

# How is Effort Estimated in Agile Software Development Projects?

TINA SCHWEIGHOFER, University of Maribor  
 ANDREJ KLINE, msg life odateam d.o.o.  
 LUKA PAVLIČ, University of Maribor  
 MARJAN HERIČKO, University of Maribor

---

Effort estimation is an important part of every software development project. Regardless of whether the development disciplines are traditional or agile, effort estimation attempts to systematically attach to other development elements. It is important to estimate the work load at the very beginning, despite the initial drawback of there being very little known about the project. And if, in addition, the effort estimations are accurate, they can contribute a lot to the success of a project being developed.

There are many approaches and methods available for performing effort estimation, each one with their own features, as well as pros and cons. Some of them are more appropriate for traditional software development projects, while others are meant for agile software development projects. The latter is also the subject of the systematic literature review presented in the article. Based on the set research questions, we researched the area of effort estimation in an agile software development project. Which methods are available, how objective the estimation is, what influence the estimation and most importantly how accurate those methods and approaches are. The research questions were answered and the basis for future empirical work was set.

Categories and Subject Descriptors: D.2.8 [Software Engineering] Metrics—Performance measures; Process metrics; Product metrics

Additional Key Words and Phrases: Effort Estimation, Estimation Accuracy, Agile, Software Development, SLR

---

## 1. INTRODUCTION

Effort estimation is the first of many steps in the software development process that can lead to a successful project's completion. It is a complex task, that constitutes the basis for all subsequent steps related to planing and management.

Effort estimation is also a very important part in agile software development projects. In order to achieve the highest possible levels of accuracy, software development teams can make use of different techniques, methods and approaches, including advising group effort estimation [Molokken and Jorgensen 2003], subjective expert judgement [Trendowicz and Jeffery 2014] and planning poker [Cohn 2005]. In a variety of approaches and methods used, different questions arise: What are the pros and cons of approaches for effort estimation, how can different models be applied to a different development environment and specific development team, and most importantly, how accurate is effort estimation when using a specific method or approach. The average error in effort estimation is measured between

---

Authors' addresses: T. Schweighofer, Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia; email: tina.schweighofer@um.si; A. Kline, msg life odateam d.o.o., Titova 8, 2000 Maribor, Slovenia; email: andrej.kline@msg-life.com; L. Pavlič, Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia; email: luka.pavlic@um.si; M. Heričko, Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia; email: marjan.hericko@um.si

Copyright © by the paper's authors. Copying permitted only for private and academic purposes.

In: Z. Budimac, Z. Horváth, T. Kozsik (eds.): Proceedings of the SQAMIA 2016: 5th Workshop of Software Quality, Analysis, Monitoring, Improvement, and Applications, Budapest, Hungary, 29.-31.08.2016. Also published online by CEUR Workshop Proceedings (CEUR-WS.org, ISSN 1613-0073)

20 and 30 percent [Abrahamsson et al. 2011; Grapenthin et al. 2014; Jørgensen 2004; Kang et al. 2010; Haugen 2006]. Therefore, the attempt to lower the number of errors in the accuracy rate is welcome.

The research area of our work is effort estimation and its accuracy in agile software development projects. In this article, a systematic literature review is presented. The relevant literature is presented and the answers for set research questions are given. The literature review also represents the research basis for our future work – an empirical study looking into the accuracy of effort estimation in a real-life industrial environment.

The content of the work is organized as follows. First, the theoretical background about effort estimation and effort estimation in an agile software development project is presented. Next is the main research, systematic literature review, together with results and a discussion. In the end the conclusion and future work is presented.

## 2. EFFORT ESTIMATION PROCESS

Effort estimation process is a procedure in which effort is evaluated and estimation is given in the context of the amount and number of needed resources by which we are able to end project activity and deliver a service or a product that meets the given functional and non-functional requirements to a customer [Trendowicz and Jeffery 2014].

There are many reasons for performing effort estimation. As presented in [Trendowicz and Jeffery 2014] it is performed to manage and reduce project risks, for the purpose of process progress and learning within an organization, for finding basic guidelines and productivity measurement, for the negotiation of project resources and project scope, to manage project changes and to reduce the amount of ballast in project management.

When choosing the most appropriate approach for effort estimation we have to be aware that delivering accurate estimations contributes to proper project decisions. Project decisions are important for short-term observation (within the context of one project) and for long-term observation where it can encourage the progress and effectiveness of work done within a development team and organization as a whole [Grimstad et al. 2006]. According to literature on the subject, 5 percent of the total development time should be aimed at effort estimation [Trendowicz and Jeffery 2014].

