An Experience Report on ERP Effort Estimation Driven by Quality Requirements

Pierre Erasmus^{1,2}, Maya Daneva²

¹ SAP, the Netherlands, pierre.erasmus@sap.com
² University of Twente, the Netherlands m.daneva@utwente.nl

Abstract. Producing useful and accurate project effort estimates is highly dependable on the proper definition of the project scope. In the ERP service industry, the scope of an ERP service project is determined by desired needs which are driven by certain quality attributes that the client expects to be present in the implemented service solution. However, most of the needs for quality are implicit at the bidding stage of an ERP service project. Using software architecture principles such as more structured approaches to determining the prioritized set of quality requirements and their traceability to functional requirements, could help improve project effort estimation accurateness, project preparation and scope validation. This paper presents a real-world example in the SAP organization, in which we demonstrate this.

Keywords: Enterprise resource planning (ERP), requirements-based effort estimation, Expert Judgment, quality requirements, software architecture, functional size measurement, empirical report.

1 Introduction

Scope and effort estimation happening under extreme time pressure is more often than not the reality of many Enterprise Resource Planning (ERP) project teams who are responding to a request for proposal, in the pre-contract stage of an ERP initiative. Due to changes and pressures in the ERP service industry (which now offers a technical landscape known as SAP HANA [1]), most ERP projects today are based on a landscape which allows for a high degree of customizable solutions. This context calls for revisions in the ways we estimate effort for ERP service project scope. The ERP industry's expectations are also becoming more demanding to deliver estimates with more precision, more efficiently, in shorter intervals (as it's the case in the prebidding and bidding stages), with less resources for a more flexible, customizable and complex landscape. At the same time, the discipline of software architecture in general has demonstrated how we can effectively deal with higher degree of uncertainty within highly complex landscapes with a higher degree of accuracy and efficiency [18]. The software architecture discipline recommends the prioritization of requirements to identify requirements and components that complement each other but also identify components that is in conflict with one another. Specifically,

Copyright © 2015 by the authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.

software architecture methods such as the Architecture Tradeoff Analysis Method (ATAM) [2] and those proposed in [18], start by identifying and prioritizing quality requirements (QR). In many ERP projects however, designing software architecture starts with the prioritization of functional requirements often without considering the qualities that the customer and end-users consider as most important. There is a dominating attitude to over-rely on the sound principles and the robust architecture concepts on which modern ERP packages are built. However, just like a company relies on corporate strategies to manage and drive a successful business, modern software architecture uses QRs (Quality Requirements) to drive the architecture [2]. Using the same criteria's "prioritized qualities" that both a customer and an "end user" value should logically be the driving force behind an architecture. This makes practical sense in providing the ERP customers with what they asked for. Also, this provide the project with the best chance to fulfill the desired qualities of the customer's organization, which will be the result in the project succeeding or not. It's too often that ERP projects deliver very good functionality but was not required or different from what was asked for also known as scope creep. Many ERP customized software projects fail even before the project was started due to incorrect scoping driven by misunderstanding of the required expected qualities [3]. In worse cases the functionality delivered could even be in conflict with the desired and expected qualities [4], in this case the authors suggest using what they call the Quality Function Deployment [19] to select functionality that will deliver a specific quality. As the customer and end user mostly communicate the success and failures of software projects in term of how well the software match the expected qualities communicated throughout the project it make sense to validate against these quality criteria's. Therefore good software architecture goes hand in hand with the scoping of a software projects and starts as early as prioritizing the QRs during effort estimation.

This paper makes the case for including QRs in the early effort and scope estimation of ERP projects, and in particular ERP Service projects. Currently, there is a lack of knowledge in both expert judgment based estimation and functional size measurement (FSM) based estimation on how effort estimation technique could take the QRs in a project as the input into the estimation process. Acknowledging the absence of any empirical work on how to include QRs in software/ERP effort estimation process, we felt motivated to initiate some empirical research in this area. This article aims at starting a conversation on both academic and industry-related questions on how to use expert judgment based methods driven by QRs, in order to improve the accuracy of effort and scope estimates. Furthermore, this article also contributes by providing insight of how to use FSM and Expert Judgment methods together. There is some but limited research available on this topic such as from authors like [16] showing how FSM methods can be combined with expert judgment in a hybrid approach with using linear regression.

