics. Fenton and Neil [31] reported that many of the proposed metrics today are not used effectively within the decision-making process due to the fact that they are not found reliable.

2.2 | Importance of Software Process Metrics

To further examine the importance of the software process metrics, Kitchenham et al. [32] hypothesized that more rigorous studies were essential for adequately measuring the software processes. Moreover, it has been reported that measuring the progress in the software process is crucial for changing the future of a product [3], which requires an active software quality team that should guide software development teams and help them to map software metrics to organizational goals [33].

Previous research has established that measurement and competition are vital components that boost individuals to improve their performance in an organization [34]. Most importunately, however, measurements certainly help us to assess project performance, and give information about the state of an organization compared to some previous progress [8].

Pfleeger and Fitzgerald [9] were one of the first researchers to highlight the need for software metrics toolkit. In particular, it has been reported that it is important to address three main concerns (i) maturity of the software development process; (ii) availability of measurement data; and, (iii) the project management requirements when choosing metrics for a software project development project. Therefore, it is important to mention that a metrics toolkit should not determine by the middle management. A software team as a whole should progress the selection process. A more substantial approach to the evaluation process suggested that the selection process should be finished prior to starting the software process improvement activities [13]. Paulish and Cartelon [13] reported that the evaluation is the right approach to address priorities and ultimately to build a productive consensus within an organization. Concurrently, Ebert et al. [8] claimed that metric designers should receive continuous feedback from metrics users. They reported that metrics should encompass crucial information about software measurement techniques.

2.3 | Analytic Hierarchy Process (AHP) Framework

The Analytic Hierarchy Process (AHP) approach is a decision-making technique for organizing and analyzing complex decisions by integrating various criteria into a hierarchical structure that allows us to evaluate different alternatives [23]. To the best of our knowledge, there have been a limited number of multi-criteria decision-making studies, especially concerning software metrics. To choose a software suitable to the project management process, Sharma et al. [35] utilized the AHP approach. In addition, a fuzzy version of the AHP method was selected to single out plugins where the decision-making criteria were based on a set of quality parameters, including but not limited to adaptability, availability, complexity, and understandability.

Sharma et al.[35] 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.[36] proposed the fuzzy AHP method for selecting the most reusable software component such as plugins. They determined criteria that pointed to reusability as adaptability, availability, interface complexity, customizability, and understandability by reviewing the related literature. Ömürbek et al.[37], the aim was to select a project management program which could be used in software development based on using AHP and the TOPSIS methods.

Sureshchandar and Leisten[38] presented an AHP framework to prioritize the criteria of metrics with respect to the three categories. The objective of a study is to define the criteria and critically evaluate the metrics. They defined robustness, simplicity, and also cost-effectiveness as important criteria for the process metric. Ahmad and Laplante[39] introduced an application of the AHP method in the selection of an appropriate software project management tool. Garg et al.[40] presented a framework for the ranking of software engineering metrics based on expert opinion with the help of the fuzzy-based matrix method. The aim of executing this factual research was to improve the comprehension of software engineering metrics that may have an effect on software reliability, and to examine the importance of its influence. Also, existing software engineering metrics were ranked systematically according to their effect on the prediction of software reliability. Li et al.[41] proposed a meticulous application of the AHP and expert opinion in choosing software reliability metrics, with relevance, experience, correctness, practicality and feasibility as the criteria determined in their study. Pandey et al.[42] explained the relationship between attributes of particular metrics with empirical approaches. Choosing the most significant attributes based on their weight values, the AHP method was used in order to help decrease the dimensionality of metrics. Cost and schedule optimization in the software development process can be achieved through dimensionality reduction. AHP was used for a continuous quality improvement program at the Latrobe Steel Company. According to the opinions of Latrobe Steel Company's experts[43], the AHP model was developed as a hierarchical cause-and-effect in order to centralize on the areas that needed the most control and process improvement.

Finnie et al.[44] underlined several factors involved in software productivity. They identified the relationships of these factors by using the AHP technique to prioritize software productivity factors. Sharma et al.[45] presented an experiment that used a real-life sample in evaluating components in terms of overall quality. The result of the study demonstrated that comparing and selecting the most suitable component can be realized with all desired quality characteristics through application of the AHP method. Ömürbek and Şimşek[22] determined why an online shopping site was preferred by instructors by using the AHP method. In addition, the importance of the features offered by online shopping were demonstrated by using the AHP method, according to the faculty members.