When estimating effort in agile development projects we can come across different challenges. We have to make a decision about which strategy of effort estimation we need to choose, how to connect good practices of agile development with efficient effort estimation and which factors have the most influence on the accuracy of the estimated effort.

## 3. SYSTEMATIC LITERATURE REVIEW

For the purposes of finding appropriate answers to a given problem, a systematic literature review was chosen as a research method. We tried to identify, evaluate and interpret all available contributions relevant to our research area [Kitchenham and Charters 2007].

### 3.1 Research questions

Within the researched area, the following research questions were formed:

- RQ1.* Which agile effort estimation methods are addressed?
- RQ2.* How objective is effort estimation and how much of a subjective evaluation is present?
- RQ3.* Which factors most influence agile effort estimation?
- RQ4.* Which studies regarding agile effort estimations have been performed?
- RQ5.* How useful are the particular agile effort estimation methods in the agile planning process?

Table I. Search Strings and Data Sources

KW1	<i>agile AND estimation</i>	ScienceDirect	<a href="http://www.sciencedirect.com/">http://www.sciencedirect.com/</a>	DL1
KW2	<i>agile AND estimation AND planning</i>	IEEE Xplore	<a href="http://ieeexplore.ieee.org/">http://ieeexplore.ieee.org/</a>	DL2
KW3	<i>agile AND estimation AND planning AND accuracy</i>	Scopus	<a href="http://www.scopus.com/">http://www.scopus.com/</a>	DL3
KW4	<i>agile AND estimation AND management</i>	SpringerLink	<a href="http://link.springer.com/">http://link.springer.com/</a>	DL4
		ACM Digital Library	<a href="http://dl.acm.org/">http://dl.acm.org/</a>	DL5

Table II. Search results

	<b>DL1</b>	<b>DL2</b>	<b>DL3</b>	<b>DL4</b>	<b>DL5</b>
<b>KW1</b>	21	200	286	1155	97
<b>KW2</b>	7	40	45	863	22
<b>KW3</b>	2	6	6	292	7
<b>KW4</b>	6	63	49	949	34

Table III. Distribution of primary studies according to data sources and type of publication

ScienceDirect	Journal article	[Torrecilla-Salinas et al. 2015] [Mahnič and Hovelja 2012] [Jørgensen 2004] [Inayat et al. 2015]
IEEE Xplore	Conference paper	[Popli and Chauhan 2014a] [Abrahamsson et al. 2011] [Nguyen-Cong and Tran-Cao 2013] [Grapenthin et al. 2014] [Kang et al. 2010] [Haugen 2006] [Popli and Chauhan 2014b]
ACM Digital Library	Conference paper	[Usman et al. 2014]

### 3.2 Search process

Based on the proposed research questions, search strings were formed and based on formed strings, the search for primary studies was carried out in selected digital libraries. The search string and selected digital libraries are presented in Table I. For the purposes of getting more exact results we used different search restrictions. In the ScienceDirect base we only searched in the abstracts, titles and keywords, in IEEE Xplore, Scopus and ACM Digital Library, in the abstracts of studies and in SpringerLink we restricted the discipline to Computer Science. The results obtained by the restricted search are presented in Table II

### 3.3 Study selection

After the potentially relevant studies were identified in selected data sources with proposed search strings, two selection cycles were carried out. First, we reviewed the title, keywords and abstract of each study, according to inclusion factors (the study addressed ways for effort estimation in agile projects and accuracy of estimated effort according to the real effort spent) and exclusion criteria (the study is not in English or German and cannot be found in digital libraries). The studies that were identified as appropriate were reviewed as a whole and a final decision was made, whereby we selected the studies that provided answers to research questions.

After the first selection cycle, 40 primary studies were selected and after the second selection cycle 12 relevant primary studies were selected for further analysis. Among primary studies, 4 of them were journal articles and 8 of them are conference papers published in conference proceedings. A detailed distribution with associated data sources and references is presented in Table III.

