Software Project Managers' Perceptions of Productivity Factors: Findings from a Qualitative Study

Edson Oliveira, Tayana Conte, Marco Cristo,
ICOMP – Instituto de Computação
UFAM – Universidade Federal do Amazonas
Manaus, Brasil
{edson.cesar, tayana, marco.cristo}@icomp.ufam.edu.br

Emilia Mendes
School of Computing
Blekinge Institute of Technology
Karlskrona, Sweden
emilia.mendes@bth.se

ABSTRACT

Context – Developers' productivity plays an important role in software development organizations; however, in many cases the management of such human capital is mainly based on how project managers perceive productivity. Therefore, it is important to investigate what these perceptions are in practice. *Goal* – This study's main goal is to understand project managers' perception regarding developers' productivity. *Method* – We employed a qualitative research methodology using semi-structured interviews for data collection. We interviewed 12 managers from three software development organizations in the city of Manaus (Brazil). *Results* – We identified that the managers' perceptions about developers' productivity are influenced by four different factors: (1) tasks delivered on time, (2) produced artifacts that do not need rework, (3) products that meet stakeholders' expectations, and (4) personal behavior such as focus and proactivity. *Conclusions* – This qualitative study shows a perception of developers' productivity different from that presented in other research papers, and suggests that human factors play an important role in managers' perceptions about productivity. Future work will investigate how these perceptions concretely influence developers' productivity, and how they relate to the existing developers' productivity factors in the literature.

CCS Concepts

• **Software Engineering → Metrics → Performance Measures**; **Software Engineering → Management → Productivity**.

Keywords

Perception; Productivity; Qualitative Study.

1. INTRODUCTION

Productivity is a key factor for many software organizations [2, 18] that focus on the development of high quality software products with low production costs. For such companies, one way to reduce the production costs of software development is through the increase of their productivity [2]. Therefore, productivity improvement in software development should be a constant focus for software organizations [7].

In Software Engineering, productivity is frequently defined, from an economic viewpoint, as "*the effectiveness of productive effort, especially in industry, as measured in terms of the rate of output per unit of input*" [13]. Usually in such definition, the unit of input used is the development effort (measured in persons-time) and the output frequently used is software size [4, 12]. Thus, to increase productivity in a software project, it is necessary to either reduce the development effort or increase the amount of developed software. In any case, the intervention must be applied to the development team by means of, for example, new processes, methods or tools. Nevertheless, the focus of the intervention should be on software developers, because software development is essentially a human activity [1], and the measure of developer's productivity in Software Engineering is the number of product units produced over some unit of effort [12].

Software development is diversified and multidisciplinary and, therefore, development teams are faced with many different kinds of tasks (e.g. analysis, design, programming, testing and others) [5]. From these tasks, programming is a central activity to the software development process, and the software developer is the core cell of the development team. Despite the importance of his/her role to the productivity of software development [14], most studies on productivity mainly focus on the organization or the project, but a few on the developer [11].

Measuring software productivity is a key element for managers since they can use this measure to compare the efficiency of different developers in the same organization and compose teams according to desirable time and quality requirements. It is important to understand managers' perceptions of developer's productivity even when productivity metrics are not available or reliable, because the management of such human resources has a significant impact on organization's performance [8]. Therefore, the goal of this study is to investigate how software managers understand and judge intuitively software developers' productivity. To the best of our knowledge, this is the first study based on data from interviews about the software managers' perceptions. These results will contribute towards enhancing our understanding on how productivity is perceived in practice.

The remainder of this paper is organized as follows. Section 2 discusses the background and related work. Section 3 describes the research questions, and the research process applied to answer them. Section 4 presents the results of our study, while Section 5 discusses these results. Section 6 discusses the threats to validity, and finally, Section 7 concludes this paper.

2. RELATED WORK

Software developers are essential contributors in software development projects [1], and often present different productivity

ESEM '16, September 08 - 09, 2016, Ciudad Real, Spain

ACM 978-1-4503-4427-2/16/09…$15.00
DOI: http://dx.doi.org/10.1145/2961111.2962626

given that people are different in many ways, which also include their production capabilities [7]. Nevertheless, there are fewer studies on the productivity of software developers that focused on the study of developer's productivity itself, not on metrics to evaluate their productivity, neither on the effect of introducing new tools and processes to enhance their productivity [11].

