# The Agile Practices Impact Model

## Idea, Concept, and Application Scenario

Philipp Diebold
Fraunhofer IESE
Fraunhofer-Platz 1, Kaiserslautern, Germany
philipp.diebold@iese.fraunhofer.de

Thomas Zehler
Fraunhofer IESE
Fraunhofer-Platz 1, Kaiserslautern, Germany
thomas.zehler@iese.fraunhofer.de

## ABSTRACT

Because agile development has become more important in software engineering in recent years, many companies want to become agile. One way to do so is context-specific improvement, preferably performed by selecting the right agile practices and integrating them into the current software development process. For making an appropriate selection, the impact on the improvement goals needs to be known. Therefore, the idea is to build an overall model that includes the impact of at least the most commonly used agile practices. This impact model is mainly built on these agile practices, on various impact characteristics, and on their connections. A larger example of some practices and their (possible) impact characteristics is presented. Additionally, the Agile Capability Analysis is introduced as an example application scenario of the model. The Agile Practices Impact Model presented in this paper could support context-specific integration of agility into any current development process. Nonetheless, the model needs to be filled with existing evidence.

## Categories and Subject Descriptors

D.2.9 [**Software Engineering Management**]: Software process models.

## General Terms

Documentation, Measurement.

## Keywords

Agile Practices Impact, Agile Practices, Impact Model, Causal Model, Agile Capability Analysis.

## 1. INTRODUCTION AND MOTIVATION

In the field of software processes, agile software development has been the dominating paradigm in recent years [19], with its agile methods and more specific agile practices. In contrast to many other publications showing the direct integration and adaptation of agile methods [6], this work focuses on software process improvement (SPI) based on the integration of single agile practices. These practices are more appropriate for context-specific SPI [7] because single agile practices are the most fine-granular element and thus can be more easily used to extend, adapt, or change the current software development process.


*ICSSP'15*, August 24–26, 2015, Tallinn, Estonia.

DOI: http://dx.doi.org/10.1145/2785592.2785609

Usually an improvement initiative is aligned with organizational improvement goals, e.g., "increasing time to market" or "improving the product quality by reducing the number of defects". Therefore, this context-specific and goal-oriented SPI is confronted with the problem of selecting the most appropriate agile practices to improve the current development process according to the goals. To find out whether specific practices are appropriate or not, the positive impact (benefit) of all single agile practices needs to be known, e.g., by having a connection between the improvement goals or problems and the agile practices.

Most often, variations of agile methods [6] and their impact on possible improvement goals are published. But these do not provide details about which elements of an agile method "really" led to the improvement. As there is a need for a deeper understanding of the effect of agile practices and an information about their impact has only been published for a few agile practices, which is not enough to provide detailed decision support with regard to the selection of appropriate agile practices for systematic SPI.

Neither in research nor in industry do we currently find the required (central) point with the information about which agile practice has which specific impact on different aspects. Therefore, this contribution introduces the Agile Practices Impact Model to represent this connection for improving context-specific SPI using agile practices.

The paper is organized as follows: In Section 2, we present some related work regarding known impacts of agile practices, causal and impact models, and experience and knowledge management. The Agile Practices Impact Model is explained in Section 3, including the general structure, possible improvement goals, and an example of some practices and related improvement goals. Section 4 presents the idea of Agile Capability Analysis as one application of the impact model. Finally, in Section 5, we summarize the paper, draw conclusions, and provide an outlook on future work regarding the Agile Practices Impact Model.

## 2. RELATED WORK

### 2.1 Impact of Agile Practices

No comprehensive impact model is available or has been published that covers a set of or even all agile practices. Nonetheless, some aspects from the existing literature are partly related to our work and were used in part to develop our model.

The only existing related work that goes in the direction of our idea of an Agile Practices Impact Model is [22], which includes two case studies describing several impacts. For us, the second one was important, where some of the Scrum elements, e.g., sprint length, sprint review meeting, cross-functional teams, and daily meetings, were graphically connected with different impacts, such as "reduce overtime", "increase productivity", and "improved teamwork".

In addition to this work regarding the impact of agile practices, little has been published about specific agile practices. Pair Programming, for example, is the practice with the best-known impact because there are several publications dealing with this topic, either in general [13], [1] or in specific contexts, e.g., [18]. In part, there are also common combinations of two agile practices that are also analyzed regarding a very specific impact, e.g., on implementation details [17]. In addition to these, there are a few other examples of agile practices in use: Planning Poker [11], Customer Involvement [12], and User Stories [9].

