

View Harmonization in Software Processes: from the Idea to QuASE

Vladimir A. Shekhovtsov¹ and Heinrich C. Mayr²

Abstract: Qualitätssicherung in Software Prozessen setzt voraus, dass die Beteiligten ein gleiches Verständnis der zu erfüllenden Anforderungen haben. Ein zentraler Aspekt des Quality-Aware Software Engineering (QuASE) ist daher die Harmonisierung von Stakeholder-Sichten bei der Systementwicklung. Das Papier zeigt auf, wie von der ersten Idee zur Unterstützung des wechselseitigen Verstehens von Qualitätsanforderungen das System QuASE entstand: dieses unterstützt nicht nur ein umfassendes „Understandability Management“, sondern integriert auch Wissen aus unterschiedlichen Ticketing-Systemen, welches damit zugänglich und wiederverwertbar gemacht wird.

Abstract: Quality assurance in software processes requires that the involved parties share a common understanding of the requirements to be addressed. As a result, a central aspect of Quality-Aware Software Engineering (QuASE) is to harmonize the stakeholder views in system development. The paper shows how our approach to QuASE evolved from the initial idea to a complete software system supporting mutual understanding of communicated information: this system not only provides a comprehensive set of understandability management techniques but also integrates knowledge from different ticketing systems.

Keywords: quality-aware software development, stakeholder view harmonization; domain-specific model; ontology; knowledge base.

1 Introduction and motivation

To ensure successful development, the participants in a software process have to be continuously aware of the quality level of the system under development regarding both: (1) the technical quality of the system as well as (2) the fulfilment of the requirements put on it. Whereas (1) can be ensured by relying on defined and proven software engineering methodologies, (2) traditionally suffers from impairments of the mutual understanding of the project stakeholders having different professional and educational background. Consequently, the quality expectations of the project stakeholders have to be collected, aligned before and during their transfer into the system and software requirements, which are expected to guide the architectural design and implementation activities. Moreover, during development continuous or at least periodical reviews and

¹ Alpen-Adria-Universität Klagenfurt, Institute for Applied Informatics, Universitätsstraße 65-67, A-9020 Klagenfurt am Wörthersee, Volodymyr.Shekhovtsov@aau.at

² Alpen-Adria-Universität Klagenfurt, Institute for Applied Informatics, Universitätsstraße 65-67, A-9020 Klagenfurt am Wörthersee, Heinrich.Mayr@aau.at

adjustments have to be performed in co-operation of all stakeholders concerned. We refer to such an approach as *quality-aware software engineering* (QuASE).

The challenge which initially motivated our QuASE research was the difficulty to involve non-IT stakeholders into the quality-aware software development process as they hardly can talk about the quality of the system without experiencing it. As a result, our first research question in a short version was: “*Is it possible to involve non-IT stakeholders into the process of eliciting quality requirements and if so, how?*” The QuASE project³ and the software solution to be implemented as a result of this project were initially intended to target that question by finding ways of making all involved parties in the software process speak the same language while discussing quality of a software under development (*harmonizing stakeholder views on quality*).

The QuASE project was carried out in co-operation with four partner software companies, each of them being active in a different field of application. While we collected the knowledge about stakeholder communications from the representatives of the partner companies, and analyzed their completed projects, we came to the conclusion that the original scope of our project had to be extended: from targeting only quality-related communication to harmonizing communicated stakeholder views on arbitrary aspects of the software under development. The rationale for this extension was that, for small and medium-sized projects supported by issue management (ticketing) systems such as Atlassian Jira, the amount of quality-related communicated information is small as compared to other kinds of information. As a result of this scope extension, the QuASE solution was redesigned to implement flexible view harmonization support. This support is provided on the basis of a knowledge-oriented representation of communicated information collected in project repositories (the databases maintained usually using industrial issue management systems).

In this paper, we track down the evolution of ideas which served as the foundations for the QuASE research, together with the evolution of the QuASE solution itself. Also, we describe the current state of the project and its future prospects.

