Measures related to social and human factors that influence productivity in software development teams

Liliana Machuca-Villegas
Facultad de Ingeniería, Universidad del Valle
Campus Meléndez, 760034 Cali, Colombia
liliana.machuca@correounivalle.edu.co

Gloria Piedad Gasca-Hurtado
Universidad del Medellín
Carrera 87 No. 30-65, 50026, Medellín, Colombia
gpgasca@udem.edu.co

Mirna Muñoz
Centro de Investigación en Matemáticas, Parque Quauntum, Ciudad del Conocimiento
Avenida Lassec, Andador Galileo Galilei, Manzana, 3 Lote 7 CP 98160 Zacatecas, México
mirna.munoz@cimat.mx

Abstract:
Software companies need to measure their productivity. Measures are useful indicators to evaluate processes, projects, products, and people who are part of software development teams. The results of these measurements are used to make decisions, manage projects, and improve software development and project management processes. This research is based on selecting a set of measures related to social and human factors (SHF) that influence productivity in software development teams and therefore in project management. This research was performed in three steps. In the first step, there was performed a tertiary literature review aimed to identify measures related to productivity. Then, the identified measures were submitted for its evaluation to project management experts and finally, the measures selected by the experts were mapped to the SHF. A set of 13 measures was identified and defined as a key input for designing improvement strategies. The measures have been compared to SHF to evaluate the development team's performance from a more human context and to establish indicators in productivity improvement strategies of software projects. Although the number of productivity measures related to SHF is limited, it was possible to identify the measures used in both traditional and agile contexts.

1. Introduction

Software development productivity is an interesting research topic within Software Engineering [1] and software project management fields [2]. Productivity is defined as the ratio between the output and input within the software development production process [3], [4]. The output is understood as the amount produced (software artifacts or services, tasks performed, quality, quantity in terms of functions, lines of code, implemented changes, among others), and the input is the effort dedicated to achieve an output (time, effort, labor cost, resources) [3][4][5][6][7]. Therefore, productivity is a key factor in project management and success [8].

Software productivity can be observed from different perspectives, namely, at a development level, a user level, and a management level as follows described: (1) At the development level, it may be related to the number of lines of code (LOC) produced [4] or to the inclusion of aspects related to the requirements, implementation, and validation [3]; (2) At the user level, it is possible to observe the degree of functionality achieved for the system, which is represented by the value delivered to the user [3], [4]; and (3) At the management level, it is focused on monetary aspects [3] and the teamwork performance [9].

All software companies need to measure their project productivity as this enables them to obtain indicators to manage and evaluate processes, projects, products, and people. The results of these measurements are used to make decisions and improve their software projects. Measuring productivity is used as a comparison tool for projects and developers [7]. Therefore, it provides work performance data for supporting the monitoring and controlling the process of software project management [2].

It is also suitable to improve decisions in software project management, define improvement strategies, and to reach high maturity levels in the organization leading to a more competitive company [6]. Besides, it is important to consider the productivity in the management of the software development project team, since the software product "is a direct product of the cognitive processes of individuals engaged in intellect-intensive, innovative teamwork" [2]. Thus, the team requires, in addition to technical skills, soft skills that promote team integration and cohesion. In this way, soft skills play an essential role in this process and can influence the productivity of the development team and project success [8].

In this context, Social and Human Factors (SHF) are of particular importance because they impact the results of software projects and are considered important elements affecting its costs [10], [11]. Failures in software projects may be related more to teamwork factors than to technical factors [12], [13]. Therefore, the study of SHF and their effect on software development productivity is in fact a matter of special interest for software companies [14] in view that personal aspects and human activities represent an opportunity to improve productivity [1].

This research aims at identifying, selecting, and defining productivity measures of software development associated with SHF. Our particular purpose is to establish a set of measures related to social and human factors that influence productivity in software development teams. Such measures are intended to help in the definition of improvement strategies with the inclusion of gamification initiatives [15]. In this case, the measures enable the evaluation of productivity strategies. These improvement strategies aim to stimulate SHF and analyze team productivity through the set of measures. To achieve it, it was necessary to distinguish from the set of measures found in the literature with those focused on the software development team and their management. Therefore, proposing strategies for software project management and productivity improvement may lead to more competitive software organizations [7]. In addition, addressing the difficulties related with SHF may help reduce software project management failures, improve team productivity, and even reduce both product cost and development time. For such purpose, a tertiary literature review was performed. Besides, the obtained measures were evaluated by a set of project management and productivity experts, the results of this evaluation enabling the selection of 13 productivity measures. Finally, the selected measures were related to SHF to get a set of productivity measures that influence software development productivity and therefore have a high impact on project management.

After the introduction, the article is organized as follows: Section 2 refers to studies related to the topics of interest in the software development area. Section 3 describes the methodological process followed to perform this research. Section 4 presents the results obtained from the tertiary literature review. Section 5 describes the selection process to evaluate the productivity measures. Section 6 states the definition of the selected measures and its comparison with SHF. Section 7 states the research threats, and finally, section 8 describes the conclusions and future work.

2. Related work

The study on software engineering productivity entails identifying factors that influence productivity and this involves defining related measures [7]. Influence factors are identified in the studies analyzed in this section, as well as productivity methods or models that are proposed from various work approaches. These studies focused on learning, modeling, and improving software development productivity, thus seeking to strengthen the development process benefiting the working team as well as the company.

Regarding the identification of factors that influence productivity, they have been classified based on different approaches, for instance, technical factors, organizational factors, product factors, and personal factors, among others. Wagner and Ruhe [16] presented in their literature review a set of productivity factors classified under technical factors and non-technical factors (*soft factors*). Oliveira et al. [17] performed a literature review and produced a list of factors classified as organizational factors, technical factors, and human factors.

Murphy-hill et al. [18] analyzed the productivity of developers in three organizations and identified that the prevailing factors were oriented toward non-technical factors, such as enthusiasm at work, colleagues supporting new ideas, and accepting valuable feedback regarding performance. Dias Canedo and Almeida Santos [19] researched those factors that affected software development productivity and open-source projects. Machuca-Villegas and Gasca-Hurtado [20] presented a classification of factors that influence software development productivity from a social and human perspective.

Regarding the measurement proposal and productivity models, Yilmaz [4] proposed a model based on social productivity and social capital. Hérnandez López [6] presented a productivity measure analysis in software development projects and proposes measuring at a job position level under a Data Envelopment Analysis (DEA) approach. De Oliveira Melo [21] describes a conceptual framework to study productivity in agile development teams.

Fatema and Sakib [22] created an agile teamwork productivity model with a Qualitative System Dynamics approach. The cause-effect relationship between productivity factors can help quantify and clarify the factor's influence to establish quantitative models.

Delaney and Schmidt [23] presented a literature review regarding the different approaches to measure and enhance software development productivity. They mention approaches oriented to quantify the number of outputs such as the function points or lines of code. They also mention approaches that compare the current effort with the estimated effort to produce outputs. This study reveals that the approaches described are oriented to specific scenarios rather than to a more general context. Likewise, Oliveira et al. [24] performed a systematic literature mapping to identify how productivity is being measured in the software development field. They discovered that the measurements are primarily used at the developer and software projects level. The prevailing measures are lines of code, time, and effort.

These studies highlight the importance of taking into account the social and human aspects of the work team. Consequently, recommendations and guidelines are suggested for project managers. However, they leave the door open for new researches on strategies that can be applied and validated easily in the context of software development teams and under the influence of SHF. Furthermore, these studies focus their attention on productivity measures, as they are important indicators for decision-making in software development team management and in the validation of new researches applied in this context. However, the number of measures related to the SHF may be limited because, in general, the measuring level is directed to the organization, the project, and the process. Thus, the interest in SHF within Software Engineering projects is still low [7].

In this context, this research is focused on finding productivity measures related to SHF that influence software development productivity [20], [25]. Of course, it should be taken into account that human factors or soft skills have a strong impact on the development team's productivity [21], IT project teams [8], and IT project complexity [26]. With this, we seek to contribute to the software engineering and software project management fields by proposing initiatives closely related to the working team [15]. We consider that the development process and its management is focused on people who play an important role in the results of the process performance.

3. Materials and methods

The process followed for the identification, selection, and definition of the measures related to SHF that influence software development productivity involves following three phases: (1) tertiary literature revision, (2) measure selection, and (3) measure comparison with SHF.