But these aspects were the only aspects we found regarding the impacts of agile practices. Because there is no impact model dealing with agile practices, the next part of the related work deals with causal and impact models.

## 2.2 Causal and Impact Models

In general, an explanation of what causal models ought to explain is given in [21]. The main idea there is a model of a formal structure of the explanation. In addition, the authors also compare causal models with other models for explanations, which do not really fit our problem.

Besides this generic causal or impact model description, two models can be found in specific areas of Software Engineering: on the one hand, a model regarding the impact of global software development characteristics, like distance, different cultures, or language differences [16]; on the other hand, causal models, which are often used in the area of cost estimation. One example is the CoBRA® method, which uses a causal effort model also known as effort overhead model [23].

In addition to these SE-specific models, there are other domains such as health science [10] or market research [14] that make more use of such causal modeling approaches than SE. Especially in health, different types of causal models are used, e.g., graphical models (causal diagrams), potential-outcome (counterfactual) models, or structural-equation models. [10] presents the logical connections among these types and their strengths.

## 2.3 Experience and Knowledge Management

The idea of structured storing and sharing of information in software engineering originated in 1988 and was evolved into the experience factory approach [3]. Because the type of development changed over time, so did dealing with experience evolve into experience management and even knowledge management [20].

In general, knowledge management is used to improve decision support (mainly in large companies), especially because "learning by doing" with new approaches or technologies results in a delay [20]. Additionally, [4] shows that one main concept is to manage knowledge about software development processes, as done in CMMI as a catalog of best practices.

# 3. AGILE PRACTICES IMPACT MODEL

After this brief excursion into different related work areas, which will be helpful for understanding the following parts, the ideas, the concepts, and an example of the new Agile Practices Impact Model (APIM) are presented next.

Since no research exists that summarizes the connection between agile practices and their influence on impact characteristics, the idea is to come up with a model covering these aspects. Within this impact model, all (or at least the most common) agile practices should be combined with the possible characteristics they are influencing either in a positive or negative way. Thus, the following conceptual structure for the model was built iteratively together with two other research experts.

## 3.1 Concept and Representation

In general, the APIM is a model for representing the influence (impact) of agile practices on different characteristics. This model consists of different elements and necessary connections between these elements to present the influencing impact. How these different elements are connected is presented in the APIM meta-model in Figure 1.

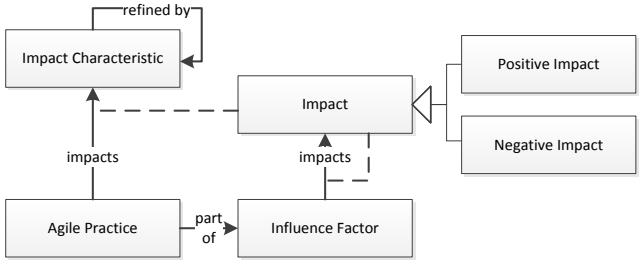


**Figure 1. APIM meta-model**

The different elements as well as the connections within the meta-model will be explained in detail in the subsequent paragraphs.

**Elements:** *Agile Practices* (AP = $\{AP_1, \ldots, AP_x\}$) are "established instructions, e.g. tasks, activities, technical aspects, or guidelines, with a specific focus or aspect in the development of software which is performed according to single or less agile core values and Agile Principles" [7]. Common examples of practices are the twelve core practices of eXtreme Programming, e.g., Pair Programming.

*Impact Characteristics* (IC = $\{IC_1, \ldots, IC_y\}$) are the different possible aspects regarding the possible impact of one or more agile practices. These characteristics can most often be linked to the (organizational) improvement goals. Chapter 3.2 provides an initial list of concrete impact characteristics and a partial hierarchy of these. The model also offers the possibility of build hierarchies of the impact characteristics, which are modeled via inheritance connections (e.g. UML). This is necessary because of the different abstraction levels of the characteristics that may be important. If the model contains more than one abstraction level of one of the impact characteristics, the connections from the agile practices can only lead to the lowest abstraction level. Nonetheless, this implicitly represents an impact on the higher level(s).

*Influence Factors* (IF = $\{IF_1, \ldots, IF_z\}$) represent any aspect that affects one or more direct influence(s) of agile practice(s) on impact characteristic(s). Lists of such factors are provided by [5], [24]. Examples could be possible organizational constraints, such as team size. Additionally, the agile practice itself can also be an influence factor.