There is a growing body of literature that are using multi-criteria decision-making methodologies (e.g., AHP) related to the selection of software metrics and their tools. Sharma et al.[35] used the AHP method, which is one of the multi-criteria decision-making method, for selecting software related to the project management process. Other alternatives were specified as HP-PPM, Microsoft Project, and Oracle Primavera. The criteria of the selection process was determined by reviewing the literature and interviewing experts, and included cost, ease of use, maturity level, vendor, and consultant support. In their study, selected criteria were assessed by interviewing five project managers with regards to the selection from three project management software tools. Microsoft Excel was used in conducting the AHP process. According to the AHP selection process, HP-PPM was selected as the most appropriate software management tool.

Also, Sagar et al.[36] 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.

In addition, six software component alternatives were specified. Java programming language was used to implement the fuzzy AHP process. As a result, AVG Antivirus software component was selected as the most reusable component. In the study of Ömürbek et al.[37], the aim was to select a project management program which could be used in software development based on using AHP method. Their study tried to determine which institutional project management program software development processes should be applied within the Computer Center of a university using the AHP method. In this context, four programs were evaluated according to 13 different criteria. Expert opinions and classifications in the literature were utilized in the determination of the evaluation criteria. These criteria were supplier firm and purchasing, ease of use, adaptation, technical infrastructure and support. Within the framework of these criteria; Atlassian-Jira, HP-PPM, IBM Rational Request, and Microsoft TFS were evaluated as alternative tools. The weighting of the criteria using the AHP method were determined using Microsoft Excel.

In a study by Al-Qutaish et al.[46], 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.[47] 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[48]. 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.

To determine how well a multi-criteria decision method suits the needs of an organization, we can utilize mechanisms that use hierarchic or network structures. There are a set of approaches to representing a decision problem and consequently define measurable goals that have appeared in the literature, such as the (i) Quality Function Deployment approach [49], which is a multi-level evaluation framework for product risk analysis, (ii) the Goal Question Metric approach [50], which is for defining and interpreting operational and measurable software by refining a set of questions into metrics, (iii) and the Software Quality Metrics approach [51] propose a formula to align quality metrics to ensure all products and the source codes are in high quality. However, neither of these mechanisms provides the same framework characteristics by quantifying its criteria and alternative options (e.g., priority scores of the elements). This capability distinguishes the AHP from other decision-making techniques.

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. To ensure the validity of the findings, we visualized the data and utilized descriptive statistics. A reason for this is that we should conduct the survey only with the software team. To help mitigate the effect of the relatively the small sample size, we added semi-structured questions to optimize the study features by focusing on the most critical parameters.

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.

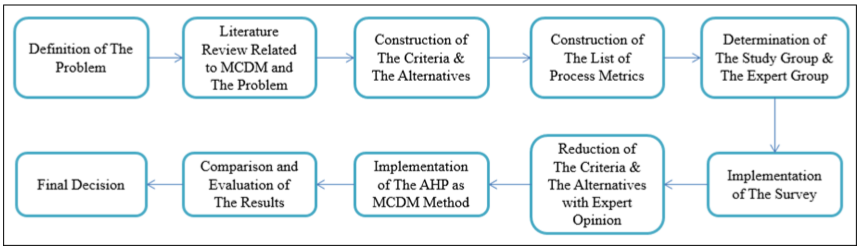

FIGURE 1 Case study process steps

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.

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 2 Software process metrics add-ons evaluation criteria

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.

The survey 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. The analytically processable format of the collected data was converted to graphical form as pie charts and bar graphs using Google Sheets and Microsoft Excel. The aim was that the expert group would be influenced by the pie charts and histograms in determining the criteria related to the AHP method in evaluating the alternatives. 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.

While some of these characteristics were found to be primitive, and certain products only offered a few metrics, some offered a comprehensive process metrics service. Considering the suggestions of the developers, an advanced add-on product that included metrics beyond that of the nine core process metrics was deemed preferable.

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 2)

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.

Expert	Experience (years)	Project role
Expert 1	16	Coordinator
Expert 2	6	Project Team Leader
Expert 3	19	Software Test Unit Chief
Expert 4	13	Software Simulation and Mission Planning Unit Chief
Expert 5	13	Embedded System Unit Chief
Expert 6	15	Software Architecture Design Unit Chief

TABLE 3 Expert Group Information

Table 3) presents the experience and project roles of the expert group. The group consisted of a Software Development Group Coordinator, four Unit Chiefs from software development units, and one Project Team Leader.