The paper is organized as follows. In Section 2, we describe the preliminary research on quality conceptualization in user-centered requirements engineering and show the connections between this research and ensuing QuASE ideas. Section 3 outlines the initial problem statement for the QuASE project. Section 4 discusses the further evolution of QuASE ideas by (1) making it model-based with the specifics of the particular deployment site defined by means of a metamodel-based modeling language and (2) extending the scope of the project to cover arbitrary communicated information. The current state of the QuASE project is sketched in Section 5. It This section also shows how the generic QuASE solution can be customized to solve the original problem

³ The QuASE project was an FFG Bridge 1 project conducted from 01.03.2013 until 28.02.2015 in cooperation with four Austrian IT small and medium-sized IT companies

of quality view harmonization, it is followed with a description of the related work, the possible directions for the future research and conclusions.

2 Preliminary research

The origins of the QuASE project can be tracked down to research activities that aimed at conceptualizing and managing quality-related information in the context of user-centered requirements engineering. The goal of introducing quality in this context was to facilitate the involvement of non-IT stakeholders into the activities of discussing and negotiating the quality of a system under development.

The following research question has been formulated at that stage: "*How to support and thus better involve non-IT stakeholders into the requirements elicitation and analysis process so that they can address the expected system quality before experiencing the completed system*". To address this question, we analyzed and adapted a predesign approach (i.e., a technique for user-centric requirements modeling [MK98]) to exploit the following research ideas:

1. using aspect orientation to incorporate crosscutting concepts (which can be used to represent quality-related issues such as security or performance) into user centered requirements models for supporting separation of concerns in a problem domain; this research led to establishing the Aspectual Predesign Model [SK05];
2. exemplifying the aspectual predesign model to support modeling qualities of the prospective system; this research led to establishing the Quality-Aware Predesign Model [Ka08, SKM08] which allows for an explicit separation of functional and quality concerns;
3. further specializing the established quality-aware model to represent qualities of software services and/or business processes [Sh11];

The techniques developed at this stage suffered from the following shortcomings:

1. support of the conceptualization of quality-related information only; no guidelines for the integration of these conceptualizations into the software process involving business stakeholders, no means supporting such involvement;
2. no tool support provided, only few practical experience.

To overcome these limitations, we drove our research into the problem area of quality view harmonization and elaborated the initial problem statement for the QuASE project.

3 Quality view harmonization

In investigating the research area of harmonizing stakeholder views on quality, we started from the idea of a model-based involvement of business stakeholders into the software development activities. These initial activities concentrated on the conceptual foundations of the user's process of evaluating the qualities of a system under development; initially the scope was limited to service-oriented systems, but soon this restriction [SMK12a, SM12] was lifted.

Prior to making further conceptualizations and, eventually, proposing the conceptualization of the entire view harmonization process, we collected knowledge from representatives of our partner companies by means of quantitative techniques (questionnaires, structured and free-form interviews etc.) [SMK12b, SMK14].

Based on this collected knowledge we elaborated both initial [SMK13] and extended [SMK14] conceptualizations of the process of harmonizing stakeholder views on quality. We defined a harmonization process as the entirety of interactions between involved process sides targeting at a set of terminology means, quality views, and qualities that satisfies all these process sides [SMK14]. We addressed the following dimensions of quality view harmonization: (1) the quality of the system under development and the related aspects as well as their interrelations, (2) the knowledge about stakeholders, the differences in their perception of quality, and their amount of knowledge about the software quality, (3) the knowledge about the activities of the harmonization process and the capabilities of the process participants needed for performing these activities.

In [SMK14], we also distinguished three levels of defining the harmonization process: (1) *terminology harmonization*, where the stakeholders seek an agreement on a commonly understandable quality-related terminology, i.e., the language constructs used for expressing expectations and opinions on quality; (2) *view harmonization*, where the stakeholders seek an agreement on the sets of objects and the types of their qualities they are interested in, as well as the procedures of assessment, and (3) *quality harmonization*, where the stakeholders seek an agreement on the evaluation schemes and the particular qualities they are interested in.

The conceptualizations served as a basis for developing the initial QuASE proposal [SM13]. We built this proposal around two main ideas which are described below.