Regarding the focus on tools, Atkins *et al*. [3] carried out a case study on developer's productivity. The authors used data from a version control system to identify improvements in developers' productivity. Kersten and Murphy [9] presented a mechanism (IDE plugin) that captures the task context of a developer's work and reduces his/her context switch. The authors showed that such reduction could make programmers more productive.

Regarding developer productivity metrics, Petersen [15] performed a systematic review on the measurement and prediction of software development productivity. He found five studies on quantifiable approaches, such as metric spaces and data envelopment analysis. Hernández-López *et al*. [7] also reviewed software engineering productivity in different development roles. This work investigated, among several jobs in software engineering, the inputs and outputs used to measure the programmer's productivity. The authors concluded that, concerning developers, the most adopted metrics were: Source Lines of Code(SLOC)/time and Tasks Completed/time.

Finally, regarding the interpretation of a developer's productivity, we identified some related studies on meanings and perceptions of productivity. Melo *et al*. [10] investigated agile teams perceptions of factors impacting their productivity. Hernández-López *et al*. [6] looked at the meanings of productivity from the point of view of software engineers through an exploratory qualitative study. Meyer *et al*. [11] also investigated software developers' perceptions regarding their productivity using a survey and an observational study. Such studies are different from ours because they investigate productivity based on the developers' perceptions of themselves while our approach is based on managers' perceptions of the developers. In order to avoid preempt confirmation bias during our study, these previous studies were fully read only after we finished our own results analysis. In Section 5, we present a comparison between our results and the results from those studies. The next section will present our research question and the adopted research process.

3. GOALS AND METHODS

We investigate how the organization understands, identifies and uses the perception of productivity of its developers. We chose software managers to represent the organizational point of view about developer's productivity, as these practitioners are responsible for setting the direction for software developments projects in the organization. We intend to answer the following research questions – all from the perspective of software managers:

RQ1 What is developer's productivity according to software managers?

RQ2 How do software managers identify the developer's productivity?

3.1 Case Study

We carried out an exploratory case study to assess how developer's productivity is understood and utilized by software project managers. Case studies are suitable tools to investigate a phenomenon within its real-life context, especially when the boundary between the phenomenon and the context cannot be clearly specified [16].

We conducted this case study in three Brazilian software organizations. The first organization (Org. 1) is responsible for development of software solutions to manage and control the State's budget execution. The second organization (Org. 2) is a state-owned company responsible for information systems developed in various areas of state's operations, such as: public education, public security, public health, human resources and government planning and administration. The third organization (Org. 3) is a non-profit private foundation, that is a technological center for research and development of projects in the areas of software and hardware, technology training and social responsibility for national and international institutions. These organizations are representative of typical software development organizations in the region. They have similar characteristics, which are presented in Table 1.

Table 1: Summary of characteristics of the selected cases.

Characteristic	Org. 1	Org. 2	Org. 3
# Employees	*78*	*392*	*260*
Social Nature	*Public*	*Public*	*Private Foundation*
Type of Software Developed	*Information systems*	*Information systems*	*Information systems*
Domains	*Government Administration*	*Public Health, Government Administration*	*Industrial automation*
Software Technologies	*Java*	*Java, Natural, PHP*	*Java, Android, iOS*
Development Process	*Prescriptive*	*Prescriptive*	*Prescriptive and Agile*

3.2 Data Collection and Analysis Procedures

We collected data from each organization using semi-structured interviews. The first author of this paper carried out the data collection, and others authors participated in activities related to the analysis and review of the findings, detailed later. We signed a non-disclosure agreement with all organizations in order to allow the interviews with their software managers.

In each organization, we first interviewed the top manager of software development. After the interview with each top manager, he indicated the experienced software managers for the other interviews. A total of 12 software managers were interviewed across the three organizations – 3 at Org. 1, 5 at Org. 2, and 4 at Org. 3.