- Tertiary literature revision. Through the tertiary literature review, the productivity measures that are part of the objective of this research are obtained. The literature research was directed through the questions stated in Section 4.1. These questions allowed segmenting the review and contextualizing the results. In this phase, the list of the productivity measures associated with the context of this research is obtained (Table 2).

- Measure selection. Following this review, the process of selecting the productivity measures occurred. In this phase, it was necessary to rely on experts in organizational productivity and to define the selection criteria to review and evaluate the measures. The result of this phase is an evaluation report of the productivity measures (Table 3) and set of the productivity measures selected with their definition (Table 4).

- Measure comparison with SHF. The results of the measure selection indicate a low number of measures related to SHF. Therefore, it was necessary to make a measurement comparison with SHF. This final phase allowed proposing a possible relationship between productivity measures and the SHF. Its output is a list of productivity measures compared to SHF (Table 5).

Details and results of each of these phases are described in the following sections.

In order to conduct the tertiary literature review, the process adapted was the one proposed by Kitchenham [27], called Systematic Literature Review (SLR). A tertiary review is a systematic review performed based on secondary studies. The purpose of this review is to identify a set of productivity measures applied to the software development zprocess; those measures are specifically defined for the working team or the developers.

This review was conducted over a period extended from November 2019 to April 2020. The observation period of the selected studies was 2010 and 2019. The process and outcome of the review are described next.

3.1 Research questions

The following research questions are framed within the main research question: How is productivity measured about the social and human factors?

RQ1. Which productivity measures are used in software development? This research question aims to gather a set of productivity measures related to the research context.
RQ2. Which software engineering measurement level is associated with the productivity measure? This research question aims to identify measurement levels where the measures have been applied: this means in working teams or individuals.
RQ3. Do productivity measures include social or human factors? This research question aims to find measures related to the SHF that influence productivity. SHF were previously identified.
RQ4. How is the productivity measure defined? This research question aims to get acquainted with the variables that define the measure and facilitate their implementation in a certain context.

3.2   Search performed of secondary studies

The structure of the search string was proposed considering the topics that support the review. Such topics were software development, software development productivity, productivity measures, and literature reviews. Based on these topics, the following search string was set and executed:

("software engineering" OR "software development" OR "software maintenance" OR "software process") AND (productivity OR performance) AND (measure OR measurement OR measuring OR metric) AND (review OR overview OR literature OR meta-analysis OR "past studies" OR "in-depth survey" OR "subject matter expert" OR "analysis of research" OR "empirical body of knowledge" OR "overview of existing research" OR "body of published research" OR "mapping study" OR "systematic map")

The selected databases were ACM, IEEE, and Scopus. These scientific databases were selected according to the following criteria: (1) their relation with the computer science area; (2) their acknowledgment in the engineering field; and (3) the access available for the sponsors of the research.

The search results show a total number of 5003 studies (Table 1). The selection process described next is performed based on the results obtained.

3.3   Selection of studies

In order to obtain the primary studies, the following filters were defined: a) delete duplicates in each database; b) select studies using Microsoft Excel advanced filter option with search string keywords; c) delete duplicates after merging the three databases; d) title filter; e) abstract filter; f) content filter.

Moreover, together with the implementation of the filter, selection criteria were defined to implement it in the title, abstract, and content. Then, we present the defined criteria:

IC1. The study describes a literature review regarding productivity measures in software development.

IC2. The study follows a literature review process systematically or formally.

IC3. The study has been published in journals and conferences reviewed by peers.

The search and selection process is summarized in Table 1.

Table 1. The selection process for secondary studies

Database	Search Results	A	B	C	D	E	F
ACM	4103	2698	537		12	8	4
IEEE	491	491	96	655	6	3	1
Scopus	409	398	121		6	4	4
TOTAL	**5003**	**3587**	**754**		**24**	**15**	**9**

A) Delete duplicates in each database; B) Select studies using Microsoft Excel advanced filter option with search string keywords; C) Delete duplicates after merging the three databases; D) Title filter; E) Abstract filter; F) Content filter.

We reviewed 15 studies and selected nine studies closely related to the purpose of this research and the established selection criteria. Then, the tertiary literature review was conducted based on the following nine secondary reviews:

ID-1. Measuring Productivity in Agile Software Development Process: A Scoping Study (2015) [28]

ID-2. An Evolution of Software Metrics: A Review (2017) [29]

ID-3. Software Product Size Measurement Methods: A Systematic Mapping Study (2014) [30]

ID-4. A Systematic Mapping Study on Dynamic Metrics and Software Quality (2012) [31]

ID-5. Software Metrics Classification for Agile Scrum Process: A Literature Review (2018) [32]

ID-6. Methods for estimating agile software projects: A Systematic literature review (2018) [33]

ID-7. Software fault prediction metrics: A systematic literature review (2013) [34]

ID-8. Software engineering job productivity: A systematic review (2013) [35]

ID-9. Measuring and predicting software productivity: A systematic map and review (2011) [3]

3.4 Data Extraction

A Microsoft Excel template was used for data extraction, in which the information from the selected studies was consolidated. The information was divided into two sections: (1) *general data of the article*: Title, Author, Digital Object Identifier (DOI), Year, Keywords, and Abstract; and (2) *data related to research questions*: the objective of the systematic literature review, research question, productivity measures, definition of measures (metrics), measuring or abstraction level, mathematical approach, purpose, relationship with SHF.

During the data extraction, a mapping between the information of the selected studies and the required data was necessary to prepare this research. The steps taken to map each study are described below:

1. Article's general data registry; i.e., title, author, year, among others.
2. Article's comprehensive reading.
3. Extract data related to the questions created for this research. In those studies, where the data are implicit, the related information is inferred or the field is left empty. The extracted data are related to the measure, the abstraction, or measurement level, the purpose, and relationship with SHF (Table 2).
4. Data registry related to the research questions.

Taking into account the data of the consolidated studies, the following analysis was performed:

4. Results

The analysis is described based on the research questions posed. It is necessary to mention that some studies do not have information which is closely related to the research questions.

RQ1. Which productivity measures are used in software development?

The purpose of this question is to collect a set of productivity measures related to the context of this research. Table 2 shows a summary of the measures found in the review process. The selected studies provided literature review results regarding measurements, approaches, and methods or metrics of the software development process. Three of them bring forward a set of measures oriented to agile development [ID-1; ID-5; ID-6]. Some of these measures were used in traditional contexts as mentioned in Shah et al. [ID-1]. However, other studies consider these measures unsuitable in an agile context but keep adapting them to the process.

Some studies present measurement approaches that are aimed at organizing the measurement area of productivity concerning the different methods used. These approaches allow the grouping of measurement methods [ID-9]. The importance of this structure is highlighted because of its suitability for categorizing the selected measures.

Other studies describe the evolution of software measures, shifting from traditional methods to the aspect-oriented paradigm [ID-2]. This presentation of measures is related to the programming paradigm used. There also exist literature reviews oriented toward specific measures, such as the software product size [ID-3]; dynamic metrics [ID-4]; prediction metrics on software failures [ID-7]; and measures at a job position level [ID-8].

With the set of identified measures, it is possible to affirm that traditional measures are still being used, such as LOC and the function points (FP). Likewise, an evolution of the measures applied can be observed in the agile context. From the reports submitted in 2015 by Shah et al. [ID-1] to the ones described in 2018 by Kurnia et al. [ID-5] and Canedo et al. [ID-6], a new set of specific contextual measures are taken, even though traditional measures are still being reported [ID-6]. Additionally, it is stated that in the last three years, the mentioned studies foresee their results toward agile development [ID-5] and [ID-6]; this points out the trend of this new work approach in software development processes.

Those productivity measures, which are oriented towards research contexts other than to the purpose of this study were excluded, such as software dynamic metrics which try to measure features when executing the software, for instance, dynamic dependencies between components [ID-4] or either prediction metrics on software failures [ID-7].