3.1 | Multi-Criteria Decision-Making and the AHP Method

Decision-making is a phenomenon which occurs throughout life. Generally, decision-making is an assessment of the most appropriate option or a set of alternatives which depend on a metric and are oriented to at least one purpose inside a cluster of alternatives. This process includes the decision maker, alternatives, criteria, environmental factors, the needs of the decision maker, and the results of the decision. The decision-making process ends with the selection, ordering or classification between existing alternatives. In the current study, multi-criteria decision-making method was employed to identify the correct decision [52]. Multi-criteria decision-making (MCDM) is a decision process which allows the use of at least two criteria from within a cluster drawn from a countable (finite) or uncountable set of alternatives.

The Analytic Hierarchy Process (AHP) is a method based on basic math and psychology and which was founded by Thomas L. Saaty in 1980 in order to solve complex economic, social and technical problems [23]. AHP is an intuitive and logical scrutinized approach to decision problems between elements based on complex relationships. It provides a simplified method to assess hierarchical structure [53], and is dependent on inferior criteria affecting superior criteria. Therefore, the degree of effect between criteria needs to be determined [54]

The steps of the AHP method are as follows [24].

Step 1: Definition of decision problem

The first step is to assess that the decision-making problem has been expressed clearly. The problem should be determined as appropriate or inappropriate to AHP based on a literature review, experience, and expert opinion. After determining the suitability of a problem to AHP, it can be divided into sub-problems. Then, each sub-problem can be resolved in order that an overall solution can be obtained.

Step 2: Creation of hierarchy structure

Hierarchy is represented by at least three levels in AHP. The goal, or purpose, is situated at the top level of the hierarchy. In the middle level are the main (primary) criteria, followed by any secondary criteria under the main criteria. At the lowest level are the alternatives. A three-level hierarchical structure is illustrated as shown in Figure 2.

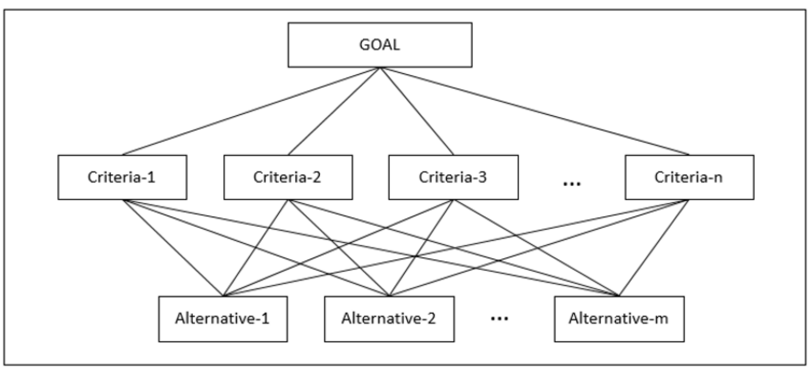

FIGURE 2 AHP hierarchy structure

Step 3: Forming the matrices of pairwise comparison of criteria

Matrices of pairwise comparison is a significant step for AHP after creating the hierarchical structure. A score scale should be used as proposed by Saaty (see Figure 3) in forming these matrices.

Intensity of importance	Definition	Explanation
1	Equal importance	Two activities contribute equally to the objective
3	Moderate importance	Experience and judgement slightly favour one over another
5	Strong importance	Experience and judgement strongly favour one over another
7	Very strong importance	An activity is strongly favoured, and its dominance is demonstrated in practice
9	Absolute importance	The importance of one over another affirmed on the highest possible order
2,4,6,8	Intermediate values	Used to represent compromise between the priorities listed above
Reciprocals of above non-zero numbers		if activity i has one of the above non-zero numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with i

FIGURE 3 Saaty Importance Scale

Criteria should be compared in order to determine the degree of importance of each criteria to be used when comparing alternatives. Comparisons should be performed by experts. According to the scoring scale (see Figure 3), comparison matrices are then formed by making pairwise comparisons between sets of criteria.

Step 4: Calculation of priority vector (W)

After forming the matrix of pairwise comparison, the priority vector (W) is then calculated. This shows the priority of each element in the matrix relative to the other elements. In order to normalize a matrix of pairwise comparison, each element is divided by summation of its column.

Step 5: Calculation of λmax and consistency index (CI)

Consistency index (CI) should be calculated by the decision maker in order to establish that the matrix of pairwise comparison, which is formed from values determined by comparison between criteria, is deemed to be consistent or inconsistent.

Step 6: Calculation consistency rate (CR)

The value of the random consistency index (RI) is required in order to calculate the consistency rate (CR).

Platform: Super Decisions software (version 2.10.0) was employed in order to implement the AHP method in the current study. This software provides the calculated weights and compares pairwise alternatives and criteria.