Finally, *Impacts* represent the "value" of the two main connections: (1) the direct influence of the agile practices on the impact characteristics and (2) the indirect influence of the influence factors on the direct connections. This can generally be refined into a positive as well as a negative impact.

The impacts can be captured more concretely than in just these two categories. Similar to the CoBRA® method, the APIM contains the possibility of specifying a single value, a range of values (with maximum and minimum), or even a probability distribution for the impact. However, these aspects are not covered in the graphical representation of the APIM, which will be shown in the example later on.

**Connections:** The main connections of this model are *unidirect connections* from the agile practices to the impact characteristics. Each connection represents the impact of exactly one agile practice on one impact characteristic (1:1-connection).

In addition to this direct connection, *indirect connections* are also possible. These connections from an influence factor to a direct connection (from AP to IC) represent an external effect on the respective direct impact. There is one specific kind: the possibility that another agile practice will influence a direct impact.

All the different connections can represent either a positive ("+") or negative ("-") impact on the specific characteristic or direct connection. Nonetheless, this plus and minus are just in the graphical representation and would be representative for real values / ranges as explained before.

## 3.2 Impact Characteristics

As mentioned in the previous chapter, the impact characteristics are an important element of the APIM. To facilitate understanding, this section will provide an initial list of possible impact characteristics elicited and structured by the authors.

Since these characteristics are strongly connected with desired organizational (process) improvement goals, we came up with this initial list from internal workshops and our experience with different industrial customers. Additionally, this list was enhanced by literature found on this topic, e.g. standards [8] that provide input.

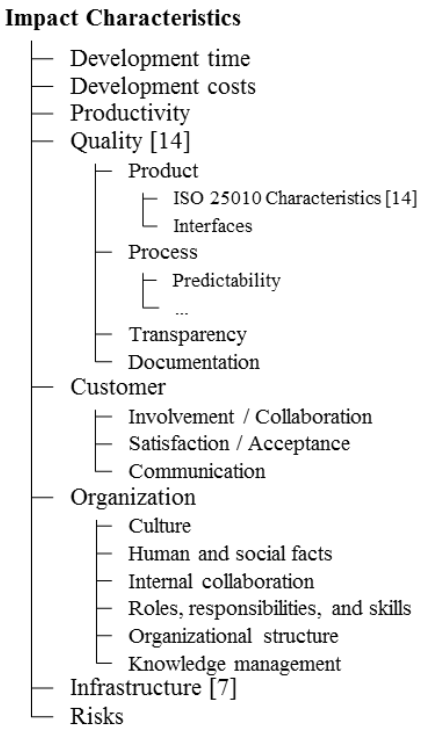


**Figure 2. Hierarchical structure of Impact Characteristics**

The preliminary list of impact characteristics presented in Figure 2 includes a hierarchy of sub-characteristics. This was necessary in order to combine characteristics on different levels of abstraction. Some impact characteristics cannot be refined into sub-characteristics (e.g., productivity or risks); for others, we decided to refine them where it made sense, e.g., quality into process quality, product quality, transparency, and documentation quality.

## 3.3 Example

An example graphical representation of the APIM is shown in Figure 3. It is a first example of the filled APIM with eight agile practices that influence four high-level impact characteristics of the list from Section 3.2. This representation shows the relations between the different elements by placing the agile practices on the right and the impact characteristics on the left (Figure 3). The impacts between the elements are shown by the different arrows (solid for direct and dashed for indirect) and their impact orientation (positive or negative) by the sign in the circles. For example, Figure 3 shows the positive impact of code reviews on quality and the concurrent negative impact on development time.