The Software managers who participated in this research are listed in Table 2. For each participant in Table 2, we present his/her id, organization, manager position title, and education degree. Column *Years* represents the number of years that manager has been working in his/her organization. As we can see, most of the managers have been working for more than 10 years (*p8* is an exception) in their organizations, which indicates that the interviewees are very experienced in their organizations. All participants have a degree in Computer Science, with exception of participants p1 and p5, that have a degree in Economy and Accounting, respectively.

Table 2: Software managers who participated in our study.

Id	Org.	Position Title	Education	Years
p1	1	Director of Department	Bachelor	23
p2	1	SW Dev. Manager	Bachelor	10
p3	1	SW Dev. Coordinator	Bachelor	20
p4	2	Technical Director	Master	14
p5	2	Chief of SW Department	Bachelor	32
p6	2	SW Dev. Manager	Bachelor	10
p7	2	SW Quality Manager	Bachelor	18
p8	2	SW Business Manager	Bachelor	5
p9	3	Portfolio Manager	Bachelor	16
p10	3	SW Dev. Manager	Bachelor	13
p11	3	SW Quality Manager	Master	13
p12	3	SW Dev. Coordinator	Bachelor	10

All interviews used a semi-structured script and were organized into two parts: part I consisted of some introductory questions about participants' background, as per the guidelines proposed by [16]; part II (see Table 3) comprised the main interview questions. For RQ1, we explored two different points of views in order to better distinguish beliefs based on personal convictions from based on organization practices and guidelines. Thus, we asked the managers for personal and organization point of view about the concept of developer's productivity. For RQ2, we also explored different points of view: we asked directly how they perceive that a developer is productive, and how they compare the productivity of developers in the same team. All the interviews' transcriptions and analysis were carried out in Portuguese, the native language of the authors and interviewees.

The interviews were done simultaneously with the analysis process. We employed a cycle of interviews, transcriptions, and analysis. Two researchers performed the transcriptions, as suggested also by Runeson *et al*. [16], which contributed for a better understanding of the collected data. The interview transcriptions were imported into the Atlas.ti[1] software.

Table 3: The semi-structured script used.

Id	*Question*
1	*In your opinion,* *what is software developer productivity?*
2	*For your organization,* *what is software developer productivity?*
3	*How does your organization* *perceive that a developer is productive?*
4	*How does your organization* *compare the productivity of the developers from a team?*

The qualitative analysis performed in this study used Grounded Theory (GT) procedures. From GT we used open coding, axial coding, but not selective coding. Open coding involves the breakdown, analysis, comparison, conceptualization, and the categorization of the data. Axial coding examines the relations between the identified categories. Finally, selective coding performs all the process refinements by identifying the core category to which all others are related. We decided not to select a core category herein because a Grounded Theory rule is the circularity between the collection and analysis stages until the theoretical saturation is reached [17]. Therefore, we decided to postpone the selective coding phase. For this reason, we do not claim that we applied the GT method, only some specific procedures.

The first author did the open coding, associating codes with quotations of transcripts, and axial coding, where the codes were merged and grouped into more abstract categories. Once prepared, the codes and identified networks (memos showing the relationships in the categories) were reviewed, analyzed and changed upon agreement with the others researchers. After that, a new round of interviews, transcriptions and analysis started. This process cycle was repeated until no further participants were available among the set of participants provided by the organizations.

[1] http://atlasti.com/

4. RESULTS

In this section, we report our findings. They are presented as related to each research question. Additionally, we also complemented our observations with participants' quotations. To refer to a participant, we will use its identification (*p1, p2 ... p12*) presented in Table 2.

4.1 What is developer's productivity according to software managers?

Task delivery on time is a factor of developer's productivity definition that was the most mentioned by software managers. Although some software managers mentioned shortest task completion time as a factor of productivity definition, delivering on time is enough for most of them. However, the quantity of tasks alone is not taken as the productivity of developers, since the complexity of the task influences that perception: the more complex the task is, the more productive the developer is if he completes it.