Information related to the process metrics was collected by way of the survey method, applied to the software developers. The survey provided all the necessary information without participants being affected by each other, and then their preferences were grouped. The structure of the study and the expert group was defined. The results of this industrial case study are presented in the next section. Results are also visualized through the use of graphs so as to aid better understanding of the findings. The findings are presented along with comments to support the graphical results.

4 | RESULTS

Twenty eight software practitioners who work in four different units namely Embedded Software (n=10), Software Simulation (n=7), Software Testing (n=6), and Software Architectural Design (n=5) were chosen as study group members for the study (see Figure 4 shows the working units of the study group members).

The results of the selection process identified in participants responses are summarized in Figure 5. Looking at Figure 5, it is apparent that Cumulative Flow Diagram was the most (selected by 15) software development process metric among the all other options. Not surprisingly perhaps,

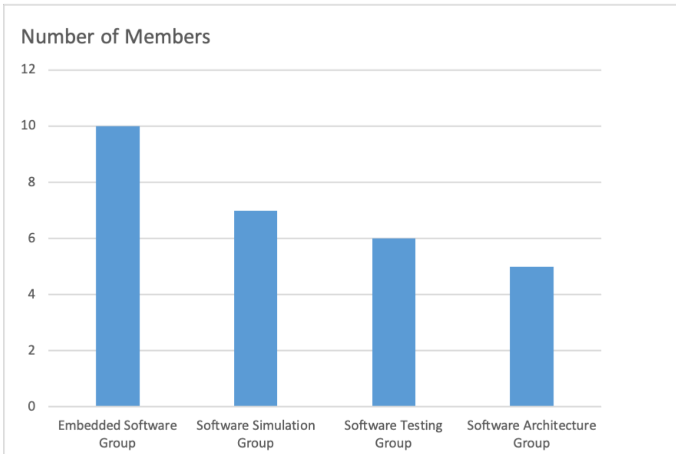

FIGURE 4 Study group by working unit

Most software development process measurement components were observed to contain the Cumulative Flow Diagram and Burndown Chart metric options. In fact, the majority of the selections made by the study group can be covered by the software development process metric providers. The three process metric options namely Burndown Chart, Velocity Chart, and Burnup Chart were selected more than 10 of the participants. It is visible from this figure that options Team Status, Estimate Accuracy, Control Chart and Sprint Report were selected by 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.

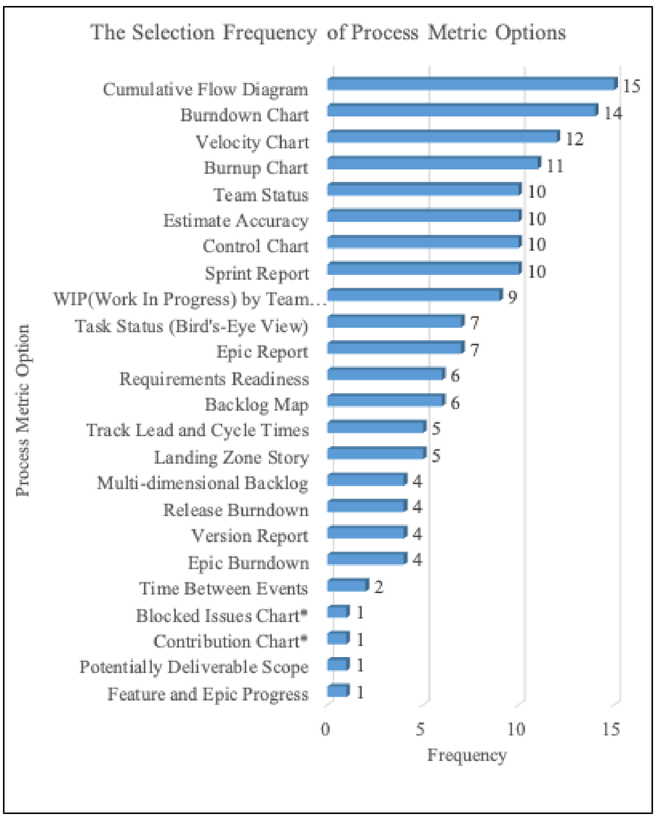

FIGURE 5 Selection frequency of process metric options (adopted from [25]).

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 Agile[1], (ii) SenseAdapt Agile[2], (iii) Screenful[3], (iv) Predictable Agile[4] with the determined criteia (1) Feasibility & Usability, (2) Functionality, (3)Relevance, and (4) Experience.