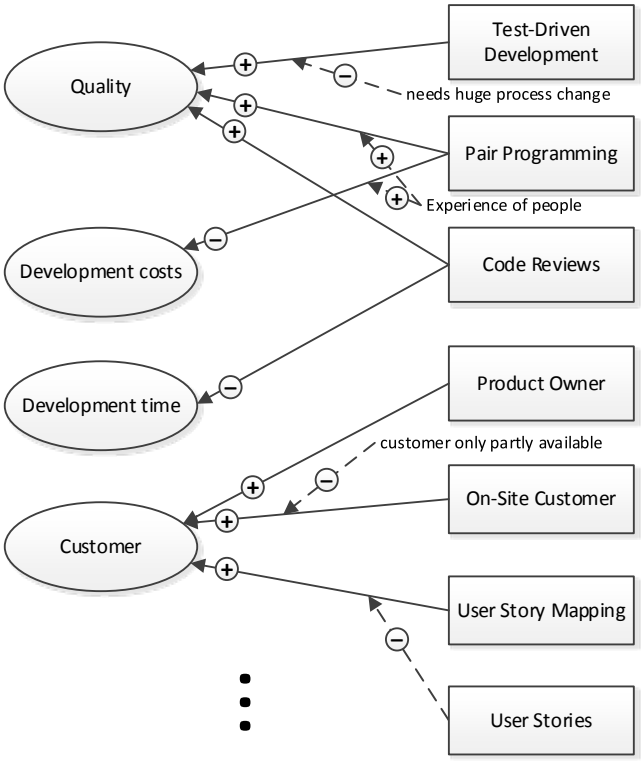


**Figure 3. Example of the APIM**

## 3.4 Discussion and Lessons Learned

During the development of the APIM as well as the creation of the example in Figure 3, we identified some lessons learned and recognized some aspects that need to be mentioned and discussed for further elaboration.

In general, the current idea of the APIM only covers a 1:N relationship from a single agile practice to one or more impact characteristics. But it was not developed to cover the aspect of the impact of a set of agile practices (M:N), which would be interesting for different combinations, especially for fixed sets of practices, namely agile methods.

One of the most important and critical aspects that needs to be discussed is how to get or collect the specific impacts, because this would requires hard measurement data in the ideal case. As provided in the related work section, a systematic literature review (SLR) has only been conducted for a few agile practices, so there is little objective data. Since for the (initial) collection of impacts, performing SLRs for all common agile practices is unfeasible, we thought of getting a wide basis by collecting different kinds of evidence, such as subjective evidence, e.g. with expert data. However, this makes it even harder to come up with quantifiable

data for the impact in order to distinguish between a strong and a weak impact. For these different kinds of evidence, a confidence level with the reference needs to be allocated to the impact so that the users of the model can decide for themselves whether they trust this evidence or not.

The different kinds of possible impacts also led to reflections on how to summarize and deal with different values and/or ranges of impacts. Additionally, all these aspects need to be unified under one single positive or negative impact (represented by arrow).

Furthermore, Figure 3 with only 12 elements (eight agile practices and four impact characteristics) shows a certain complexity, whereas it only includes three indirect connections from influence factors as an example. Thus, we assume that there will be a problem with the visualization and representation of the complete model if all agile practices are used.

To support the full implementation of the APIM with all common agile practices and their impacts and to address the two issues mentioned above, we started developing tool support based on an Enterprise Architect (EA) plugin. On the one hand, this should support filling, changing, or adapting the model with the necessary information. On the other hand, the benefit is that more semantic and other information is added to the model. This is the case because within such a modeling tool, it is possible to add structured semantic information to all elements and connections, such as percentage values, value ranges, or distributions to the positive or negative impact as well as reasons or references.

# 4. AGILE CAPABILITY ANALYSIS: APPLICATION OF THE IMPACT MODEL

Our main application of the APIM would be the Agile Capability Analysis. In contrast to many existing Agile Capability Models, that measure the maturity of an agile process, the Agile Capability Analysis is used to find the most appropriate agile practices to improve the current development process in accordance with a specific context. The impact model would be the central part of the Agile Capability Analysis (cf. Figure 4), with several context aspects being organized around this model to reduce complexity.

How the APIM is used in this analysis will be described in the following three steps:

Since such a capability analysis is based on one or more improvement goal(s), e.g. elicited with GQM+Strategies® [1], the first step (Figure 4, left part) is the restriction, enhancement, and prioritization of the given list of impact characteristics (cf. Section 3.2) based on their alignment with the improvement goals. This would result in a reduced prioritized set of impact characteristics.

Furthermore, regulations and other context characteristics restrict the set of agile practices in the second step (Figure 4, right part). These are characteristics such as different kinds of mandatory regulations (e.g., laws or standards) and their derived requirements, which could constrain agile practices in a positive way (Figure 4, allows / recommends / includes) and in a negative way (Figure 4, excludes). Beside the regulations, this could also include any other kind of organizational aspect, e.g., a management decision reducing the set of eligible agile practices.

The third step (Figure 4, middle part) uses the remaining contextual information and the current development process as influence factors of the APIM. Thus, most of these aspects from the concrete organizational or project-specific context influence the impact of the agile practices. For example team-aspects which are important in agile development.

Based on all these different kinds of information located around the APIM, a selection of the most appropriate agile practice(s) for a specific organizational or project context with respect to one or more improvement goals should be supported.