"Productivity is to be able to do the programming tasks, that were requested, on time" – p3

"Productive developers are the ones that deliver their work on time*" – p9*

"The only thing that can moderate our perception about developers' productivity is the complexity of the task that was assigned to them" – p7

We observed that **meeting stakeholder's expectations** was considered by software managers as another factor of their definition of developers' productivity. Software manager's expectation is associated with fulfilling the "promise" made by the developers about their tasks. The quality of the task delivered by the developer must also ensure customer satisfaction. Thus, developers that meet stakeholder's expectations are seen by their manager as productive.

"Productivity in programming is to do what was promised to be done" – p8

"[Developer's productivity] is to deliver on time, with quality and ensuring the customer satisfaction" – p12

Developer qualification and behavior was another factor observed in the developer's productivity definitions gave by the

software managers. Software managers have a different expectation of productivity from each developer according to their technical qualification and behavior. Developers with more technical qualifications are believed to be more productive. Developer behaviors, such as focus and commitment, were also factors associated with software managers' definitions of developer's productivity.

"The developer's qualification is the central point." – p5

"[Developer's productivity] is the ability s/he has to be focused, to stay calm and to produce what was asked in an acceptable time" – p1

"[The developers] need to be committed, and all these aspects impact on their productivity as developers" – p11

No need for rework was another factor mentioned by software manager's about developer's productivity. Software managers understand that a productive developer must accomplish their tasks without a need to rework later. The need for excessive reworking of a developer's code after a review is taken as a signal that this developer is not being productive.

"to be productive they have to do their tasks in a way that these tasks do not require reworking" – p9

4.2 How do software managers identify the developer's productivity?

The **Developer task deliveries** are used by software managers to identify developer's productivity. Developers that frequently complete their tasks without needing rework, are perceived by software manages as highly productive.

"We observe their productivity through the demands that are delivered and their corresponding quality. If what is being delivered always has to return to rework, then this weighs negatively on the productivity of that employee" – p2

"These are the developers who are delivering more, more often, and with no need for reworking, these are the more productive" – p9

The **developer's team feedback** was also mentioned by software managers as a source to identify the developer's productivity. Managers build up this productivity perception through meetings with middle managers and team leaders. Team leaders obtain these perceptions through their own perception and by their team's feedback.

"The team leader, while delivering the demands for developers and getting back their production, ends up building in his head who he sees as more or less productive" – p7

"Basically upon feedback from teams" – p10

Finally, in line with the manager's concept of developer's productivity, the **developer behavior** is also a perceived factor. Software managers perceive that developers with focus, commitment and proactivity combined with good task deliveries are considered more productive. Therefore, software managers do not expect high productivity from developers who do not present these attitudes.

"The perception of [the developer's] productivity is very associated with non-technical, but behavioral issues" – p4

"Most of the time we perceive this more clearly by their attitudes, and not only by that, we also consider the quality of the code that is produced" – p5

4.3 Software managers' feedback

We held meetings in the organizations which collaborate with this research to get feedback on the obtained results. All software managers interviewed, except one, participated in these meetings. All the comments made from software managers in these meetings mainly confirmed and explored the results already presented. There was no comment in disagreement with the results.

5. DISCUSSION

In this section, we also discuss our findings organizing them by research questions.

5.1 What is developer's productivity according to software managers?

We identified that the managers' perceptions about developer productivity are influenced by four different factors: (1) tasks delivered on time, (2) produced artifacts that do not need rework, (3) products that meet stakeholders' expectations, and (4) personal behavior such as focus and proactivity. The first factor is concerned with the number of produced units; the second factor is a perceived product quality measure; the third factor is a quality measure from the client point of view; and the last factor is a visual perception of developer's way of working.

In Software Engineering, the concept of developer's productivity regularly used is the ratio between the number of product units produced over some unit of effort [12]. Our results show that manager's perceptions for the productivity of developers considers the number of tasks which were delivered on time. This is a type of performance measure, in which the more tasks are delivered on time, the more productive the developer is. This suggests that task delays can be a signal of bad developer's productivity. This factor was the most mentioned factor by software managers.