Table 2. Summary of productivity measures

Studies	Measure / Metric / Approach / Method	Abstraction or Measurement level	Purpose	Relationship with SHF
ID-1 [28]	Lines of executable code/staff day [28]	Team	Evaluation	---
	Lines of code/person-hour [28]	Team	Evaluation	---
	Lines of code/hours [28]	Team	Evaluation	---
	Lines of code [28]	Team	Evaluation	---
	Average number of unadjusted function points completed per unit of time [28]	Team	Evaluation	---
	Resolved issues/month [28]	Per developer	Evaluation	---
	Functional size/effort [28]	Team (scrum)	Evaluation	---
	Function points/months [28]	Per developer	Evaluation	---
	Function Points/staff month [28]	Team	Evaluation	---
ID-2 [29]	**TRADITIONAL FUNCTION - ORIENTED METRICS**			
	Size Metrics: Lines Of Code (LOC): LOC/man-month [29].	Product	Estimation Evaluation	---
	Token Count: "These symbols are called tokens. The basic measures are: n1 = count of unique operators n2 = count of unique operands N1 = count of total occurrences of operators N2 = count of total occurrence of operands In terms of the total tokens used, the size of the program can be expressed as N = N1 + N2" [29]			
	Software Science Metrics: Halstead's model [29]	Product	Estimation Evaluation	---
	McCabe's Cyclomatic Metric [29]	Product	Estimation Evaluation	---
	OBJECT - ORIENTED METRICS			
	Chidamber and Kemerer's Metrics Suite [29]: - Weighted Methods per Class (WMC) [29] - Response for a Class (RFC) [29] - Lack of Cohesion of Methods (LCOM) [29] - Coupling between Object Classes (CBO) [29] - Depth of Inheritance Tree (DIT) [29] - Number of Children (NOC) [29]	Product	Evaluation	---
	MOOD'S Metrics for Object-Oriented Design [29]**:** - Method Hiding Factor (MHF) [29] - Attribute Hiding Factor (AHF) [29] - Method Inheritance Factor (MIF) [29] - Attribute Inheritance Factor (AIF) [29] - Polymorphism Factor (PF) [29] - Coupling Factor (CF) [29]	Product	Evaluation	---
	COMPONENT-ORIENTED METRICS			---
	- Average Interaction Density (AID) [29] -Incoming Interaction Density (IID) [29] - Outgoing Interaction Density (OID) [29]	Product	Evaluation	---
	ASPECT-BASED METRICS			---
	- Number of Aspects [29] - Number of Pointcuts per Aspect [29]	Product	Evaluation	---

Studies	Measure / Metric / Approach / Method	Abstraction or Measurement level	Purpose	Relationship with SHF
	- Number of Advices per Aspect [29] - Degree of Crosscutting per Pointcut [29] - Response for Advice [29]			
ID-3 [29]	**See Appendix A.**	Product	Estimation	---
ID-4 [30]	**Dynamic Metrics:** - Coupling [30] - Cohesion [30] - Complexity [30] - Method invocation [30] - Polymorphism [30] - Memory-related [30] - Code coverage [30] - Size/Structure [30]	Product	Evaluation	---
ID-5 [32]	**SPRINT PLANNING METRICS [32]:** - Effort estimate [32] - Story point [32] - Task effort [32] - Task's expected and end date [32] - Velocity [32]	Process Project Team	Estimation Evaluation	---
	DAILY SPRINT METRICS [32]: - # of an open defect [32] - Contribution [32] - The ratio of work spent and work remaining [32] - Standard violation [32] - The release burndown chart [32] - The sprint burndown chart [32]	Process Project Team-Individuals		Contribution
	SPRINT REVIEW METRICS [32]: - # of defects found in system test [32] - Bug correction time from new to the close state [32] - Business value delivered [32] - Customer satisfaction [32] - Completed web pages [32] - Defects deferred [32] - Defects per iteration [32] - Delivery on time [32] - Error density [32] - Focus factor [32] - Fulfillment of scope [32] - Number of stories [32] - Open defect severity index [32] - Percentage of Adopted work [32] - Percentage of Found work [32] - Progress chart (Scrum board) [32] - Unit test coverage for developed code [32] - Work capacity [32]	Process Project Team		Customer satisfaction
	SPRINT RETROSPECTIVE METRICS [32]: - Earn value management (EVM) [32] - Impression [32] - Influence [32] - Job satisfaction [32] - Net promoter score [32]	Process Project Team		-Impression -Influence -Job satisfaction -Net promoter score
ID-6	- Story Point [33]	Product (size) Project	Estimation	-Expert Opinion

Studies	Measure / Metric / Approach / Method	Abstraction or Measurement level	Purpose	Relationship with SHF
[33]	- Point of Function [33] - Expert Opinion [33] - Estimate based on model (COCOMO) [33] - Planning Poker [33] - Use-Case Points [33] - Custom Templates [33] - Number of Lines of Code [33] - Fuzzy based Framework for Estimation [33]			
ID-7 [34]	**See Appendix B**	Product	Prediction	---
ID-8 [35]	**Tasks/Time:** - Milestones/m [35] - (Completed program)/h [35] - (Completed tasks)/h [35] **LOC/Time:** - SLOC/h [35] - NCSS/h (No commentary Source Statement) [35]	Job position	Estimation	---
ID-9 [3]	**Measurement-based analytical models [3]:** - Weighted productivity factors [3] - Simple Input/Output Ratios [3] - Data envelopment analysis [3] - Bayesian belief networks [3] - Earned value analysis [3] - Statistical process control [3] - Balanced scorecard [3] - Metric space [3]	Project Individual Tasks Organization	Predictive Measurement Reactive Measurement	---
	Dynamic software development models [3]: - Continuous simulation [3] - Event-based simulation [3] - Hybrid simulation [3]	Project Individual Process	Predictive Measurement	---

RQ2. Which measurement level is associated with the productivity measure?

The point of this question is to identify measurement levels, i.e., in the working team or individuals where the measures have been applied. The measurement level refers, for example, to a project, module, process, developer, or tasks, among others [7]; it may also represent the analysis unit. The purpose is to analyze the measurement level to identify measures that are compatible with the team or the individual's productivity.

In connection with the above, measures related to the project, product, process, individual, tasks, organization, team, size, developer, and job position were found (See Table 2). The measures associated with the product, project, and software process prevail compared with those measures focused on the developer or on the working team, which appear on a smaller scale.

RQ3. Do productivity measures include social or human factors?

The purpose of this question is to find productivity measures that enable a relation with SHF that influence productivity in software development teams [20], [25]. The results show that a low number of measures fulfill these features. Instead, there is a trend toward a product and software process-oriented measurement. In Kurnia et al. [ID-5] and Canedo et al. [ID-6], a possible relationship with SHF was observed. The measures suggested in these studies are

intended for agile software processes. This leads to the conclusion that agile methodologies promote the development of SHF in software processes. The following measures were identified:

- Contribution: Evaluate the direct participation and the level of compromise during the daily Scrum meeting [32].
- Customer satisfaction: A quantitative evaluation of customers' satisfaction based on certain parameters [32].
- Impression: Review each team members' work based on the other team member's opinions [32].
- Influence: Measures individuals' engagement and participation in the project's progress [32].
- Job satisfaction: Developer's personal satisfaction with his work [32].
- Net promoter score: Measures customers' satisfaction and its impact. Customers' satisfaction enables them to recommend products to other potential customers [32].
- Fulfillment of scope: Shows how the team fulfills with the agreed terms in sprint planning [32].
- Expert opinion: Effort estimation technique in an agile context [33].

Six out of the eight measures defined above are related to SHF, which were already classified in previous studies Machuca-Villegas and Gasca-Hurtado [20], Machuca-Villegas et al. [25]. However, the other two measures (Customer Satisfaction and Net promoter score) are not related to such factors. The relationship between the six measures and the SHF is presented below:

1. Relationship between Contribution measure and SHF Commitment, Collaboration, Communication, Team cohesion.
2. Relationship between Impression measure and SHF Team cohesion, Autonomy.
3. Relationship between Influence measure and SHF Commitment, Collaboration, Communication, Team cohesion.
4. Relationship between Job satisfaction measure and SHF Motivation, work satisfaction.
5. Relationship between Fulfillment of scope measure and SHF Commitment.
6. Relationship between Expert Opinion measure and SHF Capabilities and Experiences in software development process, Capabilities, and Experiences in software project management.

RQ4. How is the productivity measure defined?