We make the note that this work is part of a bigger research project [5] on the development of an effort estimation model for the ERP service industry. In our previous publication [5], we presented the first design of an ERP Service project estimation method that uses expert judgments collected from multiple experts contributing to the project estimation process. The method is designed to take as input the functional requirements (FRs) known in the early stage of an ERP Service project. Drawing upon the preliminary evaluation of the method, in this paper we extend it by

accounting also for the QRs in a project. We provide an example of a real-world case in which we demonstrate the key ideas of our proposal on how to include QRs in effort estimation.

2 Background and Related Work

In ERP projects, system problems are often related to QRs, rather than to functional requirements (FRs). When customers provide feedback on newlyimplemented projects, it is more often than not in terms of quality measures, such as poor quality, unreliable, inefficient, unsecure [8]. There is therefore an agreement in the ERP community that project estimation should be grounded on both FRs and QRs. Currently, in most software effort estimation models, software effort is usually a function of software size and some other attributes such as project complexity, reusability, team experience and project type, among others [10]. Software size is estimated based on FRs [20]. QRs' contribution to the effort estimate is however often implicit. Some of the attributes mentioned earlier in this paragraph (e.g. project complexity, reusability) do represent the QRs of the software. However, a wide range of QRs is not mentioned at all in any established effort estimation model [7]. Examples of software effort prediction models that produce an estimate based on software size only are those published in [12,13,14]. In the software measurement literature, there are different opinions on the role of QRs in effort estimation and the ways in which the estimation should be done. While empirical studies tend to agree that QRs generally tend to increase the total project effort (e.g. by 30% as indicated in [10], or by 50% as found in [9]), there is no agreement regarding how to estimate the size of QRs [11]. As mentioned earlier, in this study we already accept that QRs could have an impact on ERP projects. We draw upon the previous results of several studies [7] that have shown the importance of QRs as a critical and important contributor towards delivering accurate effort estimates. However, because little is published on how to use QRs as part of software estimation in practice and almost no literature available for using QRs as part of ERP services estimations, our focus in this paper is on how to incorporate the use of QRs as part of ERP Services effort estimations. Based on a real-life example of an ERP project in a major ERP vendor's organization, we attempted to come up with some practical guidance for the industry on how to use these requirements, and when to use them as part of ERP services effort estimation. In our particular example, ERP project effort estimation will use QRs to decide on the solution architecture considering the configuration and customization of multiple ERP Service solution offerings delivered on a dedicated and standardized technology foundation (such as Netweaver or SAP HANA [1]). Even though the ERP Services offerings are rather standardized, they could be configured and modified to serve a specific need expressed in terms of a quality requirement. Often the architecture is planned around the leading set of prioritized quality needs. Such an example could be that a customer organization puts its emphasis on *maintainability*, which requires different configuration and customization effort than, e.g., architecture planned on *performance* as the driver or the leading quality requirement. Therefore, one can assume that the engineering of the QRs for ERP projects effort estimation should be one of the first processes after conducting the requirements elicitation.

3 The ERP Service Estimation Method

This section provides background on our method [5] known as the ERP Services Estimation Method (SEM) that uses FRs as input to the estimation process. As indicated in the Introduction, the method has been is designed specifically for facilitating effort estimations of ERP Services. The ERP-SEM method includes tactics and lessons learned (addressed in the previous section) and designed to decrease participation or memory bias while improving estimation accurateness. ERP-SEM includes four steps: (1) customer requirements, (2) scope formulation, (2) estimation and (4) validation, as shown in Figure 1. Depending on the project's complexity or risk, the cycles could be repeated iteratively to increase the level of detail in the expert knowledge needed to derive project estimates.



Figure 1: ERP effort estimation method

1. Customer requirement elicitation: This involves the elicitation and documentation of customer requirements. It resembles the evaluation phase of the ASAP method often used in the industry for SAP projects. More in detail, as ERP Service projects are often associated as part of a change management initiative, therefore engineering the customer requirements also includes the investigation (analysis) and documentation of business processes or refinement of these business processes (re-modeling) to present the suggested process changes.

2. Scope formulation: This is concerned with scoping of ERP Service projects. It resembles the business blue print phase used in the industry for SAP projects. More in detail, this step takes as input the requirements from the previous step, and then results in prioritized requirements. The architect can often determine the best fit (high level design) and best choice of technologies based on previous experience. The architecture (now representing groups of functional requirements) is advanced for

further refinement (as per enquiry) to specific solution experts representing specific group of functionalities. Important to good expert judgments estimation in this step is to incorporate different stakeholders during the scoping process and to decompose estimates where uncertainty is high.