The first basic idea was to make the resulting QuASE system implemented as an ontology-driven solution where the ontology incorporates the knowledge about the quality harmonization domain, and serves as a basis for establishing the harmonization process. This ontology was planned to be established based on the domain knowledge collected from the company representatives by means of a standard ontology development approach like presented in [FGJ97]. The ontology was intended to be a sole source of harmonization, to be extendable by including other specialized ontologies such as measurement ontology [BVG06] etc. We initially did not plan to make the ontology

specific for a particular QuASE deployment configuration, in particular, to a specific target IT company. In addition, we did not expect to provide any means allowing the company representatives to modify this ontological knowledge beyond standard ontology development tools such as Protégé⁴.

In [SMK14], for standardization reasons we proposed to base the QuASE ontology on the Unified Foundational Ontology (UFO) [Gu05]. Consequently, the set of QuASE concepts forms a special-purpose ontology, which is derived from the UFO concepts.

The second basic idea was to not restrict our endeavor to the direct terminology harmonization support, e.g., implemented by means of mutually “translating” the communicated information's natural-language fragments stored in the project repositories. Rather, we decided to investigate the extension of our approach by means for exploiting quality-related communication experience represented in the repository in order to support decision making and predicting the behavior of the involved parties based on past quality-related communications.

This led to the objectives of the initial QuASE proposal as follows [SM13]:

1. Acquiring and formalizing domain knowledge about handling quality-related issues in the software process;
2. Collecting the raw information about such issues from the involved parties and converting it into operational knowledge (using available domain knowledge to ensure conversion correctness);
3. Using the collected knowledge for establishing a quality-related communication basis for the involved parties, supporting decision making, reuse of quality-related experience, and the prediction of the future quality-related requirements of the involved parties.

4 Ensuring flexibility and a proper scope

During the first year of the QuASE project another qualitative research was performed in cooperation with the representatives of the partner companies. By this set of interviews we pursued the following goals:

1. To reveal the evidence of the current solution's suitability as outlined in the proposal for the industrial settings, targeting mainly small and medium-sized enterprises; a special focus was put on the opinions about the suitability of the proposed ontology-based view harmonization approach;
2. To collect the opinions and experience about the exact scope of communicated information which the company representatives consider suitable for QuASE.

⁴ <http://protege.stanford.edu>

As a result of these interviews, the purpose-built ontology proved to be too limited for our purposes:

1. The knowledge about quality-related communications that can be collected from the company representatives varies from company to company; this knowledge, while being similarly structured, is different enough to justify being separately configurable for the particular deployment site;
2. The communicated information collected by the companies is stored, as a rule, in their project repositories (e.g. Jira databases); to benefit from the QuASE approach, such information has to be converted into knowledge, which is difficult to implement based on a special-purpose ontology as this ontology was not supposed to contain the instructions for such conversion;
3. The company representatives felt very uncomfortable with the idea of the direct modification of the ontology by knowledge suppliers using low-level ontology editing tools like Protégé.

These limitations led us to the elaboration of an extended set of QuASE use cases to be supported by the software solution [SML14], and to changing the ontology and knowledge base building approach:

1. The ontology building approach was changed in favor of generating the ontology based on a special-purpose model which is developed using a tool-supported metamodel-based visual domain-specific modeling language (*QuASE site DSL* [SMK15a]); this language is supposed to be suitable for use by knowledge suppliers and system configurators;
2. The knowledge base approach has been changed to a model-based one, too: the model was made to contain the mapping between the project repository and the modeling concepts which is then used to generate the individual knowledge base instances complying to the (generated) ontology concepts from the repository data [SM14, SMK15a].

In addition to the ontological matters, the second-stage qualitative research activities also revealed the challenges related to the scope of the information to be dealt with in the QuASE system.

The interviews favored the opinion that it is not productive to limit the scope of the system to harmonizing quality-related communications while targeting IT small-to-medium size enterprises. This is related to the fact that, in the projects typical for such companies, the subset of the quality-related communicated information stored in the project repository is limited as compared to the amount of the information related to functionality and other, non-quality-related factors. The representatives of the companies emphasized that the stakeholders not only refuse to discuss the quality of the system without experiencing it, but often prefer to delegate all quality-related communications to the IT people motivating that decision by lack of immediate interest to such issues and by trusting the developer's judgments at that stage.