The purpose of this question is to find the defining variables that make its implementation easier in a certain context. The studies that present productivity measures in line with this research are [ID-5], [ID-6], and [ID-8]. Other studies were excluded since they were out of the scope of this research, either because the measures are included in the selected studies or because they represent generic approaches or highly complex quantitative approaches. A set of 48 potential measures was obtained from these three studies. It was necessary to conduct an evaluation and selection process for each of the measures. This process is described in the following section. The definition of the selected measures is stated in Section 6.

5. Selection of productivity measures

When searching the measures related to SHF that influence productivity in software development, a low number of measures that fulfill this requirement were found (RQ3). *Soft-factors* are difficult to measure [21]. Therefore, the scope of this search needs to expand toward measures that are in line with this research in such a way that: (1) they can be adapted to different contexts of software development; (2) they can be used to evaluate rather than to estimate; (3) they can be applied in development teams; and (4) they can be easily applied. These selection requirements are summarized in the criteria shown below:

- Generality: Measure suitable for various contexts, searching generality.
- Purpose: A measure in software may be used to estimate or evaluate. In this particular project, it is expected that the measure is used to evaluate rather than to estimate.
- Abstraction or measurement level: The measure can be applied in software development teams.
- A measure easy to define: A measure, which is easy to calculate. Its inputs and outputs can be easily obtained.

Besides understanding the definition of selection criteria, this selection process included an evaluation of each of the identified measures. The evaluation process included using a binary evaluation measurement for each criterion, where "1" is assigned when the measure meets the criterion and "0" when it does not. Four researchers were selected for this evaluation (two experts in organizational productivity, and two internal researchers from the project sponsoring this research). This group of researchers will be called the focus group. The focus group conducted an individual evaluation process of the measures according to the established criteria.

Table 3 shows the grade assigned by each focus group researcher and the evaluation regarding the fulfillment criteria. In this table, evaluation 1 shows the fulfillment of the four criteria assigned by each expert. While evaluation 2 shows those measures that at least meet three criteria assigned by each expert. Evaluation 2 was necessary given the low number of measures obtained in evaluation 1.

Table 3. Evaluation Report of the productivity measures

Measure	Researcher 1				Researcher 2				Productivity Expert 1				Productivity Expert 2				Evaluation - 1	Evaluation - 2
	SCPM-1	SCPM-2	SCPM-3	SCPM-4	SCPM-1	SCPM-2	SCPM-3	SCPM-4	SCPM-1	SCPM-2	SCPM-3	SCPM-4	SCPM-1	SCPM-2	SCPM-3	SCPM-4		
- Effort estimate [32]	0	0	1	1	1	0	0	1	1	0	1	0	0	1	1	0	NO	NO
- Story point [32]	1	0	1	1	1	0	0	1	0	0	1	0	0	0	1	0	NO	NO
- Task effort [32]	1	0	1	1	1	0	0	1	1	0	1	0	0	0	1	0	NO	NO
- Task's expected and end date [32]	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	NO	NO
- Velocity [32]	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	NO	YES
- # of an open defect	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	0	NO	NO
- Contribution [32]	0	1	1	1	1	1	1	0	0	1	1	1	0	1	1	0	NO	NO
- The ratio of work spent and work remaining [32]	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	YES
- Standard violation [32]	0	0	1	0	1	1	1	1	0	1	1	1	0	1	1	0	NO	NO
- The release burndown chart [32]	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	NO	YES
- The sprint burndown chart [32]	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	NO	YES
- # of defects found in system test [32]	0	1	1	0	1	1	0	1	0	1	1	1	1	1	1	1	NO	NO
- Bug correction time from new to the close state [32]	0	1	1	0	1	1	0	0	1	0	1	0	1	0	1	1	NO	NO
- Business value delivered [32]	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	NO	NO
- Customer satisfaction [32]	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	NO	NO
- Completed web pages [32]	0	0	1	0	0	1	1	1	0	1	1	1	0	0	1	1	NO	NO
- Defects deferred [32]	0	0	1	1	1	1	0	1	0	1	1	1	1	1	1	1	NO	NO
- Defect per iteration [32]	-	-	-	-	1	1	0	1	0	1	1	0	0	1	1	0	NO	NO
- Delivery on time [32]	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	NO	YES
- Error density [32]	0	1	1	1	0	1	0	0	0	1	1	1	0	1	1	1	NO	NO
- Focus factor [32]	1	0	1	0	1	1	1	1	1	1	1	0	1	1	1	0	NO	NO
- Fulfillment of scope [32]	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	NO	YES
- Number of stories [32]	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	NO	YES
- Open defect severity index [32]	0	0	1	1	0	1	0	0	0	1	1	0	0	1	1	0	NO	NO
- Percentage of Adopted work [32]	0	0	1	1	1	0	0	1	0	1	1	1	0	1	1	1	NO	NO
- Percentage of Found work [32]	1	0	1	1	1	1	0	1	0	1	1	0	0	1	1	0	NO	NO
- Progress chart (Scrum board) [32]	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	YES
- Unit test coverage for developed code [32]	0	0	1	1	1	1	0	1	0	1	1	1	0	1	1	1	NO	NO
- Work capacity [32]	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NO	NO
- Earn value management (EVM) [32]	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	1	NO	NO
- Impression [32]	1	0	1	0	1	1	1	1	1	0	0	0	1	0	1	1	NO	NO
- Influence [32]	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	NO	YES
- Job satisfaction [32]	1	0	1	0	1	1	1	1	1	1	1	1	1	0	1	1	NO	YES
- Net promoter score [32]	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	NO	YES
- Story Point [33]	1	0	1	1	1	0	0	1	1	0	1	0	1	0	1	0	NO	NO
- Point of Function [33]	0	0	1	0	1	0	0	0	0	1	0	1	0	1	1	0	NO	NO
- Expert Opinion [33]	1	0	1	0	1	0	0	1	1	0	1	1	1	0	1	1	NO	NO
- Estimate based on model (COCOMO) [33]	0	0	1	0	1	0	0	0	0	1	0	1	0	1	1	0	NO	NO

Measure	Researcher 1				Researcher 2				Productivity Expert 1				Productivity Expert 2				Evaluation - 1	Evaluation - 2
	SCPM -1	SCPM -2	SCPM -3	SCPM -4	SCPM -1	SCPM -2	SCPM -3	SCPM -4	SCPM -1	SCPM -2	SCPM -3	SCPM -4	SCPM -1	SCPM -2	SCPM -3	SCPM -4		
- Planning Poker [33]	1	0	1	1	0	0	0	1	0	1	0	1	0	1	0	1	NO	NO
- Use-Case Points (UCP) [33]	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0	NO	NO
- Custom Templates [33]	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	NO	NO
- Number of Lines of Code [33]	-	-	-	-	1	0	0	1	1	0	1	1	0	0	1	1	NO	NO
- Fuzzy based Framework for Estimation [33]	1	0	1	1	1	0	0	0	0	1	1	1	0	1	1	0	NO	NO
- Milestones/m [34]	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	NO	YES
- (Completed program)/h [34]	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	NO	YES
- (Completed tasks)/h [34]	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	NO	YES
- SLOC/h [34]	0	0	1	0	1	1	0	0	1	1	1	1	-	-	-	-	NO	NO
- NCSS/h (Non-commentary Source Statement) [34]	0	0	1	0	1	1	0	0	1	1	1	1	-	-	-	-	NO	NO

Two measures meet all the criteria: the ratio of work spent and work remaining, and the progress chart (Scrum board) (See column Evaluation 1 – Table 3). While 13 measures meet at least three criteria, including the two formerly mentioned (see column Evaluation 2 – Table 3). These are velocity, the ratio of work spent, and work remaining, the release burndown chart, the sprint burndown chart, delivery on time, fulfillment of scope, number of stories, progress chart (Scrum board), influence, net promoter score, milestones/m, (completed program)/h, (completed tasks)/h.

The set of measures with its corresponding definitions is presented below. The measures defined are those that fulfill the three selection criteria, so these answers RQ4. Table 4 provides the definition of the selected measures.

6. Productivity measures for software development influenced by SHF

The set of software development productivity measures related to SHF is represented by the measures selected from the tertiary literature review, and the evaluation conducted by the focus group. This set of measures contains two measures close to the SHF identified in Machuca-Villegas et al. [25] and 11 measures in line with this research, though these 11 measures have a weak relationship with SHF. This allows establishing the proposed set of measures.