In addition to this concrete usage of the APIM, other application scenarios might exist. We would appreciate being contacted regarding any ideas of how to use it, even if some adaptation or change might be necessary.

# 5. CONCLUSIONS AND FUTURE WORK

After an initial motivation of the need for a model containing the impact of agile practices, the related work on the impact of existing agile practices as well as on causal and impact models was combined with related work on empirical experience or knowledge management. The main contribution of this work is the Agile Practices Impact Model (APIM) with its concepts, detailed impact characteristics, a detailed example, and lessons learned during the creation of the model and the example presented.

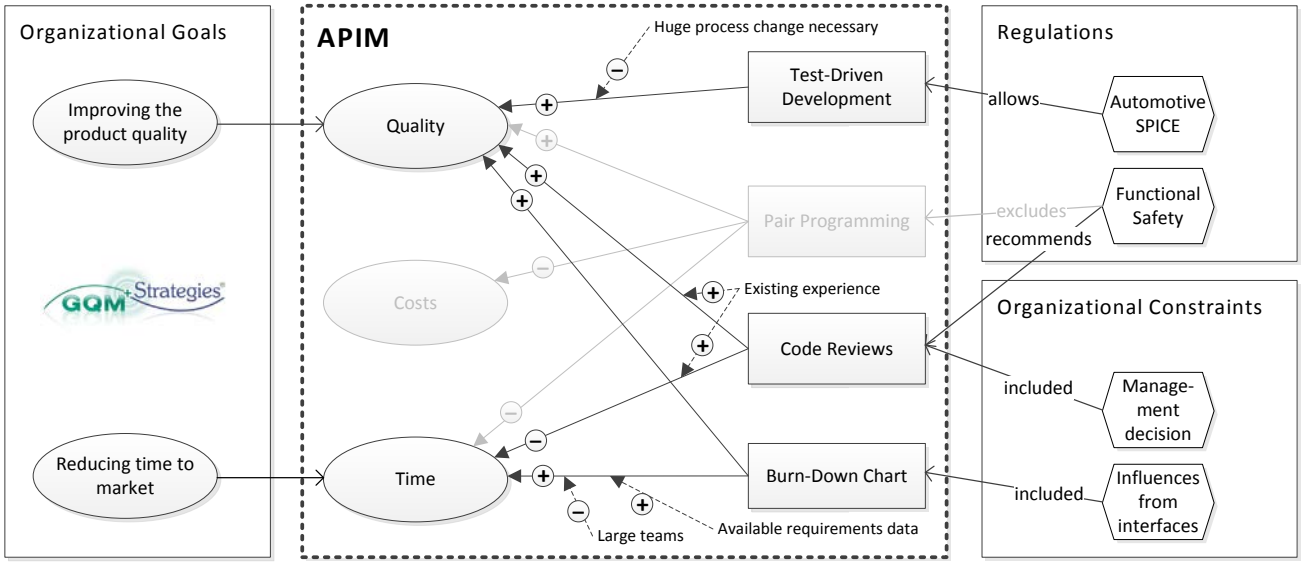


**Figure 4. Example of the Agile Capability Analysis**

The Agile Capability Analysis as an application scenario of the APIM briefly explained that such a model with respective information is necessary and could be helpful. Thus, the APIM could support the improvement of software development processes in accordance with a specific context.

Nonetheless, the model we created is only a first step towards this full idea of Agile Capability Analysis and further work needs to be done. Specifically for the APIM, future aspects will be evidence of the exhaustivity, validation of correctness and usefulness of the model and filling the model with at least the most common agile practices and impact characteristics (cf. Section 3.2). To fill this model with existing evidence and to work on the overall idea of the impact of agile practices, the "ImpAct'15"[1] workshop will be conducted. Supporting all these different aspects, the tool mentioned above will be implemented to simplify these aspects of future work.

## 6. ACKNOWLEDGMENTS

This paper is based on research being funded by the German Ministry of Education and Research (BMBF 01IS12053) in a Software Campus project. We thank Adam Trendowicz, Jens Heidrich, and Sonnhild Namingha for reviewing the model and for providing valuable feedback on the paper.

## 7. REFERENCES

[1] Arisholm, E., Gallis, H., Dyba, T. and Sjoberg, D., 2007. Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise. *IEEE Transaction on Software Engineering* 33, 2 (Feb. 2007), 65-86.

[2] Basili, V., Trendowicz, A., Kowalczyk, M., Heidrich, J., Seaman, C., Münch, J., and Rombach, D. 2014. *Aligning Organizations through Measurement: The GQM+Strategies Approach*. Springer, Germany.

