ICEIS 2015

17th International Conference on Enterprise Information Systems

PROCEEDINGS

Volume 2

Barcelona, Spain

27 - 30 April, 2015

www.iceis.org

ICEIS 2015

Proceedings of the
17th International Conference on
Enterprise Information Systems

Volume 2

Barcelona, Spain

27 - 30 April, 2015

Edited by Slimane Hammoudi, Leszek Maciaszek and Ernest Teniente

http://www.iceis.org

iceis.secretariat@insticc.org

BRIEF CONTENTS

INVITED SPEAKERS

George Giaglis

Athens University of Economics and Business

Greece

Witold Staniszkis

Rodan Development

Poland

Martin Mocker

MIT, USA and

Reutlingen University, Germany

SPECIAL SESSION CHAIRS

SPECIAL SESSION ON SECURITY IN INFORMATION SYSTEMS

David G. Rosado, University of Castilla-la Mancha, Spain

Carlos Blanco, University of Cantabria, Spain

Daniel Mellado, Spanish Tax Agency & GSyA Research Group, Spain

SPECIAL SESSION PROGRAM COMMITTEE

SPECIAL SESSION ON SECURITY IN INFORMATION SYSTEMS

Carlos Blanco, University of Cantabria, Spain

Kevin Butler, University of Oregon, U.S.A.

Pino Caballero-Gil, Universidad de La Laguna, Spain

Luis Enrique Sánchez Crespo, University of Castilla-La Mancha, Spain

Jaime Delgado, Universitat Politècnica de Catalunya, Spain

Csilla Farkas, University of South Carolina, U.S.A.

Maria Carmen Fernández, University of Málaga, Spain

Eduardo Fernández-medina, University of Castilla-La Mancha, Spain

Steven Furnell, Plymouth University, U.K.

Juan Manuel Carrillo de Gea, Software Engineering Research Group, Spain

Paolo Giorgini, University of Trento, Italy

Luigi Lo Iacono, Cologne University of Applied Sciences, Germany

Shareeful Islam, University of East London, U.K.

Hugo Jonker, University of Luxembourg, Luxembourg

Jan Jürjens, Fraunhofer ISST and TU Dortmund, Germany

Stamatis Karnouskos, SAP, Germany

Spyros Kokolakis, University of the Aegean, Greece

Antonio Maña, University of Malaga, Spain

Raimundas Matulevicius, University of Tartu, Estonia

Daniel Mellado, Spanish Tax Agency & GSyA Research Group, Spain

Haralambos Mouratidis, University of East London, U.K.

Federica Paci, University of Trento, Italy

Brajendra Panda, University of Arkansas, U.S.A.

Siani Pearson, HP Labs, Bristol, U.K.

Günther Pernul, University of Regensburg, Germany

Mario Piattini, Escuela Superior de Informatica, Spain

Indrakshi Ray, Colorado State University, U.S.A.

Alfonso Rodriguez, University of Bio-Bio, Chile

Luis Enrique Sánchez, Universidad de las Fuerzas Armadas (ESPE-L), Spain

Ambrosio Toval, University of Murcia, Spain

Juan Trujillo, University of Alicante, Spain

Sabrina de Capitani di Vimercati, Università degli Studi di Milano, Italy

Toshihiro Yamauchi, Okayama University, Japan

George Yee, Carleton University, Canada

SELECTED PAPERS BOOK

A number of selected papers presented at ICEIS 2015 will be published by Springer-Verlag in a LNBIP Series book. This selection will be done by the Conference Co-chairs and Program Co-chairs, among the papers actually presented at the conference, based on a rigorous review by the ICEIS 2015 Program Committee members.

FOREWORD

This book contains the proceedings of the 17th International Conference on Enterprise Information Systems (ICEIS 2015), which was sponsored by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC), held in cooperation with the Association for the Advancement of Artificial Intelligence (AAAI), IEICE Special Interest Group on Software Interprise Modelling (SWIM), ACM SIGMIS - ACM Special Interest Group on Management Information Systems, ACM SIGAI - ACM Special Interest Group on Artificial Intelligence and in collaboration with the Informatics Research Center (IRC). This year ICEIS was held in Barcelona, Spain from 27 - 30 April, 2015.