The purpose of this research is limited to identify productivity influencing factors of software development to produce improvement strategies. Therefore, the aim is to evaluate the results of such strategies by measuring productivity with the measures detailed in the literature, especially, those measures associated with the development team level rather than evaluating SHF themselves.

6.1 Definition of the measures

Table 4 provides the definition of the selected measures.

Table 4. Set of the productivity measures selected

Name	Description	Measure
Velocity [ID-5] [36]	The number of work completed during the sprint. A measure used to calculate team productivity during the sprint. Besides, "it is used as a reference to forecast the amount of work that can be completed in the next sprint and estimate the number of sprints required to complete the project" [32].	$\sum$ all work accepted
The ratio of work spent and work remaining	"The ratio between the number of completed tasks (per day), and the remaining task (per day)" [32].	Work spent on day i for each task j in the sprint backlog – $Wsij$ [32]

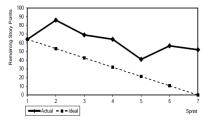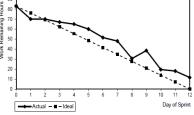| Name	Description	Measure
[ID-5] [37]	"The objective value is 1 or less, which means that the amount of work remaining decreases proportionally to the amount of work spent" [37].	Remaining work on day i for each task j in the sprint backlog – *Wrij* [32] $$\frac{\sum_{i=d1}^{d2-1}\sum_{j=1}^{n}WS_{i,j}}{\sum_{j=1}^{n}WR_{d1,j}-\sum_{j=1}^{n}WR_{d2,j}}$$
The release burndown chart [ID-5] [38]	Represents the amount of remaining work, and its decrease for each sprint keeping track of the sum of each story point to all incomplete stories in the Product Backlog [38]. Indicates team performance during the development process. This metric is aimed at monitoring the entire project, and the activities performed by the team [38].	A chart to describe the project's progress (ideal progress vs. real progress) based on the remaining tasks in a time framework (sprint).
The sprint burndown chart [ID-5] [38]	Represents the amount of remaining work that must be completed until the end of the Sprint [32] [38]. Indicates the team's performance during the development process. This metric is aimed at monitoring the entire project and the activities performed by the team [32] [38]. It is used to monitor the project's progress based on total work, and remaining sprints duration [32] [38].	A graphic that represents the work to be done in the sprint and if the teams are planned.
Delivery on time [ID-5] [39]	Indicates if the scope is being managed and understood [32] [39]. It is useful for tracking and predicting the project's progress [32][39]. The outcome is the customer's real value when delivering the features performed [32][39].	The features proportion is made on the planned delivery schedule.
Fulfillment of scope [ID-5] [37]	Shows how the team complies with the agreed commitments under the sprint planning [32] [37]. The objective value is 1. This means that the agreed commitments at the beginning of the Sprint or release were fulfilled [32][37].	Implemented #PBI / planned #PBI #Completed tasks / # Sprint Backlog tasks #Completed tasks during the sprint PBI (Product Backlog Item)
Number of stories [ID-5] [40]	Project process tracking based on the number of accepted stories [40] . "This metric is calculated as a simple count or weighing due to the story complexity, as simple, medium, and complex, regarding the number of stories in the sprint" [40].	# Developed and accepted stories

Name	Description	Measure
Progress chart (Scrum board) [ID-5] [40]	A tool to track the team member's progress. This shows the project's progress according to the status of the task (not started, in progress, and completed) in the chart [32]. It is useful for: - Comparing the current project status vs. the estimated status - Check if the team member's workload is balanced as to the number of hours and tasks. - Quickly check the project's progress [40].	
Influence [ID-5] [41]	Measures the individual's participation and commitment within the project's progress [32]. This activity can be in the form of information or as an intellectual and creative vision, production results, and project's general communication and team management [40].	Average (individual events per day) / Average (total events of the team per day) [26].
Net promoter score [ID-5] [42]	Measures customer's satisfaction and obtain feedback through a qualitative survey [32][42].	Qualitative survey
Milestones per minute [ID-8]	A measure used to evaluate the project's progress or team/developer performance [35].	Milestones/m Minutes (m)
Programs completed per hour [ID-8]	A measure used to evaluate the project's progress or team/developer performance [35].	(Complete programs)/h Hours (h)
Completed tasks per hour [ID-8]	A measure used to evaluate the project's progress or team/developer performance [35].	(Complete tasks)/h Hours (h) Difficulty and effort may also be taken into account.

6.2 Comparison of the measures selected with SHF

From the selected productivity measures, 11 do not reveal a relationship with SHF that influence productivity in software development. However, a comparison can be made between these measures and the definitions of SHF expressed in Machuca-Villegas et al. [25] to find a relationship between them. To such end, SHF were analyzed as factors influencing professional's productivity in a transversal manner to the software development process. Thus, in this relationship, the SHF is perceived as an implicit and underlying aspect of the individual, impacting his/her behavior, and, consequently, his/her productivity. Therefore, it is possible to identify an implicit relationship between the SHF with the selected productivity measures. This identified relationship is presented in Table 5.

The comparison process was conducted under the following steps:

1. Reviewing each of the productivity measures definitions described in Table 4.
2. Reviewing each of the SHF definitions described in Machuca-Villegas et al. [25].
3. Mapping and finding the relationship between the productivity measure and SHF according to each factor definition.
4. Justification of the established relationship. A productivity measure may be related to more than one SHF.
5. Having the comparison reviewed by a more experienced researcher.

The results of this comparison indicate that one of the SHF more closely related to these measures is "commitment". This reveals the importance of committing to doing the necessary tasks to obtain a successful project in line with the objectives set. On the other hand, in those measures associated with the team's performance, it was possible to identify a

close relationship with the SHF of "collaboration", "team cohesion", capabilities, and experiences in the software development process. These factors represent the need to conduct integrated teamwork.

Table 5. Comparison of measures and SHF

SHF	Measure	Comparison Justification
Commitment	• Velocity [ID-5] [36] • The ratio of work spent and work remaining [ID-5] [37] • Delivery on time [ID-5] [39] • Fulfillment of scope [ID-5] [37] • Number of stories [ID-5] [40] • Progress chart (Scrum board) [ID-5] [40] • Influence [ID-5] [41] • Milestones per minute [ID-8] • Programs completed per Hour [ID-8] • Task completed per hour [ID-8]	The team should carry out the necessary tasks to obtain a successful project in line with the objectives set. The progress and goal's achievement shall be communicated timely to team members.
Motivation	• Velocity [ID-5] [36] • The ratio of work spent and work remaining [ID-5] [37] • Influence [ID-5] [41]	Team members shall feel that the tasks they perform are valuable to achieve the objectives. Intrinsic motivation means doing something because it is enjoyable and important.
Collaboration	▪ The release burndown chart [ID-5] [38] ▪ The sprint burndown chart [ID-5] [38] ▪ Influence [ID-5] [41] ▪ Milestones per minute [ID-8] ▪ Programs completed per Hour [ID-8] ▪ Task completed per hour [ID-8]	Team members shall work collaboratively to reach project goals. Team members shall be willing to assist, support, and encourage their colleagues.
Team Cohesion	▪ The release burndown chart [ID-5] [38] ▪ Progress chart (Scrum board) [ID-5] [40] ▪ Influence [ID-5] [41] ▪ Milestones per minute [ID-8] ▪ Programs completed per Hour [ID-8] ▪ Task completed per hour [ID-8]	Team members need to work at similar rates. Activities should be executed on time and all responsible parties shall participate. It is important that members feel identified with the team in which they participate in a voluntary and motivated manner. Each team member shall enjoy performing tasks with their colleagues.
Capabilities and experiences in software development process.	▪ The release burndown chart [ID-5] [38] ▪ Milestones per minute [ID-8] ▪ Programs completed per Hour [ID-8] ▪ Task completed per hour [ID-8]	The team shall know the subject or have experience working in similar contexts. The team shall have knowledge or experience with the tools and programming language necessary for the project. Team members must have the ability to implement efficient solutions to meet the project's requirements.
Team Cohesion	▪ The sprint burndown chart [ID-5] [38]	Team members need to work at similar rates. Activities must be executed on time and all responsible parties shall participate.