3. Estimation: The effort estimation approach is initially top down followed by a bottom up approach. The estimation is initially rolled down with a top down approach which starts with the architecture and design of the project are defined (or refined in the case of a consecutive iteration). The level of abstraction is determined on the level of uncertainty or risk associated to the project. A project with a high degree of uncertainty or risk is often required to carry out estimation on a more thorough level of abstraction. The bottom-up approach is often carried out in this case when a detailed estimation is required of the effort (time) to implement specific solution objects or functions. Linking common casualty as a common practice by creating rules of thumb is often created for reusable estimation purposes but does not fit every estimation situation. This is considered as good practice as long as there is feedback provided to validate these rules. Inaccurate configurable rules may persist because experts get little or no feedback. FSM methods should also not be used for all situations and should or could only be used when there is an advantage to do so. Therefore the ERP Effort Estimation method could be supported by the use of estimation strategies to indicate when to use these methods in which situation.

4. Validation: The validation phase represents activities associated with the validation of the estimate. Among the techniques include creating a process whereby the effort estimates are validated. The validation process itself can make use of expert judgment, actual recorded values, functional size measurement points and group based validation.

4 Projects Driven by Quality Requirements

This section shows how ERP-SEM could be extended to include QRs. We provide details on our design for steps 1,2 and 3 in Figure 1. Step 4 is outside the scope of the paper.

4.1. Customer requirements elicitation

The SAP organization recognizes the role of QRs as the driver for the ERP software architecture. Therefore, in this step we assume ATAM [2] be used as part of the QRs elicitation activity. This assumption is justified as ATAM is a popular approach, broadly used in projects happening in multiple countries and in a variety of business sectors. ATAM would help the project estimation team identify the architectural drivers, namely those quality attributes that "shape" the architecture. A utility tree as defined in [2], is usually created (similar to a Work Breakdown Structure) with an initial top down vehicle for characterizing the "driving" attribute or requirements which is the most important or valued quality goals to be the high-level nodes [2,17]. If the project landscape in question is deemed complex (e.g. if multiple

SAP service offerings are to be integrated), it is realistic to expect more than one desired quality. This means, multiple QRs would be deemed important and, in turn, prioritization of QRs would be instrumental to determine potential conflicting design decisions during architecture, design and implementation phases. Knowing the priorities upfront has a significant impact on scope and project estimation.



Figure 2: Prioritized list of QR

For the purpose of using ERP-SEM, we suggest the use of a prioritized list of QRs (Figure 2) derived from the customer requirements. This prioritized list will then drive the suggested architecture and design. The impact that a suggested architecture will have on an estimate, would in turn determine the estimation scope, especially during bottom-up estimation practices.

4.2. Scope formulation

This step is grounded on the practice of decomposing QRs definitions into operationalized specifications of FRs. This choice is motivated because in ERP Service projects more often than not QRs are used to define the initial scope of an estimate while FRs are used to derive the functional components estimated to fit the scope. Clearly, there should be a traceability link between each QR and its underlying FRs as shown in Figure 3.



Figure 3: Tracking Relationship between QRs and FRs

As Figure 3 indicates, a QR is decomposed in two or more FRs; there might be several QRs with several associated functional requirements. It might well be possible that FRs uniquely trace back to only one QR. Or alternatively, it might well be possible that one FR is shared among several QRs. We note that the relationships among the QRs and FRs are determined during the estimation scope formulation step described in the ERP-SEM method [5].

For more accuracy, a FR is usually broken down into detailed subcomponent requirements as shown below (Figure 4).



Figure 4: FR subcomponents

In cases where the QRs are not known, or rather there is no process in place to help determine the driving qualities of the project, it is possible for the scope to get misjudged, which could eventually result in effort under-estimation. To avoid this situation, the project organization should determine the requirements which are missing and those which are not desired or "not needed". It is well possible that the lack of clarity on desired qualities or the inclusion of undesired qualities might produce an estimate different from what is needed in reality. This would render all further work done towards effort and scope estimation less useful, even if the effort estimation includes the precise formulation of FRs. Previous observations report an average of 50% error due to scoping related mistakes made early on in the estimation process whereby not all desired qualities were included in the initial estimate [5].

Lack of proper traceability of QRs with its underlying supporting FRs often leads to high level estimations which exclude the complexity associated to certain detailed customization scenarios as indicated in Figure 5 below. The estimates might be considered to be vague without a proper understanding of what needs to be done in terms of execution.