## 4. RESULTS AND DISCUSSION

### 4.1 Methods for effort estimation in agile software development

Many effort estimation methods in agile software development can be found. Among the found methods and techniques, the majority used subjective expert effort estimation. This includes techniques such as planning poker, expert judgement and story points [Usman et al. 2014; Mahnič and Hovelja 2012;

Nguyen-Cong and Tran-Cao 2013; Torrecilla-Salinas et al. 2015; Jørgensen 2004; Popli and Chauhan 2014b; Haugen 2006; Popli and Chauhan 2014a]. Additionally, planning poker as a estimation method should be used in a controlled environment, with no boundary conditions, in a known domain and within a team where anyone can and dares to express their opinion.

Estimation by analogy was also frequently used [Abrahamsson et al. 2011; Grapenthin et al. 2014; Torrecilla-Salinas et al. 2015; Jørgensen 2004; Kang et al. 2010; Popli and Chauhan 2014b; Haugen 2006], done with the help of knowledge base [Nguyen-Cong and Tran-Cao 2013; Torrecilla-Salinas et al. 2015; Popli and Chauhan 2014b]. In that context, the tool that supports the approach can be established and maintained. It can be used for data input about conducted estimation cases and also the retrospective of the conducted estimation. The database can be used by the broader community of evaluators over an extended time period, which allows for the comparison of present estimation cases to examples of estimations and time used in an actual project from the related domain.

On the other hand, other methods and techniques used for effort estimation that are not based on experimental judgement or a group's approach to assessing are not so frequently used in agile software development projects. Those techniques are, for example, COCOMO (Constructive cost model), SLIM and regression analysis [Usman et al. 2014; Nguyen-Cong and Tran-Cao 2013; Torrecilla-Salinas et al. 2015; Jørgensen 2004; Popli and Chauhan 2014b].

After the detailed analysis, we did not find an answer as to how successful and efficient the use of proposed methods and techniques is.

**4.1.1 RQ1 - Which agile effort estimation methods are addressed?** Most commonly used are different point methods, like story and functional points, user stories and expert judgements. Also, good practices of agile development, like pair programming, planning games, documentation in the form of user stories and other things that significantly contribute to the accuracy and quality of effort estimation, need to be properly taken into consideration.

## 4.2 The objectiveness of effort estimation

In selected primary studies, the objectiveness of effort estimation is identified in different ways. One of the reasons for this is that studies are carried out in different environments, whereas only five of them present real-life industrial cases [Abrahamsson et al. 2011; Grapenthin et al. 2014; Jørgensen 2004; Kang et al. 2010; Haugen 2006]. Additionally, not a lot of knowledge can be found about the success of a project from the aspect of accuracy of effort estimation. Therefore, the objectiveness of effort estimation is measured mainly according to subjective opinions of participants.

Estimation represent a subjective expert judgement [Usman et al. 2014]. Regardless of the used method for effort estimation, the final decision is made by one or more participants. Thus, they need to be experienced in the area in which they are performing estimation. As claimed in [Jørgensen 2004], subjective expert judgement is usually more accurate as formalized estimation models. Also, in domains where the team is experienced, the effort estimates are more reliable [Haugen 2006].

Authors [Inayat et al. 2015] are looking into multi-functional teams where experiences and knowledge of domain is distributed. Customer-on-site for prompt and on-going evaluation of the work done, lead to higher estimation accuracy [Inayat et al. 2015]. If human factors, like team experience and ability to manage projects, are on high level, then quality of estimate creation is high [Popli and Chauhan 2014a].

The approach of estimating stories, based on empirical knowledge base and using key words [Abrahamsson et al. 2011] is objective as much as entries in knowledge base from where the evaluator get their data.

4.2.1 *RQ2 - How objective is effort estimation and how much of a subjective evaluation is present?* Subjective expert judgement is the most widespread method for effort estimation [Usman et al. 2014]. It is hard to assess the accuracy of effort estimation without any statistical measurement errors [Nguyen-Cong and Tran-Cao 2013]. Otherwise, the Kalman filter algorithm that is used to track the project's progress, systematically summarizes the current errors, but the result very much depends on the given function points [Kang et al. 2010]. In conclusion, the proposed algorithm does not contribute much to the agile effort estimation.

### 4.3 Effort estimation factors

When modelling productivity and estimating project effort within software development, we need to be aware that present success does not necessarily guarantee future project success if projects are placed within a new context [Trendowicz and Jeffery 2014].

The effort needed for software development depends on factors, divided between context and scale factors [Trendowicz and Jeffery 2014; Trendowicz et al. 2008]. Context factors include: the programming language, application domain of the software, type of development and life cycle (methodology) of development. Scale factors include: the size of software, complexity of system interfaces and integration, project effort, project duration, maturity of software development process, the size and structure of the development team and the project budget [Trendowicz and Jeffery 2014].