SHF	Measure	Comparison Justification
Capabilities and experiences in software development process	▪ The sprint burndown chart [ID-5] [38]	The team shall know the subject or have experience working in similar contexts. The team shall have knowledge or experience with the tools and programming language necessary for the project. Team members must have the ability to implement efficient solutions to meet the project's requirements.
Communication	▪ Delivery on time [ID-5] [39] ▪ Influence [ID-5] [41]	The project's goals and activities involved should be clearly and expressly communicated to all project participants. Communication among team members is very important support.
Job satisfaction	▪ Progress chart (Scrum board) [ID-5] [40]	Team members shall be satisfied with the equal distribution of the tasks.
Not applicable	▪ Net promoter score [ID-5] [42]	SHF influence the team's productivity; the relationship with the client has not been taken into account.

7. Threat to the validity of the results

It is important to take into account the threats to the validity of the results of all research studies. Particularly in this research, among the identified threats are those related to the generalization of the results, lack of details in the studies analyzed, the researcher's bias, and the limitation regarding the number of measures closely related to SHF.

The generalization of the results is restricted to some secondary studies selected in the tertiary literature review. The search strategy used could have omitted collecting some relevant articles that have an impact on the results. Therefore, three databases were used—all of scientific nature and specialized in computer science. Likewise, the selection of studies was led by the inclusion criteria involved in the research questions defined for this investigation.

The lack of details in the analyzed studies could have influenced the interpretation of the results, especially when selecting the productivity measures. In some cases, it was necessary to resort to a primary source to ease the definition of the selected measures.

The researcher's bias is another frequent threat found in the investigation. In this research, bias is exposed in the process of the tertiary literature review, in the measure selection process, and the process regarding the comparison of the selected measures with SHF. In order to mitigate this threat, the following was considered:

- ▪ The support of a more experienced researcher to accomplish the tertiary review protocol.
- ▪ The experience of experts on organizational productivity (focus group) to assess the selection criteria of the productivity measures.
- ▪ Having the comparison results reviewed by a more experienced researcher.
- ▪ Having a third researcher as a reviewer for the entire process and the results obtained.

To conclude, the limited number of measures associated with SHF risked the research outcomes. However, the selection of new criteria was proposed to obtain measures in line with the research context. Moreover, a comparison between the selected measures and SHF was proposed to suggest possible relationships between them.

8. Conclusion and future work

Software companies need to obtain indicators to manage and evaluate processes, projects, products, and people involved. Indicators are used to make decisions, manage projects, and improve processes. In most cases, the indicators are related to productivity measures. In software development companies, these indicators are used as a comparison tool between projects and developers to define improvement strategies and make decisions to manage software projects.

In this research, a set of productivity measures in software development is presented, and defined based on the results of a tertiary literature review and an evaluation process performed by project management and productivity experts. As result, a set of 13 measures were identified. These measures become a basis for assessing the software development teams' management and the results of productivity improvement strategies when SHF are introduced within these teams.

The results of this comparison indicate that one of the SHF more closely related to these measures is "commitment". This reveals the importance of committing to doing the necessary tasks to obtain a successful project in line with the objectives set. These results are in line with the results obtained from the study performed by Cunha De Oliveira [7] which revealed that software managers and project leaders consider that the commitment factor is essential to obtain a successful project. On the other hand, in those measures associated with the team's performance, it was possible to identify a close relationship with the SHF of "collaboration", "team cohesion", "capabilities, and experiences in the software development process". These factors represent the need to conduct integrated teamwork as the success of a project also depends on how professionals perform their tasks and the way they interact with their team. The above ratifies Capretz and Ahmed's thoughts [43] which promote the importance of soft skills in the field of Software Engineering in the performance of professionals.

The tertiary review helped to identify that traditional measures such as LOC and FP are still being used. Although Hernadez López [6] indicated that these measures help to assess the project delivery efficiency, they still do not reflect the work team's activities in such a way as to facilitate their relationship with the SHF. Similarly, new measure proposals applied in the agile context were identified. In this context, productivity measures can be more related to the SHF [28] and, therefore, they turn out to be key measures for the work team. Software dynamic metrics were also identified, which focus on executing software and the prediction measures related to software failures.

The abstraction level of the identified measures was associated with the project, product, process, individuals, tasks, organizations, team, size, developer, and job position. The levels associated with the product, project, and software process prevail while the levels focused on the developer, or the working team appears on a smaller scale. These results show a constant in the use of traditional productivity measures and, as expressed by Cunha De Oliveira [7] the SHF are gaining importance in the management of software projects.

The secondary studies related to agile methods present measures associated with SHF, which confirm that agile methodologies improve the SHF development in the software projects. This is related to the Agile Manifesto which highlights the importance of individuals and interactions over processes and tools (https://agilemanifesto.org/). In this research, 76.92% of the measures identified are included in the context of Agile development.

Moreover, it was possible to compare the obtained set of 13 measures and the SHF. Through such comparison, a preliminary approach between *soft factors* and productivity measures is proposed, of course having into account that these factors are not easy to measure [21] and that measuring them is out of this research scope.

Since SHF were included in the performance of work teams, this research support in understanding the area of knowledge associated with software development and project management processes. This allows establishing a set of useful management and decision-taking measures based on concrete measurement indicators. The understanding of this knowledge area based on a set of measures, such as those identified in this research, provides support for designing improvement strategies and sets productivity indicators which are important for software development projects.

The main research findings facilitate working on new proposals within this context. Some of these proposals can be geared toward the following future works:

- Inclusion of a set of 13 measures to a model based on gamification and SHF to influence in software development productivity. Based on this set of measures, it will be possible to have indicators to analyse how the model influences on the productivity of the development team.
- Design of improvement strategies. The main goal is to focus these strategies on the management of software development projects beginning with the measures set identified. Such a set of measures is an essential input to design strategies focused on the encouragement of SHF supported in gamification.
- Design of quasi experiment to analyze the impact of the measures in the improvement strategies applied.
- Creation of a Simulation system based on system dynamics to study the relationship between the identified measures and the SHF.
- Broaden the search for productivity measures to identify those that facilitate a general measurement of both teams and individuals may be necessary.

Acknowledgements

References

[1]   S. C. de Barros Sampaio, "Uma meta estratégia para melhoria de produtividade no desenvolvimento de software, baseada em melhores práticas identificadas na literatura," Master's Thesis, Centro de Informática, Universidade Federal de Pernambuco, Brasil, 2010.

[2]   Project Management Institute and IEEE Computer Society, *Software Extension to the PMBOK® Guide Fifth Edition*, Fifth. January. Project Management Institute, Inc., 2013.

[3]   K. Petersen, "Measuring and predicting software productivity: A systematic map and review," *Information and Software Technology*, vol. 53, no. 4, pp. 317–343, 2011.

[4]   M. Yilmaz, "A Software Process Engineering Approach to Understanding Software Productivity and Team Personality Characteristics: An Empirical Investigation," PhD. dissertation, Faculty of Engineering and Computing, School of Computing, Dublin City University, Dublin, Ireland, 2013.

[5]   E. A. de Barros, "Catálogo de fatores que influenciam a produtividade no desenvolvimento de software," Master's Thesis, CIN - Centro de Informática, Universidade Federal de Pernambuco, Recife, Brasil, 2010.

[6]   A. Hernández López, "Medidas de productividad en los proyectos de desarrollo de software: una aproximación por puestos de trabajo," Tesis Doctoral, Departamento de Informática, Universidad Carlos III, España, 2014.

[7]   E. C. Cunha De Oliveira, "Fatores de influência na produtividade dos desenvolvedores de organizaciones de software," Tese de Doutorado, Instituto de Computação, Universidade Federal Do Amazonas, Manaus, Brasil, 2017.

[8]   C. Iriarte and S. Bayona, "It projects success factors: A literature review," *International Journal of Information Systems and Project Management*, vol. 8, no. 2, pp. 49–78, 2020.

[9]   Project Management Institute, *A Guide to the Project Management Body of Knowledge*, Fifth. Pennsylvania: Project Manager Institute, Inc. 2013.

[10]   S. Adolph, W. Hall and P. Kruchten, "Using grounded theory to study the experience of software development," *Empirical Software Engineering*, vol. 16, no. 4, pp. 487–513, 2011.