Quality Requirement				
	Functional Requirement1			
	Functional Requirement1.1			
	Functional Requirementn			
	→ Functional Requirement1.1			

Figure 5: Lack of functional complexity

4.3. The Effort Estimation Process

To introduce this step, we necessarily have to provide some background on a standard and well-accepted vehicle to estimate project scope of ERP projects, namely the Work Breakdown Structure (WBS) template. It is usually provided by ERP vendors. In case of SAP, it has often been reused from one project to another and is broadly accepted within the industry. WBS templates are made available by vendors for each solution in their ERP offerings and account for core process implementation and reusability of effort estimation. The WBS is used to organize work packages as activities to a more granular (executable) level, therefore can provide an estimate with more precision. When using a predefined WBS caution should be taken to only make reuse of the parts related to the project scope derived from the QRs. In case of SAP, the WBS template cannot be easily related to QRs. This is because the WBS has been initially designed to tell implementation teams "what can be done" in terms of functional implementation as part of a bigger list of tasks associated to a specific solution. The WBS template cannot indicate "what to do" to achieve a certain quality. Therefore estimators can turn to the FRs that are already known, in order to determine which activities in the WBS to use; at this point, estimators can also identify which other activities to include on a more granular level. The standard WBS template is not meant to be used as-is but rather represent the baseline for estimators to decide which tasks (implementation of functionality) are available as standard content. These tasks only represent the baseline and not customized scenarios. The WBS template is helpful to assist during estimation scoping by showing the relationship between functional activities and showing the sub activities often reflecting a deeper level insight of a higher level task as shown below (Figure 6).



Figure 6: Requirements derived from a SAP WBS

In cases where there is a lack of knowledge or experience is missing to determine the traceability links between QRs and the respective FRs, the estimators could involve experts to complete the missing information by using their expert judgments. Usually, such experts use their knowledge of the FRs embedded in the ERP solutions and can identify which are those that are needed to achieve a specific QR. If functional size measurements are available from past implementation projects, these are used as the baseline to back up the expert judgments that formed the input into the estimation step. However, it's important to mention that functional size measurements are often unavailable and therefore expert judgments are often used alone. To sum up, in the estimation step of the ERP-SEM method, the SAP WBS template is the vehicle during top-down and high-level estimation whereby technology architects are considered to be the best experts for deriving the scope and initial architecture. Expert judgments are used to fill the gaps regarding he traceability links between QRs and FRs, and acquire further detailed insight for a certain task and solution, whereby solution consultants are considered to be the best experts during bottom-up and detailed effort estimation (Figure 7).



Figure 7: Top down and Bottom up effort estimation

Further reuse of estimates of past projects can be enabled by recording the estimates derived via expert judgment as rules of thumb for often repeated customization scenarios associated to a certain solution and WBS. Therefore rules of thumb can be used in the absence of expert knowledge on a specific scenarios or can even be reused and benchmark against for future estimates as shown in Figure 8.



Figure 8: Record Expert Judgment

5 Reflection on the Early Use of the Method and Discussion

The ERP SEM approach described in Section 4 was carried out in 6 SAP projects by the first author. Three of the projects were in the category of the so-called rapid deployment solutions (RDS), where there were a detailed WBSs and well-engineered services in scope. The other projects were categorized as innovation projects, where new customer requirements and functionality was implemented. In the case of the RDS projects, we noted that even though project documentation was available such as predefined WBS, it was not enough to determine the scope and critical desired "value added" benefits to the customers or projects. Using the ERP-SEM method, two-three high priority QRs were selected and these then drove the estimation team to scope and focus the project on the real value added scenarios aligned and driven by the most desired qualities. In one example, the high priority requirements were interoperability and integration, which were deemed most critical to the ERP client's CIO in order to leverage several SAP systems (known as trade promotion optimization, demand signal management and advanced planner and optimizer). In this case, if the effort estimating specialist would have only adopted the traditional estimation process that accounts for the WBS of a single solution and its functional requirements, the would have never been able to determine the desired needs and qualities that are valued the most, even if it is only focusing on the engineered services.