To overcome this limitation, the QuASE solution was redesigned to implement flexible view harmonization support based on a knowledge-oriented representation of the communicated information collected in project repositories. The proposed extension was targeting the whole volume of communicated information with the possibility to filter it by topic – with the topics including quality of the prospective system and other aspects of software development.

The base set of the QuASE concepts underlying the QuASE site DSL [SMK15a] and the generated QuASE site ontology were modified to be generic enough to cover different categories of communicated information, the final set of concepts is depicted on Fig.1.

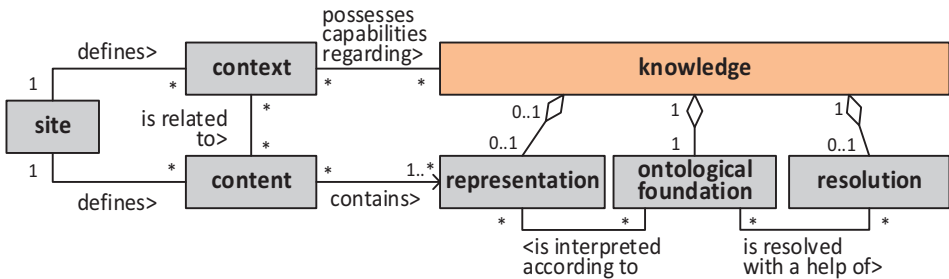


Fig.1. Generic QuASE concepts covering arbitrary communicated information [SMK15a]

This set includes the following concepts [SMK15a]: (1) *site*: owner of the given QuASE installation, e.g. a software provider; (2) *context*: units having particular views on communicated information e.g. projects, organizations, involved stakeholders; (3) *content*: units shaping communicated information e.g. issues/tickets; (4) *knowledge*: units encapsulating communicated knowledge that is a subject of harmonization.

Every QuASE knowledge unit is composed of: (1) *ontological foundation*: a reference to the conceptualization of the particular piece of knowledge through ontological means; (2) *representation*: the representation of the knowledge unit in a format that could be perceived by the communicating parties (e.g. plain text); representation units are also contained in content units; (3) *resolution means*: the means of resolving understandability conflicts related to the particular knowledge unit (e.g. explanations).

Context units possess *capabilities* to deal with knowledge units. In particular, the capabilities could refer to the ability of understanding a given knowledge unit at hand or explaining a knowledge unit using resolution means.

5 Current state of the QuASE research

Every QuASE knowledge unit is composed of: (1) *ontological foundation*: a reference to the conceptualization of the particular piece of knowledge through ontological means;

(2) *representation*: the representation of the knowledge unit in a format that could be perceived by the communicating parties (e.g. plain text); representation units are also contained in content units; (3) *resolution means*: the means of resolving understandability conflicts related to the particular knowledge unit (e.g. explanations).

The QuASE project has been successfully completed in 2015: a comprehensive tool support for view harmonization activities was implemented, which is currently available in a release quality to be used in real pioneer industrial settings; the subsequent release will target on normal professional usage. In this section we outline the current state of the solution: its architecture, the supported scenarios and support techniques, and the ability of the QuASE solution to address original QuASE research goals related to quality view harmonization.

5.1 QuASE system architecture

The current high-level architecture of the QuASE system is depicted in Fig.2. It includes the following components [SMK15a]:

1. A *QuASE site modeling tool* which supports (1) QuASE site DSL for describing the site-specific communication environment by means of the QuASE generic concepts; (2) a means of defining the site configuration; (3) and a means of specifying a mapping between the model and the particular project repository (e.g. Jira database).
2. A *QuASE ontology builder utility* transforming the models expressed in the QuASE site DSL into OWL2 representations of the QuASE site ontology, and the control information for mapping project repositories.
3. A *QuASE knowledge base builder utility* converting the data from project repositories into individuals corresponding to the QuASE site ontology based on the mapping specified by means of the QuASE modeling tool.
4. The interactive web-based *QuASE tool* implementing (1) the end-user support scenarios; (2) the acquisition of the information not found in project repositories from knowledge suppliers (e.g. user-supplied attribute values or the decision information).
5. A *QuASE knowledge base* which is a triple store containing OWL2 representation of site ontology (TBox) and knowledge base individuals (ABox).
6. An *IMS QuASE integration module* which allows using QuASE functionality from the user interface of issue management system such as Atlassian Jira.