[11]   L. Fernández-Sanz and S. Misra, "Influence of human factors in software quality and productivity," in *Computational Science and Its Applications - ICCSA 2011*, B. Murgante, O. Gervasi, A. Iglesias, D. Taniar, B.O. Apduhan, Eds, Berlin, Heidelberg: Springer, 2011, vol. 6786, pp. 257–269.

[12]   M. Muñoz, L. Hernández, J. Mejia, A. Peña, N. Rangel, C. Torres and G. Sauberer, "A Model to Integrate Highly Effective Teams for Software Development," in *European Conference on Software Process Improvement*, Ostrava, Czech Republic, 2017, pp. 613–626.

[13]   N. Rangel, C. Torres, A. Peña, M. Muñoz, J. Mejia and L. Hernández, "Team Members' Interactive Styles Involved in the Software Development Process," in *European Conference on Software Process Improvement*, Ostrava, Czech Republic, 2017, pp. 675–685.

[14]   S. C. de Barros Sampaio, E. A. Barros, G. S. De Aquino, M. J. Carlos E Silva and S. R. De Lemos Meira, "A review of productivity factors and strategies on software development," in *5th International Conference on Software Engineering Advances, ICSEA 2010*, Nice, France, 2010, pp. 196–204.

[15]   L. Machuca-Villegas and G. P. Gasca-Hurtado, "Aproximación de un modelo basado en gamificación para influir en la productividad de equipos de desarrollo de software," in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, Coimbra, Portugal, 2019.

[16]   S. Wagner and M. Ruhe, "A Systematic Review of Productivity Factors in Software Development," in *Proceedings of the 2nd International Software Productivity Analysis and Cost Estimation (SPACE 2008)*, 2008, pp. 1–6.

[17]   E. Oliveira, T. Conte, M. Cristo and N. Valentim, "Influence Factors in Software Productivity - A Tertiary Literature Review," *International Journal of Software Engineering and Knowledge Engineering*, vol. 28, no. 11–12, pp. 1795–1810, 2018.

[18]   E. Murphy-hill, C. Jaspan, C. Sadowski, D. Shepherd, M. Phillips, C. Winter, E. Smith and M. Jorde, "What Predicts Software Developers ' Productivity ?," *IEEE Transactions on Software Engineering*, vol. 47, no. 3, pp. 582 - 594, 2019.

[19]   E. D. Canedo and G. A. Santos, "Factors affecting software development productivity: An empirical study," in *SBES 2019: Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, Salvador, Brazil, pp. 307–316, 2019.

[20]   L. Machuca-Villegas and G. P. Gasca-Hurtado, "Towards a Social and Human Factor Classification Related to Productivity in Software Development Teams," in *Trends and Applications in Software Engineering. International Conference on Software Process Improvement (CIMPS 2019)*, Leon, México 2019, pp. 36–50.

[21]   C. de Oliveira Melo, "Productivity of agile teams: an empirical evaluation of factors and monitoring processes," PhD. dissertation, Institute of Mathematics and Statistics, University of São Paulo, São Paulo, Brazil, 2015.

[22]   I. Fatema and K. Sakib, "Using Qualitative System Dynamics in the Development of an Agile Teamwork Productivity Model," *International Journal on Advances in Software*, vol. 11, pp. 170–185, 2018.

[23]   S. Delaney and D. Schmidt, "A Productivity Framework for Software Development Literature Review," in *2nd International Conference on Software Engineering and Information Management*, Bali, Indonesia 2019, pp. 69–74.

[24]   E. Oliveira, D. Viana, M. Crist and T. Conte, "How have software engineering researchers been measuring software productivity?: A systematic mapping study," in *ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems*, Porto, Portugal, 2017, pp. 76–87.

[25]   L. Machuca-Villegas, G. P. Gasca-Hurtado, L. M. Restrepo Tamayo and S. Morillo Puente, "Social and Human Factor Classification of influence in Productivity in Software Development Teams," in *Systems, Software and Services Process Improvement. EuroSPI 2020. Communications in Computer and Information Science*, Düsseldorf, Germany, 2020, pp. 717–729.

[26]   S. Morcov, L. Pintelon and R. Kusters, "Definitions, characteristics and measures of IT project complexity - a systematic literature review", *International Journal of Information Systems and Project Management,* vol. 8, no. 2, pp. 5-21, 2020.

[27]   B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3," *Engineering*, vol. 45, no. 4ve, p. 1051, 2007.

[28]   S. M. A. Shah, E. Papatheocharous and J. Nyfjord, "Measuring productivity in agile software development process: A scoping study," in *ICSSP 2015: Proceedings of the 2015 International Conference on Software and System Process*, Tallinn, Estonia, 2015, pp. 102–106.

[29]   R. S. Chhillar and S. Gahlot, "An evolution of software metrics: A review," *ICAIP 2017: Proceedings of the*

International Conference on Advances in Image Processing, Bangkok, Thailand, 2017, pp. 139–143.

[30]   S. S. Bajwa, C. Gencel and P. Abrahamsson, "Software product size measurement methods: A systematic mapping study," in *2014 Joint Conference of the International Workshop on Software Measurement, IWSM 2014 and the International Conference on Software Process and Product Measurement, Mensura 2014*, Rotterdam, Netherlands, 2014, pp. 176–190.

[31]   A. Tahir and S. G. MacDonell, "A systematic mapping study on dynamic metrics and software quality," in *IEEE International Conference on Software Maintenance, ICSM*, Trento, Italy, 2012, pp. 326–335.

[32]   R. Kurnia, R. Ferdiana, and S. Wibirama, "Software metrics classification for agile scrum process: A literature review," in *2018 International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2018*, Yogyakarta, Indonesia, 2018, pp. 174–179.

[33]   E. D. Canedo, D. P. Aranha, M. De Oliveira Cardoso, R. P. Da Costa, and L. L. Leite, "Methods for estimating agile software projects: Systematic literature review," in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE,* 2018, pp. 34–39.

[34]   D. Radjenović, M. Heričko, R. Torkar and A. Živkovič, "Software fault prediction metrics: A systematic literature review," *Information and Software Technology*, vol. 55, no. 8, pp. 1397–1418, 2013.

[35]   A. Hernández-lópez, R. Colombo-Palacios and Á. García-Crespo, "Software Engineering Job Productivity - A systematic Review," *International Journal of Software Engineering and Knowledge Engineering,* vol. 23, no. 03, pp. 387-406, 2013.

[36]   S. Downey and J. Sutherland, "Scrum metrics for Hyperproductive Teams: How they fly like fighter aircraft," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, Wailea, HI, USA, 2013, pp. 4870–4878.

[37]   V. Mahnič and I. Vrana, "Using stakeholder-driven process performance measurement for monitoring the performance of a Scrum-based software development process," *Elektrotehniski Vestnik/Electrotechnical Review*, vol. 74, no. 5, pp. 241–247, 2007.

[38]   V. Mahnic and N. Zabkar, "Measuring progress of scrum-based software projects," *Elektronika ir Elektrotechnika*, vol. 18, no. 8, pp. 73–76, 2012.

[39]   K. V. J. Padmini, H. M. N. Dilum Bandara and I. Perera, "Use of software metrics in agile software development process," in *MERCon 2015 - Moratuwa Engineering Research Conference*, Moratuwa, Sri Lanka , 2015, pp. 312–317.

[40]   M. Agarwal and R. Majumdar, "Tracking Scrum projects Tools, Metrics and Myths About Agile," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 3, pp. 97–104, 2012.

[41]   R. F. Gamble and M. L. Hale, "Assessing individual performance in Agile undergraduate software engineering teams," in *Proceedings - Frontiers in Education Conference, FIE*, Oklahoma City, OK, USA, 2013, pp. 1678–1684.

[42]   P. Green, "Measuring the impact of scrum on product development at adobe systems," in *Proceedings of the 44th Hawaii International Conference on System Sciences*, Kauai, HI, USA, 2011, pp. 1–10.

[43]   L. F. Capretz and F. Ahmed, "A Call to Promote Soft Skills in Software Engineering," *Psychology and Cognitive Sciences*, vol. 4, no. 1, pp. e1–e3, 2018.