After completion of the criteria evaluation by the expert group, Cronbach's alpha value was calculated on the results of the evaluative process. MedCalc[5] , which is a statistical calculation software, was used to measure the internal consistency of the results. The internal consistency value was obtained as .81. The mean of the value represents a good level of internal consistency. Also, the error variance of the evaluation results was calculated as being .35. According to Cronbach's alpha value and error variance, which was obtained by performing different calculations, the evaluation process of the study that took place with the expert group, and particularly that of the reduction process, was found to be satisfactory in terms of its reliability. Therefore, the proposed selection method can be generalized to other problem domains to a level of statistical confidence.

The hierarchical structure was established between the criteria and the alternatives. Figure 6 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.

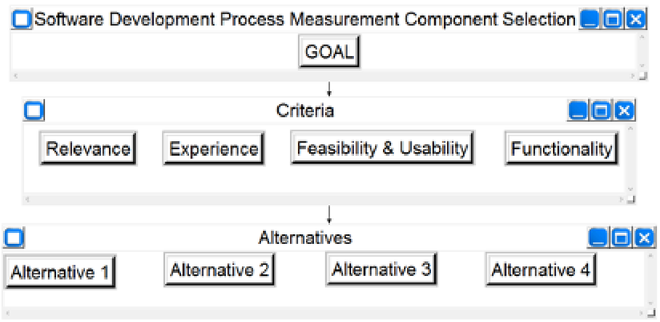

FIGURE 6 AHP hierarchical structure of component selection

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.

While selecting project management tools in the literature, plug-ins were selected as a software component that could strengthen the process management tool according to the current study. The AHP approach was also considered applicable to the current study, together with the studies of Sureshchandar and Leisten [38], Sharma et al. (2015), Sagar et al. [35], Ömürbek et al. [37], and Al-Qutaish et al. [46]. In the current study, the criterion of Usability was based on Ease of Use, as determined by Sharma et al. [35]; the Experience criterion was based on Likely Understandability, as determined by Sagar et al. [36]; and the Maturity Level criterion was based on the study of Sharma et al. [35]. The Functionality criterion was determined by the expert group based on the Technical Infrastructure criterion, as determined by Ömürbek et al. [37], and the Functionality criterion, as determined by the works of Al-Qutaish et al. [46] and Zaidan et al. [47]. Unlike these other studies, the current study considered that the high prioritization of the

[1] https://actionableagile.com/
[2] https://ripplerock.atlassian.net/wiki/spaces/SEN/overview
[3] https://screenful.com/tour
[4] http://agilemontecarlo.com/
[5] https://www.medcalc.org/

Relevance criterion has a likely effect on study group cooperation. In contrast to the other studies in the literature, the contribution of the study group was highlighted within the evaluation process of the criterion pool by the expert group in the current study.

In this industrial case study, a strategic planning tool was constructed to explore the potential criteria for selecting software development process metrics. The objective of this study was to present a tailored process where a saturation on the multi-criteria decision-making can be achieved, and thereby metric selection can be conducted more effectively. Due to the iterative nature of the study structure, this was done by a decision analysis method (AHP). Cumulative Flow diagrams and Burndown Chart were found to be the most preferable among all the proposed metrics, which concurrently provided by nearly all candidate process measurement components. To evaluate the criteria in each iteration, we added more experts to review the landscape. The experts selected for this study confirmed that the software practitioner's preferences were relevant. However, there were some contradicting ideas reported by the experts. Firstly, the expert had a disagreement on the computational requirement when using AHP, even for a small problem. In addition, AHP was reported to be based on probability and possibility measures. Consequently, an expert reported that the approach might have a subjective nature, highlighting that the decision cannot be guaranteed to be perfect. Lastly, a reviewer confirmed that if the level hierarchy in our model increases, the system complexity increases drastically, i.e., the number of pair comparisons would ultimately increase.

4.1 | Threats to Validity

Yilmaz[12] described "threats to validity" as one of the leading factors that can decrease the usefulness, trustability, and correctness of a study. The following methods used for mitigating threats to validity are discussed under the following headings as Internal Validity, External Validity, and Construct Validity.

Internal Validity: It was shown that the expert group properly completed this process in terms of its validity, with adequate time and opportunity afforded to them in order to make their selections. In order to prevent the participants of the study group from being influenced by each other during the interpretation of the survey results, separate interviews were conducted in separate environments. Throughout the study process, the measurement instrument remained unchanged.