A detailed description of the implementation architecture of the QuASE software solution has been published in [Sh15, Sh16b]; it includes server and client components, the security subsystem and the knowledge base support by means of Jena TDB triple store. It is extendable on the server side through web service-based API and on the client

side as web application based on AngularJS framework. To implement Jira integration, it includes the QuASE Jira plugin.

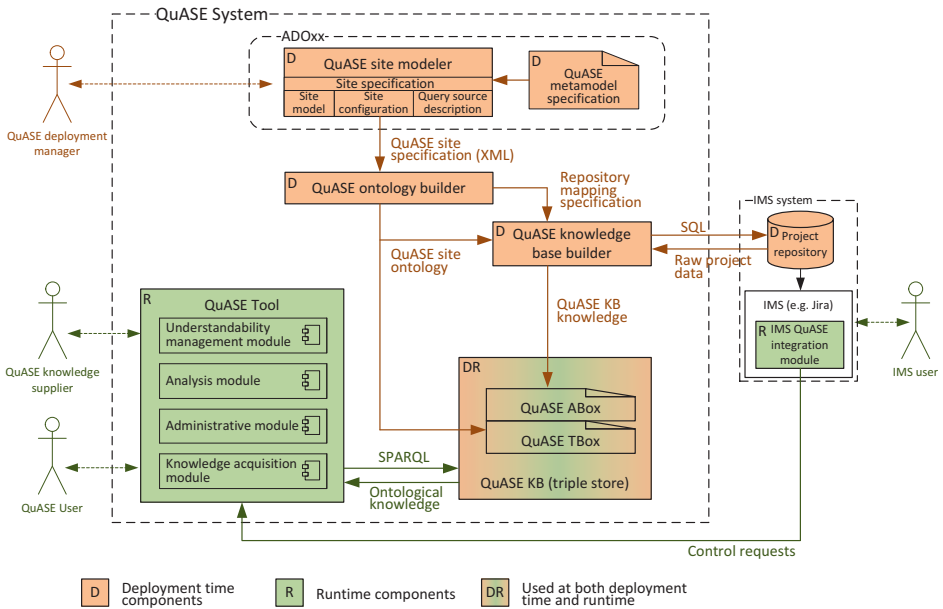


Fig.2. QuASE system architecture (extended version based on [SMK15a])

5.2 QuASE functionality

The complete set of QuASE tool usage scenarios is described in its user guide [Sh16a], it includes understandability management and analytical scenarios.

We established the scope of terminology harmonization support in QuASE in [SM14b] by providing the definition for *understandability management* activities. The implementation of the understandability management support in QuASE [Sh15] applies natural language processing techniques to localize the source of understandability conflicts in communicated information and to provide conflict resolution through terminology translation and issuing targeted explanations.

The current architecture and tool support for analytical activities in QuASE are described in [SMK15b]. The implementation of such support applies machine learning techniques to the set of QuASE knowledge base individuals: (1) similarity search for implementing information reuse; (2) regression analysis for predicting attribute values; (3) hybrid (partially supervised) learning for classification-based recommendation and decision support.

5.3 QuASE as a knowledge-based integrated solution

The advantage of the QuASE approach is that after extracting the data from a project repository, this data is immediately converted into knowledge available for understandability management and analysis. As the mapping is flexible by means of using a special-purpose DSL, this allows large amounts of existing data to be the subject of applying these techniques. The QuASE system can be seen as a bridge which connects end users, the data in project repositories and the (extendable) set of machine learning and natural language processing techniques. These techniques become applicable to the repository data automatically after the repository and communication environment are described by means of the QuASE site DSL.

5.4 Using QuASE for harmonizing quality views

While being a generic solution dealing with arbitrary communicated information, QuASE clearly is also able to address the original problem of harmonizing stakeholder views on quality. To allow this, it includes support for filtering the terminology used for understandability management by the communication domain; the QuASE knowledge base includes a set of concepts corresponding to such domains including the hierarchy of software quality characteristics. By filtering the ontological units based on one of such characteristics, the QuASE tool can be used to support quality view harmonization.