The quality of the tasks delivered is also considered by software managers. Beyond task delays, which are perceived during the developer's work, clues about the quality are also provided by the stakeholders. When they are not satisfied with the results of developer's work, they can inform the software manager and can also ask for a fix. In both cases, a developer's fault is discovered, and this contributes to a perception of bad productivity by the software managers.

Finally, developers' qualification and behavior are also factors of productivity considered by software managers. A software manager perceives more qualified developers as more productive. Personal behavior such as focus and commitment were also mentioned as developer's productivity factors. This perception of developer's productivity is different from the ones portrayed in the literature. This finding also supports the growing importance of human factors in Software Engineering [1].

5.2 How do software managers identify the developer's productivity?

The developer's productivity is identified by means of (1) the deliveries resulting from the developer tasks, (2) the developer's team feedback and (3) the developer's behavior. The software manager leverages these three items to identify which developers are productive and which are not.

Software manager's perception of their developer's productivity is mainly based on the tasks that they deliver. This is no surprise considering the results explained in the previous subsection. The outcomes from all tasks delivered by the developers are

continuously shaping the software manager perception about their productivity. Developer's focus and commitment associated with task deliveries without delays and quality problems positively influence the manager's productivity perception.

The manager also perceives the productivity of their developers by the feedback given by their development teams. This feedback comes from the team leaders and meetings with the development team. The managers consider as very productive the developer who is taken as role model by the remaining of the team.

Finally, software managers build a kind of confidence in their productive developers. That confidence allows them to demand tasks to these productive developers without any fear of the outcomes. This finding confirms that human factors play an important role in developer's productivity. This also suggests that more research is needed to investigate what are these human factors, and how they influence developer's productivity.

5.3 Relation to existing Evidence

Focusing on the meaning of developer's productivity, the most closely related works to ours are the ones by Melo *et al*. [10], Hernández-López *et al*. [6] and Meyer *et al*. [11]. Melo *et al*. [10] investigated agile teams' perceptions of factors impacting their productivity. Hernández-López *et al*. [6] examined the meaning of productivity from the perspective of developers. Finally, Meyer *et al*. [11] explored software developers' perceptions about their software development productivity. This subsection provides a comparison of the findings from these studies and ours findings.

Melo *et al*. [10] adopted a qualitative research approach. The main data collection methods were semi-structured interviews, and they used thematic analysis to inspect the data. They interviewed a total of 13 participants, comprising 2 project managers, 2 product owners, 8 developers and 1 scrum master. Their result for the definition of productivity was unclear for them, but included the following terms: *timeliness*, *quantity*, *quality* and *customer satisfaction*. This study differs from ours in the type of participants and their context, which was focused on agile teams.

Hernández-López *et al*. [6] also used an exploratory qualitative study to carry out their investigation. They interviewed a total of 15 participants: 4 project managers, 9 senior software engineers and 2 junior software engineers. The participants had worked an average of 2.5 years in their respective organizations. In that study, the authors investigated the definitions of productivity for software engineers, project managers, project and organization. Considering only the definition of productivity for software engineers, their result had the following terms: *work accomplished with required quality*, *solving problems*, *bugs found* and *in the specified time*. This study differs from ours in the type of participants, as we interviewed only software managers. Our results show that beyond their main terms for software engineering, software managers also consider the developer's behavior as part of their productivity concept.

Finally, Meyer *et al*. [11] applied a survey and an observational study. They carried out a survey with 379 professional software developers and an observational study with 11 professional software developers, which had an average of 13.3 years of total development experience. The results from their survey showed that participants often assessed their productivity based on their tasks or completed work items. This result is in accordance with ours when software managers consider tasks delivered on time as one of their perceived factors on developer's productivity. The observational study found out that developers prefer to organize their work to get in "the flow" (to have focus) so as to have less interruptions and context switches. In our results, software managers mentioned focus, but also commitment and proactive as behaviors that characterize a productive developer.