External Validity: Throughout the study process, the participants and the study setting, such as the software unit, remain unchanged. The study group which was determined in the initial phase of the experiment was kept the same through all steps of the process. Only professional software developers were selected as members of the study group. In addition, only those with at least 1 year of professional experience were included in the study. The study was conducted using a survey which included multiple answering options. Metric preferences were not only performed using multiple-choice question types. The study group was able to make their own suggestions through an open-ended part of the survey.

The AHP approach was not applied with only one single expert. The opinions of six different experts were collected, as well as criteria in the literature evaluated in their determination. The selection of the criteria was based on majority opinion among the study group and expert group. Evaluation of the criteria and the alternatives by all members of the expert group increased the reliability of the process. In other words, the same method was applied by more than one expert. A more reliable result was also ensured by calculating and applying the mean result across the whole group of experts.

Construct Validity: Qualitatively, the evaluations made by the researcher and of the study's participants were checked to see whether or not they shared similarities in their interpretations of the survey results. Quantitatively, the internal consistency of the assessment was checked using Cronbach's alpha in order to measure the reliability of the criteria reduction operation.

5 | CONCLUSIONS

A systematic process was followed in the present study for the selection of software development process measurement components, which is generally considered to be a complex issue. In this context, a selection approach in which the software team was included in the selection process was proposed, together with collaboration across the team. This approach was taken in accordance with the characteristic structure and requirements of the company, the projects being undertaken, and the software development team in place. A systematic process was followed in order to achieve more objective and accurate decisions. As a result, the software development process measurement component selection performed was deemed to be both appropriate and suitable.

This study used an industrial qualitative case study approach based on the AHP method, which was adequate for dealing with decision-making situations, especially with uncertainties within the institution, the projects, and the software development team proposed metrics. The method of AHP improves the strategic planning processes, which can also be considered as a tool for communication, especially where multiple decision-makers are involved. The study results demonstrated the importance of the Relevance criterion discovered as the most serious concern by the software development team. Due to its simplicity and ability to deal with qualitative and quantitative criteria, it is well-suited to deal with the factors considered by the study group's members.

Consequently, the expert group assessed the collected results of the process metrics. We confirmed that the decision-making expert group was inclined to highlight the importance of the Relevance criterion. However, during the second iteration, it was observed that experts agreed to accept the second priority as the Experience criterion. Furthermore, it was documented that group members voted equally for the Functionality and the Feasibility and Usability criteria.

According to our experiences from this study, the use of AHP analysis was promising. Making comparisons forces the expert group to reexamine the weights of the factors and analyze the situation more accurately and with increased intensity. In light of this remark, a rigorous process to assess available alternatives was determined. For example, Alternative-1 was prioritized as the most preferred with 40.26%. However, it was previously observed that much managerial decision-making is based on subjective judgments instead of the outcomes of systematic planning. Also, it is important to note that expanding the presented formulation to cover a wider range of expert groups to introduce their ideas could benefit the planning process.

The process of evaluation by the expert group was shaped based on human perception, which may initially have difficulty in focusing on discrete evaluation and discrete criteria. In the alternative evaluation process, alternatives are usually reviewed, and dominant alternatives are determined for one or more reasons. In the current study, it is considered that the expert group evaluated the criteria pool based on their perceptions in a holistic way, or that the focus on a single alternative for a particular reason was broken. For this reason, a selection process was conducted by considering the problems and requirements of the study group during the implementation process.

The present study was designed to determine the effect of the multi-criteria decision-making process, mainly using the AHP method to select a set of software metrics components suitable for a software domain. These results suggest that the contribution of the software development team directly supports selecting an efficient set of software metrics components. Although the current study is based on a small sample of participants, the findings suggest that using relevant software metric components would undoubtedly improve software development productivity in the near future. With a small sample size, caution must be applied, as the findings might not be transferable to other Scrum landscapes.

Although the study has successfully served as an example of the AHP method being preferred when systematic decision-making is required in the field of software development, it has certain limitations in terms of size. The present investigation was specifically designed to evaluate factors derived from a single software company. Therefore, these results may not be applicable to all situations. Further research might explore the applicability of the AHP method in software planning. This would result in new alternative resource allocation approaches from practitioners' viewpoints and likely to include more diversity in the planning process.