5.5 Tool deployment and usage status

As of June 2016, the QuASE tool has been installed at the sites of two partner companies, and it performed well for their repositories which included up to 26.000 issues. The resulting knowledge base contained up to 1.7 million axioms without significant performance problems. The main usage patterns for the tool are as follows:

1. The end users prefer accessing the system through the Jira integration interface which provides familiar means for arriving to the particular context or content unit;
2. The perceived usefulness of the understandability management capabilities of the tool depends on the quality of the terminological knowledge in the knowledge base, in particular, on the completeness of this knowledge with respect to the particular application domain (e.g. software quality).
3. The perceived usefulness of the analytical capabilities of the tool depends on the completeness of the set of metrics available for analysis. It is recommended to perform qualitative studies in cooperation with the company representatives to define the metrics influencing the development process to be included in the site model.

6 Related Work

A comprehensive review of the related work on quality view harmonization has been published in [SMK14], a review of the work similar in scope to the QuASE project – in [SMK15a].

7 Conclusions and the directions for future research

Based on the description of the evolution of the QuASE approach provided in this paper, we can distinguish the following major turning points in the evolution of the underlying ideas:

1. Advancing from quality conceptualization to the idea of harmonizing quality views;
2. Augmenting harmonization techniques with facilitating reuse of communicated information and supporting decisions based on this information;
3. Moving from harmonizing quality-related views to the generic idea of harmonizing communicated information;
4. Going from establishing a special-purpose ontology without end user support to a (meta-)model-based generic solution.

The QuASE approach is the result of a ten year's evolution of ideas on view harmonization, finally arriving to a comprehensive and flexible software solution which offers different techniques of harmonization support and makes them applicable to the communicated information of different nature stored in industrial project repositories.

In future, we see this approach applied to view harmonization problems occurring in other domains. Also we plan to enhance the understandability management support with more advanced terminology specification and translation techniques.

References

- [BVG06] Bertoa, M., Vallecillo, A., García, F.: An Ontology for Software Measurement. In *Ontologies for Software Engineering and Software Technology*, 175–196, Springer, 2006
- [FGJ97] Fernández, M., Gómez-Pérez, A., Juristo, N.: METHONTOLOGY: From Ontological Art towards Ontological Engineering. In *AAAI Spring Symp. Series*, 33–40, 1997
- [Gu05] Guizzardi, G.: *Ontological foundations for structural conceptual models*. University of Twente, 2005
- [Ka08] Kaschek, R., Kop, C., Shekhovtsov, V.A., Mayr, H.C.: *Towards Simulation-Based*