All these related studies described above had the same focus as ours: developer's productivity. They investigated the concept of productivity from developer points of view. Unlike these studies, we investigated the productivity's concept only from the perspective of the software managers. The factors found by these studies were included in our results, but differently from their results, managers also consider others factors such as developer's behaviors and need for reworking. Furthermore, we presented not only the factors of software managers' definitions of productivity and developer's productivity, but we also presented how software managers perceive the productivity of their developers.

6. THREATS TO VALIDITY

The main threat to the validity of this study is the generalization of our results for all organizations. Qualitative studies cannot use anything similar to a statistical argument to claim the generalization of their results. We mitigate such problem by interviewing experienced software managers from three organizations, which are representative of one Brazilian region.

Another threat is that perceptions of the interviewees could be biased towards their own beliefs. These beliefs could cause some distortions when interpreting the reality. To reduce this threat, the chosen software managers were those who had more experience within their organizations.

Another threat to the validity of our results is the possibility that the first author might have introduced his bias in the data collection process. In this respect, the analysis process of collected data was performed along with other researchers. These researchers reviewed and analyzed all the intermediate results (codes, memos). This iterative process was repeated until the end of data collection and data analysis processes. And we also held meetings with the software project managers to validate the obtained results.

7. CONCLUSION

This study presents a qualitative research investigating the developer's productivity from the perspective of software managers. Three case studies were conducted in three selected organizations containing similar contexts, thus allowing for the integration of results. In all, 12 software development managers were interviewed, including the organization's software directors, software quality managers, and software development managers, among others. Most managers had over 10 years of work experience in their organizations.

The measure of developer's productivity in Software Engineering is the number of product units produced over some unit of effort [12]. Our results indicate a different concept of productivity. The manager's perceptions are influenced by four different factors: tasks delivered on time, produced artifacts that do not need rework, products that meet the stakeholders' expectations and personal behavior such as: focus, proactivity and commitment.

Software managers have a perception of which developers are more productive even without using objective metrics. This perception is built from the history of the developer's delivered tasks. Those developers who carried out more tasks on time, positively influence their manager's perception regarding their productivity. On the other hand, if their tasks had to be reworked, then it negatively influenced their manager's perception. Therefore, tasks delivered on time without reworking are used as

the subjective productivity metric that is intuitively employed by software managers.

This qualitative study shows a perception of productivity different from traditional literature. A future work is to validate our results using a large scale survey with software managers. This study is also part of a larger research about factors that impact software developer's productivity. Based on the obtained results from this study, software developer's productivity metrics will be assessed in these organizations. Later, selected metrics will serve as a comparison basis for studying the factors that impact the developers' productivity within software organizations.

8. ACKNOWLEDGMENTS

We thank the financial support granted by SEFAZ, FAPEAM through processes numbers 062.00600/2014 and 062.00578/2014, and CAPES. We thank the researchers of USES group for their support during this study. Finally, we also thank all the organizations and individuals that contributed to this study.

9. REFERENCES

[1] Amrit, C., Daneva, M. and Damian, D. 2014. Human factors in software development: On its underlying theories and the value of learning from related disciplines. A guest editorial introduction to the special issue. *Information and Software Technology*. 56, 12 (Dec. 2014), 1537–1542. DOI= http://dx.doi.org/10.1016/j.infsof.2014.07.006.

[2] Aquino Junior, G.S. de and Meira, S.R.L. 2009. Towards effective productivity measurement in software projects. In *Proceedings of the 4th International Conference on Software Engineering Advance* (Porto, 2009). Proc 4th Int Conf Softw Eng Adv. IEEE. 241–249. DOI= http://dx.doi.org/10.1109/ICSEA.2009.44.

[3] Atkins, D.L., Ball, T., Graves, T.L. and Mockus, A. 2002. Using version control data to evaluate the impact of software tools: a case study of the Version Editor. *IEEE Transactions on Software Engineering*. 28, 7 (Jul. 2002), 625–637. DOI= http://dx.doi.org/10.1109/TSE.2002.1019478.

[4] Cheikhi, L., Al-Qutaish, R.E. and Idri, A. 2012. Software Productivity: Harmonization in ISO/IEEE Software Engineering Standards. *Journal of Software*. 7, 2 (Feb. 2012), 462–470. DOI= http://dx.doi.org/10.4304/jsw.7.2.462-470.