Based on an analysis of selected primary studies influencing effort estimation were extracted. Presented factors are classified into four groups: personnel-, process-, product- and project factors, as presented in [Trendowicz and Jeffery 2014; Trendowicz and Münch 2009].

#### **Personnel factors**

Personnel factors are the characteristics of people involved in the software development project. They usually take into consideration the experience and capabilities of project stakeholders as development team members, as well as of software users, customers, maintainers, subcontractors, etc. [Trendowicz and Münch 2009]. In the analysed primary studies, the following personnel factors can be found:

- [Usman et al. 2014]: The team's previous experiences, task size, efficiency and risk level of testing, domain and task knowledge.
- [Mahnič and Hovelja 2012]: Experiences and motivation of the development team, ability to combine different-knowledge developers.
- [Torrecilla-Salinas et al. 2015]: Team size, duration of iteration, experiences gained within an iteration, the achieved value of the tasks.
- [Jørgensen 2004]: Experiences and the knowledge of experts in the development team, the ability to combine experts, willingness and ability to educate new members regarding effort estimation.

#### **Process factors**

Process factors are connected with the characteristics of software development and also with the methods, tools and technologies applied during development [Trendowicz and Münch 2009]. The following process factors are found in primary studies:

- [Abrahamsson et al. 2011]: User stories description quality.
- [Nguyen-Cong and Tran-Cao 2013]: Need for measurement of statistical errors in the form of MRE and MMRE.
- [Grapenthin et al. 2014]: Accuracy of requirement knowledge during the project.

#### **Project factors**

Project factors include various qualities of project management and organization, resource manage-

ment, working conditions and staff turnover [Trendowicz and Münch 2009]. Among primary studies, the following project factors are found:

- [Inayat et al. 2015]: The use of agile practices (for example, cooperation with the customer, testing approach, retrospective, project organization and others).
- [Kang et al. 2010]: A common understanding of what one measuring point is, changing requirements from customers, new requirements, changing priority of existing requirements.

### **Product factors**

Product factors describe the characteristics of software product being developed through all development phases. The factors refer to products as software code, requirements, documentation and others, as well as their characteristics [Trendowicz and Münch 2009]. In the analysed primary studies none of the product factors were found.

4.3.1 *RQ3 - Which factors most influence agile effort estimation?* An analysis of primary studies shows that personnel factors come before project factors in the agile effort estimation process. This means that the level of knowledge and the experiences of experts in teams and also the way in which development teams are constructed, are crucial for effort estimation. The communication can improve estimates and reduce task changes during the projects [Grapenthin et al. 2014]. It is important to use data from past tasks and control lists for evaluation. The feedback regarding estimation also needs to be given and presented to all team members [Jørgensen 2004]. Based on all the findings, we can conclude that personal factors really are more important than project factors, which is also confirmed in [Popli and Chauhan 2014a].

## 4.4 Conducted studies in the area of agile effort estimation

Among analysed primary studies, different reports about conducted studies can be found. The studies vary by their scope, by the methods used for effort estimation and in particular by the end results. Some studies [Abrahamsson et al. 2011; Grapenthin et al. 2014; Jørgensen 2004; Kang et al. 2010; Haugen 2006] present cases from an industrial environment, while other studies [Usman et al. 2014; Mahnič and Hovelja 2012; Nguyen-Cong and Tran-Cao 2013; Torrecilla-Salinas et al. 2015; Inayat et al. 2015; Popli and Chauhan 2014b; 2014a] present cases from an academic environment. It can be seen that studies repeat and that a little knowledge about the conducted project of agile development can be found.

4.4.1 *RQ4 - Which studies regarding agile effort estimations have been performed?* Many studies are available, but many of them repeat. Very little empirical knowledge is available. The accuracy of effort estimation is not very good, which can be seen by the amount of work that needs to be done and on the release date. Both evaluations are often missed as claimed in [Popli and Chauhan 2014b].