References

1. Ferris TL. A new definition of measurement. *Measurement* 2004; 36(1): 101–109.
2. Fenton N, Bieman J. *Software metrics: a rigorous and practical approach*. CRC press . 2014.
3. Pfleeger SL. Software metrics: Progress after 25 years?. *IEEE Software* 2008; 25(6): 32–34.
4. Yilmaz M. Observed Effects of Software Processes Change in Three Software Firms: Industrial Exploratory Case Study. *Pamukkale Univ Muh Bilim Derg* 2019; 25(2): 240-246.
5. Lee MC, Chang T. Software measurement and software metrics in software quality. *International Journal of Software Engineering and Its Applications* 2013; 7(4): 15–34.
6. Farooq SU, Quadri S, Ahmad N. Software measurements and metrics: Role in effective software testing. *International Journal of Engineering Science and Technology* 2011; 3(1): 671–680.
7. Yilmaz M, O'Connor RV, Clarke P. Effective Social Productivity Measurements during Software Development—An Empirical Study. *International Journal of Software Engineering and Knowledge Engineering* 2016; 26(03): 457–490. doi: 10.1142/s0218194016500194
8. 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.
9. Pfleeger SL, Fitzgerald Jr J. Software metrics tool kit: support for selection, collection and analysis. *Information and software technology* 1991; 33(7): 477–482.

10. Saraiva RM, Perkusich M, Almeida HO, Perkusich A. A Process to Calculate the Uncertainty of Software Metrics-based Models Using Bayesian Networks.. In: SEKE. ; 2017: 467–472.

11. Manifesto . *Software Process Improvement (SPI)*. Lero; Dublin: Last accessed September 3, 2019. https://2019.eurospi.net/images/eurospi/spi_manifesto.pdf.

12. Yilmaz M. *A software process engineering approach to understanding software productivity and team personality characteristics: an empirical investigation*. PhD thesis. Dublin City University, DCU; 2013.

13. Paulish DJ, Carleton AD. Case studies of software-process-improvement measurement. *Computer* 1994; 27(9): 50–57.

14. 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)* 2012; 3(2): 40–62.

15. Clarke P, O'Connor RV, Yilmaz M. In Search of the Origins and Enduring Impact of Agile Software Development. In: ICSSP '18. ACM. Association for Computing Machinery; 2018; New York, NY, USA: 142–146

16. Highsmith J. *Agile project management: creating innovative products*. Pearson education . 2009.

17. Kuhrmann M, Tell P, Hebig R, et al. What makes agile software development agile. *IEEE Transactions on Software Engineering* 2021.

18. Yilmaz M, O'Connor RV. A Scrumban integrated gamification approach to guide software process improvement: a Turkish case study. *Tehnički vjesnik* 2016; 23(1): 237–245.

19. Jones C. *A guide to selecting software measures and metrics*. CRC Press . 2017.

20. Layton MC, Ostermiller SJ, Kynaston DJ. *Agile project management for dummies*. John Wiley & Sons . 2020.

21. Schwaber K, Beedle M. *Agile software development with Scrum*. 1. Prentice Hall Upper Saddle River . 2002.

22. Ömürbek N, Şimşek A. Analitik hiyerarşi süreci ve analitik ağ süreci yöntemleri ile online alişveriş site seçimi. *Yönetim ve Ekonomi Araştırmaları Dergisi* 2014; 12(22): 306–327.

23. Saaty TL. What is the analytic hierarchy process?. In: Springer. 1988 (pp. 109–121).

24. Saaty TL. *Theory and applications of the analytic network process: decision making with benefits, opportunities, costs, and risks*. RWS publications . 2005.

25. Tekin N, Kosa M, Yilmaz M, Clarke P, Garousi V. Visualization, Monitoring and Control Techniques for Use in Scrum Software Development: An Analytic Hierarchy Process Approach. In: Yilmaz M, Niemann J, Clarke P, Messnarz R., eds. *Systems, Software and Services Process Improvement*Springer. Springer International Publishing; 2020; Cham: 45–57.

26. Yilmaz M, Atasoy B, O'Connor RV, Martens JB, Clarke P. Software Developer's Journey. In: Kreiner C, O'Connor RV, Poth A, Messnarz R., eds. *Systems, Software and Services Process Improvement*Springer. Springer International Publishing; 2016; Cham: 203–211.

27. Yilmaz M, O'Connor R, Clarke P. A systematic approach to the comparison of roles in the software development processes. In: . 290. Lero. Springer-Verlag; 2012: 198–209

28. Wan J, Zhu Y, Zeng M, others . Case study on critical success factors of running Scrum. *Journal of Software Engineering and Applications* 2013; 6(2): 59.

29. Card DN, Glass RL. *Measuring software design quality*. Prentice-Hall, Inc. . 1990.

30. Wallace LG, Sheetz SD. The adoption of software measures: A technology acceptance model (TAM) perspective. *Information & Management* 2014; 51(2): 249–259.