[5] Cruz, S.S.J.O., da Silva, F.Q.B., Monteiro, C.V.F., Santos, C.F. and dos Santos, M.T. 2011. Personality in software engineering: preliminary findings from a systematic literature review. In *Proceedings of the 15th Annual Conference on Evaluation & Assessment in Software Engineering* (Durham, 2011). Proc 15th Annu Conf Eval Assess Softw Eng. IET. 1–10. DOI= http://dx.doi.org/10.1049/ic.2011.0001.

[6] Hernández-López, A., Colomo-Palacios, R. and Garcia-Crespo, A. 2012. Productivity in software engineering: a study of its meanings for practitioners Understanding the concept under their standpoint. In *Proceedings of the 7th Iberian Conference on Information Systems and Technologies* (Madrid, 2012). CISTI '12. IEEE. 1–6.

[7] Hernández-López, A., Colomo-Palacios, R. and García-Crespo, Á. 2013. Software Engineering Job Productivity—a Systematic Review. *International Journal of Software Engineering and Knowledge Engineering*. 23, 03 (Apr. 2013), 387–406. DOI= http://dx.doi.org/10.1142/S0218194013500125.

[8] Huselid, M. 1995. The impact of Human Resource Management practices on turnover, productivity, and corporate financial performance. *Academy of Management Journal*. 38, 3 (1995), 635–872. DOI= http://dx.doi.org/10.2307/256741.

[9] Kersten, M. and Murphy, G.C. 2006. Using task context to improve programmer productivity. In *Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering* (New York, NY, 2006). FSE '06. ACM Press. 1. DOI= http://dx.doi.org/10.1145/1181775.1181777.

[10] Melo, C., Cruzes, D.S., Kon, F. and Conradi, R. 2011. Agile Team Perceptions of Productivity Factors. In *Proceedings of the 2011 Agile Conference* (Salt Lake City, UT, Aug. 2011). AGILE '11. IEEE. 57–66. DOI= http://dx.doi.org/10.1109/AGILE.2011.35.

[11] Meyer, A.N., Fritz, T., Murphy, G.C. and Zimmermann, T. 2014. Software developers' perceptions of productivity. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (New York, New York, USA, 2014). FSE '14. ACM Press. 19–29. DOI= http://dx.doi.org/10.1145/2635868.2635892.

[12] Mockus, A. 2009. Succession: Measuring transfer of code and developer productivity. In *Proceedings of the 2009 IEEE 31st International Conference on Software Engineering* (Vancouver, BC, May 2009). ICSE '09. IEEE. 67–77. DOI= http://dx.doi.org/10.1109/ICSE.2009.5070509.

[13] Oxford Dictionaries: 2016. *http://www.oxforddictionaries.com/us/definition/english/productivity*. Accessed: 2016-02-25.

[14] Paiva, E., Barbosa, D., Lima, R. and Albuquerque, A. 2010. Factors that Influence the Productivity of Software Developers in a Developer View. In *Innovations in Computing Sciences and Software Engineering*, T. Sobh and K. Elleithy, eds. Springer Netherlands. 99–104. DOI= http://dx.doi.org/10.1007/978-90-481-9112-3_17.

[15] Petersen, K. 2011. Measuring and predicting software productivity: A systematic map and review. *Information and Software Technology*. 53, 4 (2011), 317–343. DOI= http://dx.doi.org/10.1016/j.infsof.2010.12.001.

[16] Runeson, P., Host, M., Rainer, A. and Regnell, B. 2012. *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, Hoboken, NJ.

[17] Strauss, A. and Corbin, J. 2008. *Basics of qualitative research: Techniques and procedures for developing grounded theory, 2nd ed.* SAGE publications, Thousand Oaks, CA.

[18] Trendowicz, A. and Münch, J. 2009. Factors Influencing Software Development Productivity - State of the Art and Industrial Experiences. *Advances in Computers*. 77, 09 (2009), 185–241. DOI= http://dx.doi.org/10.1016/S0065-2458(09)01206-6.