## 4.5 Methods for effort estimation in agile project planning and development

In primary studies, the most commonly used methods for effort estimation are XP [Usman et al. 2014; Abrahamsson et al. 2011; Nguyen-Cong and Tran-Cao 2013; Inayat et al. 2015; Kang et al. 2010; Popli and Chauhan 2014b; Haugen 2006; Popli and Chauhan 2014a] and SCRUM [Usman et al. 2014; Mahnič and Hovelja 2012; Nguyen-Cong and Tran-Cao 2013; Grapenthin et al. 2014; Torrecilla-Salinas et al. 2015; Inayat et al. 2015; Popli and Chauhan 2014b], which can be related and explained by their popularity and understanding in an agile development society. Some other methods are also found, but they do not constitute a significant share. Those methods include, for example: RUP (Rational Unified Proces) [Nguyen-Cong and Tran-Cao 2013], Lean [Jørgensen 2004; Nguyen-Cong and Tran-Cao 2013], Hybrid method [Nguyen-Cong and Tran-Cao 2013] and Crystal [Popli and Chauhan 2014b].

Important to note is that some of the primary studies go back to the year 2002 and beyond. This causes some methods of effort estimation, now used in agile effort estimation, did not yet get their nowadays names, since agile methods were then in Europe at the beginning of recognition and use.

4.5.1 *RQ5 - How useful are the particular agile effort estimation methods in the agile planning process?* The planning poker works well when evaluating smaller tasks [Mahnič and Hovelja 2012], also user stories combined with the knowledge base give more accurate results for smaller tasks [Abrahamsson et al. 2011]. As mentioned in [Torrecilla-Salinas et al. 2015] function points and summary of time used is presented in the context of an agile web project together with an empirical report. It is important to adjust to the latest findings regarding effort estimation, which means continuous learning [Torrecilla-Salinas et al. 2015]. Using agile development practices can contribute to higher accuracy in effort estimation [Inayat et al. 2015].

## 5. CONCLUSION

The area of effort estimation in agile software development was researched based on the proposed research questions. Many studies can be found in different articles, but a lot of them repeat themselves. On the other hand, there are also quite a few articles that provide empirical knowledge about effort estimation. This is especially surprising since agile software development methods emerged as early as the year 2000 (in Europe).

As data extraction within a systematic literature review shows, the most widely used estimation methods are user card points, story points and functional points, user stories and experience-based experts' estimation. The authors' experience set personnel factors ahead of project factors, which means that for the estimation, the knowledge and skill level of the group experts is essential, as well as the ability to form proper working teams. Also, the studies prove that the usage of agile practices in the software development process, such as working in pairs (concept creation, testing, refactoring), planning games, user stories as well as documentation, all lead to a higher quality of effort estimation, and thus to more accurate estimations.

In the article [Nguyen-Cong and Tran-Cao 2013], many presented effort estimation models are not empirically proven and objectivity is measured from the viewpoint of the accuracy of effort estimation. Basically, this can be widespread to the others primary studies. The presented studies usually present a narrow business area, where many restrictions are pointed out, for example restrictions about the development team size, profile of evaluators, techniques used and, primarily, the duration of the presented studies. The duration is usually not long enough for an objective evaluation of effort estimation method usefulness.

The conducted systematic literature review did disclose a lot of options for future work. Among other reasons the review was conducted for the purpose of setting a theoretical background for an empirical study that will be done. The study will look into the accuracy of effort estimation in a real life industrial environment. It will track the estimation accuracy of user stories in an environment, where twenty five developers using extreme programming discipline, are tracked for a period of two years and measured by their software development effort estimation accuracy. Also, future work will be oriented towards an attempt to improve accuracy by creating a knowledge database of elapsed user stories and a retrospective of similar work done in the past. This was already mentioned by some authors in the primary studies that proposed the introduction of a knowledge base to be used in the effort estimation process. This was one of the triggers for carrying out an empirical study that will present and introduce such a knowledge base in a real life industrial environment.