31. Fenton NE, Neil M. Software metrics: roadmap. In: IEEE. ; 2000: 357–370.

32. Kitchenham B, Pfleeger SL, Fenton N. Towards a framework for software measurement validation. *IEEE Transactions on software Engineering* 1995; 21(12): 929–944.

33. Akarsu Z, Ozgun M, Kuru Y, Yilmaz M. Using Adapted Version of Hoshin Matrix for Selection of Agile Software Development Processes. In: UYMS. ; 2018: 1-11.

34. Marks G, O RV, Yilmaz M, Clarke PM. An ISO/IEC 12207 Perspective on Software Development Process Adaptation.. *Software Quality Professional* 2018; 20: 48-58.

35. Sharma NK, 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)* 2015; 5(3): 188–195.

36. Sagar S, Mathur P, Sharma A. Multi-criteria selection of software components using fuzzy-AHP approach. *International Journal of Innovative Computing Information and Control* 2015; 11(3): 1045–1058.

37. ÖMÜRBEK N, MAKAS Y, ÖMÜRBEK V. AHP VE TOPSIS YÖNTEMLERİ İLE KURUMSAL PROJE YÖNETİM YAZILIMI SEÇİMİ. *Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü Dergisi* 2015(21): 59–83.

38. Sureshchandar G, Leisten R. A framework for evaluating the criticality of software metrics: an analytic hierarchy process (AHP) approach. *Measuring Business Excellence* 2006.

39. Ahmad N, Laplante PA. Software project management tools: making a practical decision using AHP. In: IEEE. ; 2006: 76–84.

40. Garg R, Sharma K, Nagpal C, Garg R, Garg R, Kumar R. Ranking of software engineering metrics by fuzzy-based matrix methodology. *Software Testing, Verification and Reliability* 2013; 23(2): 149–168.

41. Li H, Lu M, Li Q. Software reliability metrics selecting method based on analytic hierarchy process. In: IEEE. ; 2006: 337–346.

42. Pandey AK, Agrawal C, Sharma A, Sasikala P. Study of empirical approaches to analyze the software metrics. *ACM SIGSOFT Software Engineering Notes* 2013; 38(4): 1–5.

43. Forman EH, Gass SI. The analytic hierarchy process—an exposition. *Operations research* 2001; 49(4): 469–486.

44. Finnie GR, Wittig GE, Petkov DI. Prioritizing software development productivity factors using the analytic hierarchy process. *Journal of Systems and Software* 1993; 22(2): 129–139.

45. Sharma A, Kumar R, Grover P. Estimation of quality for software components: an empirical approach. *ACM SIGSOFT Software Engineering Notes* 2008; 33(6): 1–10.

46. Al-Qutaish RE, Muhairat MI, Al-Kasasbeh BM, Al-Kasasbeh B. The analytical hierarchy process as a tool to select open source software. In: UK. ; 2009: 39–44.

47. Zaidan AA, Zaidan BB, Al-Haiqi A, Kiah MLM, Hussain M, Abdulnabi M. Evaluation and selection of open-source EMR software packages based on integrated AHP and TOPSIS. *Journal of biomedical informatics* 2015; 53: 390–404.

48. 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: UYMS. ; 2018: 1-12.

49. Kogure M. Quality function deployment and CWQC in Japan. *Quality Progress* 1983: 26–32.

50. Caldiera VRBG, Rombach HD. The goal question metric approach. *Encyclopedia of software engineering* 1994: 528–532.

51. Boehm BW, Brown JR, Lipow M. Quantitative evaluation of software quality. In: ACM. ; 1976: 592–605.

52. Dağdeviren M, Tamer E. Tedarikçi firma seçiminde analitik hiyerarşi prosesi ve 0-1 hedef programlama yöntemlerinin kullanılması. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi* 2001; 16(1): 41–52.

53. Felek S, Yuluğkural Y, Aladağ Z. Mobil iletişim sektöründe pazar paylaşımının tahmininde AHP ve ANP yöntemlerinin kıyaslanması. *Endüstri Mühendisliği* 2007; 18(1): 6–22.

54. Saaty TL. *Decision making for leaders: the analytic hierarchy process for decisions in a complex world*. RWS publications . 1990.

ACKNOWLEDGMENTS

This research work is supported, in part, by Science Foundation Ireland grant $13/RC/2094\_2$ to Lero - the Science Foundation Ireland Research Centre for Software (www.lero.ie). During the research, we have received help and generous support from many people, to all of whom we would like to express our gratitude.

□