[3] Basili, V. R., Caldiera, G., and Rombach, H. 2001. Experience Factory. In *Encyclopedia of Software Engineering*., J.J. Marciniak (Ed.), John Wiley & Sons, USA, 511-519.

[4] Bjørnson, F.O. and Dingsøyr, T. 2008. Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Inf. Softw. Technol.* 50, 11 (October 2008), 1055-1068.

[5] Clark, P. and O'Connor, R.V. 2012. The situational factors that affect the software development process: Towards a comprehensive reference framework. *Inf. Softw. Technol.* 54, 5 (May 2012), 433-447.

[6] Diebold, P., Ostberg, J.P., Wagner, S., and Zendler, U. 2015. What Do Practitioners Vary in Using Scrum? In *Proceedings of XP'15*. Springer, Germany, LNBIP 212, 40-52.

[7] Diebold, P. and Zehler, T. 2015. The Right Degree of Agility in Rich Processes. In *Managing Software Process Evolution*, M. Kuhrmann, A. Rausch, J. Münch, I. Richardson, and H. Zhang (Eds.), Springer, Berlin, Germany. (Accepted)

[8] Deutsches Institut für Normung e.V. 2005. *DIN EN ISO 9000:2005, Quality Management Systems – Fundamentals and Vocabulary*. Beuth Verlag, Berlin, Germany.

[9] Eocha, C. and Conboy, K. 2010. The Role of the User Story Agile Practice in Innovation. In *Proceedings of LESS'10*. Springer, Berlin, Germany, 20-30.

[10] Greenland, S. and Brumback, B. 2002. An overview of relations among causal modelling methods. *Int. Journal of Epidemiology* 31, 5 (2002), 1030-1037.

[11] Haugen, N. 2006. An Empirical Study of Using Planning Poker for User Story Estimation. In *Proceedings of AGILE'06*. IEEE Computer Society, Washington, DC, USA, 23-34.

[12] Hoda, R., Noble, J., and Marshall, S. 2011. The impact of inadequate customer collaboration on self-organizing Agile teams. *Inf. Softw. Technol.* 53, 5 (May 2011), 521-534.

[13] Hulkko, H., and Abrahamsson, P. 2005. A multiple case study on the impact of pair programming on product quality. In *Proceedings of ICSE '05*. ACM, New York, NY, USA, 495-504.

[14] Hulland, J., Chow, Y., and Lam, S. 1996. Use of causal models in marketing research: A Review. *Int. Journal of Research in Marketing* 13, 2 (April 1996), 181-197.

[15] International Organization for Standardization. 2011. ISO/IEC 25010:2011. *Systems and software engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*.

[16] Lamersdorf, A. and Münch, J. 2010. Studying the Impact of Global Software Development Characteristics on Project Goals: A Causal Model. *The Open Software Engineering Journal* 4, 1 (2010), 2-13.

[17] Madeyski, L. 2007. The Impact of Pair Programming and Test-Driven Development on Package Dependencies in Object-oriented Design – An Experiment. In *Proceedings of PROFES'07*. Springer, Berlin, Germany, 278-289.

[18] McDowell, C., Werner, L., Bullock, H.E., and Fernald, J. 2003. The impact of pair programming on student performance, perception and persistence. In *Proceedings of ICSE '03*. IEEE Computer Society, Washington, DC, USA, 602-607.

[19] Rajlich, V. 2006. Changing the paradigm of software engineering. Commun. ACM 49, 8 (August 2006), 67-70.

[20] Rus, I. and Lindvall, M. 2002. Knowledge Management in Software Engineering. *IEEE Software* 19, 3 (May 2002), 26-38.

[21] Russo, F. 2011. Explaining causal modelling. Or, what a causal model ought to explain. *New Essays in Logic and Philosophy of Science*.

[22] Sison, R. and Yang, T. Use of Agile Methods and Practices in the Philippines. In *Proceedings of APSEC'07*. IEEE Computer Society, Washington, DC, USA, 462-469.

[23] Trendowicz, A. 2013. *Software Cost Estimation, Benchmarking, and Risk Assessment*. Springer, Germany.

[24] Trendowicz, A. and Münch, J. 2009. Factors Influencing Software Development Productivity - State of the Art and Industrial Experiences. *Advances in Software Engineering* 77 (2009), 185-241.

[1] International Workshop on Impact of Agile Practices (ImpAct)