## REFERENCES

- P. Abrahamsson, I. Fronza, R. Moser, J. Vlasenko, and W. Pedrycz. 2011. Predicting Development Effort from User Stories. In *2011 International Symposium on Empirical Software Engineering and Measurement*. 400–403. DOI: <http://dx.doi.org/10.1109/ESEM.2011.58>
- Mike Cohn. 2005. *Agile Estimating and Planning*. Prentice Hall PTR, Upper Saddle River, NJ, USA.
- S. Grapenthin, S. Poggel, M. Book, and V. Gruhn. 2014. Facilitating Task Breakdown in Sprint Planning Meeting 2 with an Interaction Room: An Experience Report. In *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*. 1–8. DOI: <http://dx.doi.org/10.1109/SEAA.2014.71>
- Stein Grimstad, Magne Jørgensen, and Kjetil Moløkken-Østfold. 2006. Software effort estimation terminology: The tower of Babel. *Information and Software Technology* 48, 4 (2006), 302 – 310. DOI: <http://dx.doi.org/10.1016/j.infsof.2005.04.004>
- N. C. Haugen. 2006. An empirical study of using planning poker for user story estimation. In *AGILE 2006 (AGILE'06)*. 9 pp.–34. DOI: <http://dx.doi.org/10.1109/AGILE.2006.16>
- Irum Inayat, Siti Salwah Salim, Sabrina Marczak, Maya Daneva, and Shahaboddin Shamshirband. 2015. A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior* 51, Part B (2015), 915 – 929. DOI: <http://dx.doi.org/10.1016/j.chb.2014.10.046> Computing for Human Learning, Behaviour and Collaboration in the Social and Mobile Networks Era.
- M. Jørgensen. 2004. A review of studies on expert estimation of software development effort. *Journal of Systems and Software* 70, 1–2 (2004), 37 – 60. DOI: [http://dx.doi.org/10.1016/S0164-1212\(02\)00156-5](http://dx.doi.org/10.1016/S0164-1212(02)00156-5)
- S. Kang, O. Choi, and J. Baik. 2010. Model-Based Dynamic Cost Estimation and Tracking Method for Agile Software Development. In *Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on*. 743–748. DOI: <http://dx.doi.org/10.1109/ICIS.2010.126>
- B. Kitchenham and S. Charters. 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Technical Report. School of Computer Science and Mathematics Keele University, University of Durham.
- Viljan Mahnič and Tomaž Hovelja. 2012. On using planning poker for estimating user stories. *Journal of Systems and Software* 85, 9 (2012), 2086 – 2095. DOI: <http://dx.doi.org/10.1016/j.jss.2012.04.005> Selected papers from the 2011 Joint Working IEEE/IFIP Conference on Software Architecture (WICSA 2011).
- K. Molokken and M. Jorgensen. 2003. A review of software surveys on software effort estimation. In *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*. 223–230. DOI: <http://dx.doi.org/10.1109/ISESE.2003.1237981>
- Danh Nguyen-Cong and De Tran-Cao. 2013. A review of effort estimation studies in agile, iterative and incremental software development. In *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2013 IEEE RIVF International Conference on*. 27–30. DOI: <http://dx.doi.org/10.1109/RIVF.2013.6719861>
- R. Popli and N. Chauhan. 2014a. Agile estimation using people and project related factors. In *Computing for Sustainable Global Development (INDIACom), 2014 International Conference on*. 564–569. DOI: <http://dx.doi.org/10.1109/IndiaCom.2014.6828023>
- R. Popli and N. Chauhan. 2014b. Cost and effort estimation in agile software development. In *Optimization, Reliability, and Information Technology (ICROIT), 2014 International Conference on*. 57–61. DOI: <http://dx.doi.org/10.1109/ICROIT.2014.6798284>
- C.J. Torrecilla-Salinas, J. Sedeño, M.J. Escalona, and M. Mejías. 2015. Estimating, planning and managing Agile Web development projects under a value-based perspective. *Information and Software Technology* 61 (2015), 124 – 144. DOI: <http://dx.doi.org/10.1016/j.infsof.2015.01.006>
- Adam Trendowicz and Ross Jeffery. 2014. *Software Project Effort Estimation: Foundations and Best Practice Guidelines for Success*. Springer Publishing Company, Incorporated.
- Adam Trendowicz and Jürgen Münch. 2009. Chapter 6 Factors Influencing Software Development Productivity State of the Art and Industrial Experiences. *Advances in Computers*, Vol. 77. Elsevier, 185 – 241. DOI: [http://dx.doi.org/10.1016/S0065-2458\(09\)01206-6](http://dx.doi.org/10.1016/S0065-2458(09)01206-6)
- Adam Trendowicz, Michael Ochs, Axel Wickenkamp, Jürgen Münch, and Takashi Kawaguchi. 2008. Integrating Human Judgment and Data Analysis to Identify Factors Influencing Software Development Productivity. (2008).
- Muhammad Usman, Emilia Mendes, Francila Weidt, and Ricardo Britto. 2014. Effort Estimation in Agile Software Development: A Systematic Literature Review. In *Proceedings of the 10th International Conference on Predictive Models in Software Engineering (PROMISE '14)*. ACM, New York, NY, USA, 82–91. DOI: <http://dx.doi.org/10.1145/2639490.2639503>