Appendix A. Summary of productivity measures [ID-3]

Studies	Measure / Metric / Approach / Method	Abstraction or Measurement level	Purpose	Relationship with SHF
ID-3 [30]	Albrecht/IFPUG FPA MK II FPA Project Size Unit Size as a Vector The mini model method Use Case Size Point Estimate (estimate size and effort) Full Function Points (FFP) COSMIC RmFFP COSMIC like model for Web Based Appl. Object Oriented Method CFP (OOmCFP) Model for size estimation from S-BPM Data Mart Size Measurement Improved FPA (Fuzzy Rules and BP Network) Fuzzified FPA Object Oriented Hypermedia Function Point Requirement Points Updating OOmFP OOmFP Object Oriented Method FP for WEB (OOmFPWeb) COSMIC with PRiM Non-Functional Req. Size Measurement Method Cloud Migration Point Method Multi Granularity OO Est. Model Fuzzy Size Estimation Procedure Class Point Extension Class Point Component Point UCP Fuzzy Logic Model to Approximate Size UML based COSMIC COSMIC to Problem Frames Approximate COSMIC FP using Text Mining E-COSMIC UML based COSMIC Functional Size of Interactive Sys. FAST FSM Refined FSM for Embedded System (Simulink) FSM for Embedded System (Req. expressed as Simulink) Guidelines to measure size in the context of Autostar Fine grain measurement for UML use case diagram Early and Quick Estimation Technique for COSMIC COSMIC for Real Time OO Modeling Lang. (ROOM) COSMIC from Business Process Model Method to estimate size for CAL system Software Size Model Rapid FPA A refined method to measure UML Model Object Oriented Function Point Simplified IFPUG Object Oriented Design Function Point NESMA	Product	Estimation	---

Studies	Measure / Metric / Approach / Method	Abstraction or Measurement level	Purpose	Relationship with SHF
	A size estimation method			
	Estimating size of Formal Comm. Protocol Specification			
	Database Size (No. of Entities, No. of Screens)			
	Method to measure database size			
	Database Size Estimation Based on ER (DSER)			
	Probabilistic Size			
	Neural Network for size estimation			
	No. of Test Cases for system size			
	SLOC estimation from Conceptual data model			
	Estimation of test suite size from test case number			
	SLOC Estimation from UML Class Diagram			
	Component Size Estimation			
	Refined Predicting Object Point (PoP)			
	Fuzzy Function Point Analysis (FFPA)			
	Counting rules for MK II FP in SSAD Environment			
	A generalization of FP			
	Size Estimation Method			
	Measurement of amount of Information			
	MTPF FP Measure Method			
	Bottom Up Software Size Estimation			
	Bottom up and top down approaches of FPA			

Appendix B. Summary of productivity measures [ID-7]

Studies	Measure / Metric / Approach / Method	Abstraction or Measurement level	Purpose	Relationship with SHF
ID-7 [34]	AHF - Attribute Hiding Factor	Product	Prediction	---
	AIF - Attribute Inheritance Factor			
	COF - Coupling Factor			
	MHF - Method Hiding Factor			
	MIF - Method Interface Factor			
	POF - Polymorphism Factor			
	SCC - Similarity-based Class Cohesion			
	ANA - Avgrage Number of Ancestors			
	CAM - Cohesion Among Methods			
	CIS - Class Interface Size			
	DAM - Data Accesss Metric			
	DCC - Direct Class Coupling			
	DSC - Design size in classes			
	MFA - Measure of Functional Abstraction			
	MOA - Measure of Aggregation			
	NOH - Number of hierarchies			
	NOM - Number of Methods			
	NOP - Number of polymorphic methods			
	LCC - Loose class cohesion			
	TCC - Tight class cohesion			
	ACAIC			
	ACMIC			
	AMMIC			
	Coh - A variation on LCOM5			
	DCAEC			

Measures related to social and human factors that influence productivity in software development teams

Studies	Measure / Metric / Approach / Method	Abstraction or Measurement level	Purpose	Relationship with SHF
	DCMEC			
	DMMEC			
	FCAEC			
	FCMEC			
	FMMEC			
	IFCAIC			
	IFCMIC			
	IFMMIC			
	OCAEC			
	OCAIC			
	OCMEC			
	OCMIC			
	OMMEC			
	OMMIC			
	ATTRIB - Attributes			
	DELS - Deletes			
	EVNT - Events			
	READS - Reads			
	RWD - Read/write/deletes			
	STATES - States			
	WRITES - Writes			
	CBO - Coupling between object classes			
	DIT - Depth of inheritance tree			
	LCOM - Lack of cohesion in methods			
	LCOM2 - Lack of cohesion in methods			
	NOC - Number of children			
	NTM - Number of trivial methods			
	RFC - Response for a class			
	WMC - Weighted methods per class			
	AMC - Average method complexity			
	Past faults - Number of past faults			
	Changes - Number of times a module has been changed			
	Age - Age of a module			
	Organization - Organization			
	Change set - Number of modules changed together with the module			
	N1 - Total number of operators			
	N2 - Total number of operands			
	$\eta 1$ - Number of unique operators			
	$\eta 2$ - Number of unique operands			
	AID - Average inheritance depth of a class			
	LCOM1 - Lack of cohesion in methods			
	LCOM5 - Lack of cohesion in methods			
	Co - Connectivity			
	LCOM3 - Lack of cohesion in methods			
	LCOM4 - Lack of cohesion in methods			
	ICH - Information-flow-based cohesion			
	ICP - Information-flow-based coupling			
	IH-ICP - Information-flow-based inheritance coupling			
	NIH-ICP - Information-flow-based non-inheritance coupling			
	CMC - Class method complexity			
	CTA - Coupling through abstract data type			
	CTM - Coupling through message passing			
	NAC - Number of ancestor			
	NDC - Number of descendent			
	NLM - Number of local methods			
	DAC - Data abstraction coupling			

Measures related to social and human factors that influence productivity in software development teams

Studies	Measure / Metric / Approach / Method	Abstraction or Measurement level	Purpose	Relationship with SHF
	DAC1 - Data abstraction coupling			
	MPC - Message passing coupling			
	NCM - Number of class methods			
	NIM - Number of instance methods			
	NMA - Number of methods added			
	NMI - Number of methods inherited			
	NMO - Number of methods overridden			
	NOA - Number of attributes			
	NOAM - Number of added methods			
	NOO - Number of operations			
	NOOM - Number of overridden methods			
	NOP - Number of parents			
	NPAVG - Average number of parameters per method			
	SIX - Specialization index			
	C3 - Conceptual cohesion of Classes			
	CC - McCabe's Cyclomatic Complexity			
	Delta - Code delta			
	Churn - Code churn			
	Change request - Change request			
	Developer - Number of developers			
	CLD - Class-to-leaf depth			
	NOA - Number of ancestors			
	NOD - Number of descendants			
	LOC - Lines of Code			

Biographical notes

Liliana Machuca-Villegas

Liliana Machuca-Villegas is a System Engineer from Universidad Francisco de Paula Santander, Colombia. She received the MSc in Engineering from Universidad del Valle, Colombia. Currently she is PhD. candidate from Universidad de Medellín, Colombia.

She is assistant professor at Escuela de Ingeniería de Sistemas y Computación at the Universidad del Valle, Colombia. Her main research areas include Project Management, Gamification, and Software Engineering.

Gloria Piedad Gasca-Hurtado

Gloria Piedad Gasca-Hurtado is associated professor-researcher in the Engineering Faculty of Universidad de Medellín. Her Ph.D. was taken in Universidad Politécnica de Madrid, Spain in Languages, Informatics Systems and Software Engineering Department in Informatics Faculty. Her research areas include Information technology and communications (TIC) software process improvement and optimization, multi-model environment for software development, software development and agile methodologies applied to small enterprises (SME's), security informatics, among others. She serves as Director of Software Engineering Academic Program.

Mirna Muñoz

Mirna Muñoz has a PhD in Computer Languages, Informatics Systems and Software Engineering from the Polytechnic University of Madrid. She has held a postdoctoral stay at the University of Carlos III of Madrid. She is working as a Software Engineering researcher at the Software Engineering Unit of CIMAT in México. She is a member of the AMEXCOMP, and the WG24 of ISO. Her current research interest is on process and its assets definition in software and different domains, project management, software process improvements focusing on the human factor, multi-model environments, the implementation of quality models and standards as well as methodologies, and the integration of high effective teams.