- Quality Requirements Elicitation: A Position Paper. In REFSQ 2008, Lecture Notes in Computer Science, Vol. 5025, 135-140, Springer, 2008
- [MK98] Mayr, H.C., Kop, C.: Conceptual Predesign - Bridging the Gap between Requirements and Conceptual Design. In ICRE '98, 90-100, IEEE CS Press, 1998
- [Sh11] Shekhovtsov, V.A., Kaschek, R., Kop, C., Mayr, H.C.: Relational service quality modeling. In Milanovic, N. (ed.): Non-Functional Properties in Service Oriented Architecture: Requirements, Models and Methods, 172-193, IGI Global, 2011
- [Sh15] Shekhovtsov, V.A., Mayr, H.C., Ianushkevych, S., Kucko, M., Lubenskyi, V., Strell, S.: Implementing tool support for effective stakeholder communication in software development – a project report. In Ausgewählte Beiträge zur Anwenderkonferenz für Softwarequalität Test und Innovation - ASQT 2014, books@ocg.at, Vol. 310, 45-58, Österreichische Computer Gesellschaft, 2015
- [Sh16a] Shekhovtsov, V.A. QuASE Tool User Guide v.1.3, http://quase-ainf.aau.at/documents/QuASE_User_Guide_1.3.pdf, accessed on 18.06.2016
- [Sh16b] Shekhovtsov, V.A. QuASE Architectural Description v.1.0, http://quase-ainf.aau.at/documents/QuASE_Architecture_1.0.pdf, accessed on 18.06.2016
- [SK05] Shekhovtsov, V.A., Kostanyan, A.: Aspectual predesign. In Information Systems Technology and its Applications - ISTA'2005, Lecture Notes in Informatics, Vol. P-63, 216-226, GI, 2005
- [SKM08] Shekhovtsov, V.A., Kop, C., Mayr, H.C.: Capturing the Semantics of Quality Requirements into an Intermediate Predesign Model. In SIGSAND-EUROPE'2008 Symposium, Lecture Notes in Informatics, Vol. P-129, 25-37, GI, 2008
- [SM12] Shekhovtsov, V.A., Mayr, H.C.: Let Stakeholders Define Quality: A Model-Based Approach. In Qualitätsmanagement und Vorgehensmodelle - 19. Workshop der GI-Fachgruppe Vorgehensmodelle, 101-110, Shaker Verlag GmbH, 2012
- [SM13] Shekhovtsov, V.A., Mayr, H.C.: Towards intelligent handling of quality related issues in software development – a project report. In Ausgewählte Beiträge zur Anwenderkonferenz für Softwarequalität Test und Innovation - ASQT'12, books@ocg.at, Vol. 295, 113-129, Österreichische Computer Gesellschaft, 2013
- [SM14a] Shekhovtsov, V.A., Mayr, H.C.: Managing Quality Related Information in Software Development Processes. In CAiSE-Forum-DC 2014, CEUR Workshop Proceedings, Vol. 1164, 73-80, CEUR-WS.org, 2014
- [SM14b] Shekhovtsov, V.A., Mayr, H.C.: Towards Managing Understandability of Quality-Related Information in Software Development Processes. In Murgante, B., Misra, S., Carlini, M., Torre, C., Nguyen, H.Q., Taniar, D., Apduhan, B., Gervasi, O. (eds.): ICCSA 2014, Proceedings, Part V., Lecture Notes in Computer Science, Vol. 8583, 572-585, Springer, 2014
- [SMK12a] Shekhovtsov, V.A., Mayr, H.C., Kop, C.: Stakeholder Involvement into Quality Definition and Evaluation for Service-Oriented Systems. In USER'12 Workshop at ICSE'12, 49-52, IEEE, 2012
- [SMK12b] Shekhovtsov, V.A., Mayr, H.C., Kop, C.: Acquiring Empirical Knowledge to Support Intelligent Analysis of Quality-Related Issues in Software Development. In Faria, J.P.,

Silva, A., Machado, R.J. (eds.): QUATIC 2012, 153-156, IEEE Press, 2012

- [SMK13] Shekhovtsov, V.A., Mayr, H.C., Kop, C.: Towards Conceptualizing Quality-Related Stakeholder Interactions in Software Development. In Mayr, H.C., Kop, C., Liddle, S., Ginige, A. (eds.): Information Systems: Methods, Models, and Applications, Lecture Notes in Business Information Processing, Vol. 137, 73-86, Springer, 2013
- [SMK14] Shekhovtsov, V.A., Mayr, H.C., Kop, C.: Harmonizing the Quality View of Stakeholders. In Mistrik, I., Bahsoon, R., Eeles, R., Roshandel, R., Stal, M. (eds.): Relating System Quality and Software Architecture, 41-73, Morgan-Kaufmann, 2014
- [SMK15a] Shekhovtsov, V.A., Mayr, H.C., Kop, C.: Facilitating Effective Stakeholder Communication in Software Development Processes. In Nurcan, S., Pimenidis, E. (eds.): Information Systems Engineering in Complex Environments, Lecture Notes in Business Information Processing, Vol. 204, 116-132, Springer International Publishing, 2015
- [SMK15b] Shekhovtsov, V.A., Mayr, H.C., Kucko, M.: Implementing tool support for analyzing stakeholder communications in software development. In Software Testing, Verification and Validation Workshops (ICSTW), 2015 IEEE Eighth International Conference on, 1-10, 2015
- [SML14] Shekhovtsov, V.A., Mayr, H.C., Lubenskyi, V.: QuASE: A Tool Supported Approach to Facilitating Quality-Related Communication in Software Development. In 9th International Conference on the Quality of Information and Communications Technology (QUATIC), 2014, 162-165, 2014