In the other three projects (the innovation ones), the scope was rather unclear or functionality not yet developed. There was no project documentation or predefined WBS. In one case of a customer in the consumer goods industry, the project was triggered by the marketing department which was interested in analytics based on consuming big data and calculate meaningful market share trends. In this case, the scenarios for creating analytics and for creating market trend analytics represented well known functionality by consultants (experts), but there was no clarity on the underlying technology infrastructure that will be the foundation for implementing the functionality. Only by starting with the desired QRs – performance and mobility, would have been possible to indicate which technology and FRs to focus on, which ultimately supported the estimation and bidding process.

In both groups of projects, using a prioritized list of "value added" high priority QRs with existing and underlying new and existing knowledge (namely, the WBS's) documentation proved to deliver more accurate and timely estimates.

6 Project Pilot

We were interested to know how the ERP-SEM method that is QRs-driven and that is presented in this paper, compares with the method that uses FRs only and that is introduced in [5]. To answer this question, we used the QRs-driven ERP-SEM method in a pilot project. The estimation results are measured in terms of estimation accuracy. The pilot project staff was divided in two groups and independently of each other was given exactly the same ERP component and tasks to provide estimates for. The participants did not know each other and were given the tasks at the same time. They had two days to provide their estimates in line with the current workload and

actually considered the estimation task as a true customer estimate. We choose to keep the pilot as close as possible to a real industry experience in a normal day to day task. Twenty SAP participants were involved, ten solution consultants and ten technology architects. The architects used the QRs-driven approach, while the solution consultants followed the ERP-SEM approach as described in [5]. Tables 1 and 2 presents the results of our pilot.

Participant Number	Total tasks identified	Total effort in man-hours
P1	15	125
P2	14	132
P3	24	232
P4	17	156
P5	26	213
P6	36	289
P7	25	212
P8	16	129
P9	25	202
P10	20	166
Mean	22	186
Standard Dev	7	53
Median	22	184
Actual		308
Error		-40

Table 1: Responses of technology architects using the QRs-driven approach.

In each table, the first column indicates the participant number, while the second column reports the number of tasks correctly identified by each participant. The third column reports the total effort that each participant arrived at after using available project documentation, his/her own knowledge of the ERP product and the client's organization, and his/her experience as a SAP technology architect or a consultant.

Participant Number	Total tasks identified	Total effort in man-hours
P11	12	176
P12	8	105
P13	8	87
P14	15	254
P15	12	167
P16	16	154
P17	8	117
P18	9	135
P19	16	210
P20	17	139
Mean	12	154
Standard Dev	4	50
Median	12	147
Actual		308
Error		-50

Table 2: Responses of solution consultants using the original approach [5].

We found that an average of 17 sub tasks was identified for this specific scenario in our pilot project. The amount of sub-tasks identified as part of the estimation scope represented by a specific scenario directly affects the total effort. If for example an expert overlooks or forgets a certain sub-task associated to a certain scenario, then it will lead to an inaccurate estimate.

The experiment also indicates that the SAP technology architects (Table 1) provided more accurate scope of the estimations, while using the ERP-SEM method driven by QRs. The result showed an average of 40% error while the SAP solution consultants (Table 2) achieved an average of 50% error, only considering FRs as inputs into the estimation process.

We could see a larger deviation if looking at the median instead of the mean. For the QRs-driven approach, the median was 184 hours vs 174 hours for the FRs-driven ERP-SEM. Another finding was that solely looking at the average amount of subtasks detected using the QRs-driven approach on average found 22 subtasks vs an average of 12 subtasks found through the FRs-driven approach. In our 2013 study [5], we found that an architect is less accurate in determining estimates than solution consultants per individual sub-task. In contrast to this finding, the present paper suggests that technology architects are better to determine scope based on ORs. This is a reason to believe that if we form heterogeneous teams of estimators -i.e. mixed teams of architects and solution consultants, it might well be that the difference in estimation accuracy between QRs-driven approach and the FRs-driven approach might be even bigger. This of course could make the argument in favor of the QRsapproach even stronger. Based on the first authors experience in the six projects and the findings in the pilot, it seems that a QRs-driven estimation approach could deliver more accurate estimates, especially if estimators use the advantages provided by the two different types of experts (solution consultant and technology architect).

7 Conclusions

This paper presented a proposal of how to include prioritized QRs into the process of estimating scope for ERP service projects. The first author used the proposal in six projects; however we deem this to be a very early evaluation of the applicability of our approach. In these six projects the approach turned out useful as it provided a structured process of reasoning about the estimates and refocusing the participants in the estimation process on the QRs. This alone encouraged us to use a pilot project in which to compare our proposal and the one from our previously published work [5]. In this follow-up pilot, we had twenty SAP participants divided into two groups, each using one version of the ERP-SEM method. One group (composed of technology architects) used it while accounting for QRs, and the other group (composed of solution consultants) used the FRs only, as the input to the estimation process. We found that the SAP solution consultants had on average 50% error, while the SAP technology architects – on average 40% error. We could therefore conclude that using a QRs-driven ERP-SEM approach could provide on average 10% more accurate estimations, when compared to estimation that deploys an FRs-driven approach.

We consider the experiences of using our estimation method only indicative. We hope the estimation principles discussed in this paper apply to ICT projects in general and to the improvement of ERP services effort estimates, specifically. This is because we deployed the decomposition of QRs into functional requirements, which is a practice broadly used in studies of other researchers and practitioners attempting to estimate projects by acknowledging the role of QRs. However, as we are conscious about the validity threats [21] to our results, our highest priority is to carry out a number of case studies in a more systematic fashion, in order to collect evidence to confirm or disconfirm the experiences made so far. Without such follow-up studies, it is impossible to generalize our results to other settings beyond the example projects and the pilot. Other lines of future work could include more emphasis on how the architecture principles and methods can be applied in the industry together with solution consultants and the lessons learned during identifying the tradeoffs between making use of different types of experts in combination using the same methods as suggested in this paper.

References

- 1. H. Plattner, A. Zeier, In-Memory Data Management An Inflection Point for Enterprise Applications, Heidelburg: Springer, 2011.
- 2. R. Kazman, Architecture Tradeoff Analysis Method, SEI, 1998.
- E. J. Umble, Enterprise resource planning: Implementation procedures, Elsevier, no. 146, p. 241–257, 2003.
- 4. B. Nuseibeh, Requirements Engineering: A Roadmap, in Proceedings of the Conference on The Future of Software Engineering, New York, 2000.
- I. P. Erasmus, M. Daneva, ERP Effort Estimation Based on Expert Judgments, IWSM-Mensura, IEEE CS Press, 2013.
- L. Chung, On Non-Functional Requirements in Software, Conceptual Modeling: Foundations and Applications, vol. 5600, pp. 363-379, 2009.
- M. Kassab, M. Daneva, O. Ormandjieva, Towards an Early Software Effort Estimation Based on Functional and Non-Functional Requirements, IWSM- Mensura, pp. 182-196, 2009.
- L. Chung, J. do Prado Leite, On Non-Functional Requirements in Software Engineering, Conceptual Modeling: Foundations and Applications, vol. 5600, pp. 363-379, 2009.
- 9. G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, New York: John Wiley, 1998.
- D. D. Galorath and M. W. Evans, Software Sizing, Estimation, and Risk Management, Boston: Auerbach Publications, 2006.
- 11. M. Glinz, On non-functional requirements, RE, 2007.
- 12. Z. Jiang, P. Naudé and B. Jiang, The effects of software size on development effort and software quality, International Journal of Computer and Information Science and Engineering, vol. 1, pp. 230-234, 2007.
- 13. W. Xia, L. F. Capretz, D. Ho and F. Ahmed, A new calibration for Function Point complexity weights, Information and Software Technology, vol. 50, pp. 670-683, 2008.
- C. Lopez-Martin, C. Isaza and A. Chavoya, Software development effort prediction of industrial projects applying a general regression neural network, Empirical Software Engineering, vol. 17, pp. 1-19, 2011.

- 15. I. P. Erasmus, The COSMIC EPC method An ERP functional size measurement method delivering time and cost estimates, Chalmers University, 2012.
- 16 D. R. Baker, A Hybrid Approach to Expert and Model Based Effort Estimation, West Viginia: UMI, 2007.
- 17 P. Clements, R. Kazman and M. Klein, Evaluating Software Architectures: Methods and Case Studies, Addison-Wesley Professional, 2001.
- 18 Mistrik, I., Bahsoon, R., Kazman, R., Zhang, H., Economics-Driven Software Architecture, Morgan Kaufmann, 2014.
- 19. Herzwurm, G., Schockert, S., Pietsch, W., QFD for Customer-Focused Requirements Engineering. RE 2003: 330.
- Daneva, M. (2010) Balancing uncertainty of context in ERP project estimation: an approach and a case study. Journal of Software Maintenance and Evolution: Research and Practice, 22 (5). pp. 310-335
- 21. Wieringa, R., Design Science Methodology for Information Systems and Software Engineering, Springer 2014