The purpose of the 17th International Conference on Enterprise Information Systems is to bring together researchers, engineers and practitioners from the areas of "Databases and Information Systems Integration", "Artificial Intelligence and Decision Support Systems", "Information Systems Analysis and Specification", "Software Agents and Internet Computing", "Human-Computer Interaction" and "Enterprise Architecture", interested in the advances and business applications of information systems.

ICEIS 2015 received 327 paper submissions from 52 countries in all continents, which represents a growth in relation to last year and makes ICEIS one of the largest conferences in the World in the area of Information Systems, thus demonstrating the success and global dimension of this conference. From these, 73 papers were published and presented as full papers (30min oral presentation), 83 papers reflecting work-in-progress were accepted for short presentation and another 40 papers were presented in a poster session. These numbers, leading to a full-paper acceptance ratio of 22%, show the intention of preserving a high quality forum for the next editions of this conference.

The high number and high quality of the received papers imposed difficult choices in the selection process. To evaluate each submission, a double blind paper review was performed by the Program Committee, whose members are highly qualified researchers in ICEIS topic areas.

All presented papers will be available at the SCITEPRESS Digital Library and will be submitted for indexation by Thomson Reuters Conference Proceedings Citation Index (ISI), INSPEC, DBLP, EI (Elsevier Index) and Scopus.

Additionally, a short list of presented papers will be selected to be expanded into a forthcoming book of ICEIS 2015 Selected Papers to be published by Springer in the LNBIP Series.

The technical program of the conference included a panel and 3 invited talks delivered by internationally distinguished speakers, namely: George Giaglis (Athens University of Economics and Business, Greece), Witold Staniszkis (Rodan Development, Poland) and Martin Mocker (MIT, USA and Reutlingen University, Germany). Their participation positively

contributes to reinforce the overall quality of the Conference and to provide a deeper understanding of the fields addressed by the conference.

Moreover, ICEIS 2015 had a special session on Security in Information Systems, a Doctoral Consortium on Enterprise Information Systems and 3 tutorials. We are thankful to the chairs for their dedication and hard work in organizing these events.

We sincerely thank all the authors for their submissions and participation in ICEIS 2015. Furthermore, we would like to thank all the members of the program committee and reviewers, who helped us with their expertise, dedication and time. We would also like to thank the invited speakers for their excellent contribution in sharing their knowledge and vision and the workshop/special session chairs whose collaboration with ICEIS 2015 was much appreciated. Finally, we gratefully acknowledge the professional support of the ICEIS 2015 team for all organizational processes.

We hope that all colleagues find this a fruitful and inspiring conference. We hope to contribute to the development of the Enterprise Information Systems community and look forward to having additional research results presented at the next ICEIS, to be held in Rome, Italy.

Slimane Hammoudi
ESEO, MODESTE, France

Leszek Maciaszek
Wroclaw University of Economics, Poland and Macquarie University, Sydney, Australia

Ernest Teniente
Polytechnic University of Catalonia, Spain

Contents

Using Technical-Action-Research to Validate a Framework for Authoring Software Engineering Methods

Miguel Morales-Trujillo[1], Hanna Oktaba[1] and Mario Piattini[2]

[1]*KUALI-KAANS Research Group, National Autonomous University of Mexico, Mexico City, Mexico*
[2]*Alarcos Research Group, University of Castilla – La Mancha, Paseo de la Universidad 4, 13071, Ciudad Real, Spain*
{migmor, hanna.oktaba}@ciencias.unam.mx, mario.piattini@uclm.es

Keywords: TAR, SEMAT, OMG, Case Studies, Evidence-based, Validation, Software Engineering.

Abstract: The validation of proposals has become a fundamental part of the creation of knowledge in Software Engineering. Initiatives like SEMAT have highlighted the need to base the correctness, usefulness and applicability of Software Engineering theories and practices on solid evidence. This paper presents the validation process used for KUALI-BEH, a proposal that became part of an OMG standard. The validation strategy applied was the result of integrating Technical-Action-Research and Case Study methods. After three years of work, we can conclude that TAR is a valuable research method, emphasizing that the main advantages of Technical-Action-Research are continuous feedback and the validation of an artifact, in this case KUALI-BEH, in a real context.

1 INTRODUCTION

Software Engineering is one of the most knowledge intensive disciplines (Edwards, 2003; Bjørnson and Dingsøyr, 2008). However, this particular field contains many proposals or theories that have no theoretical rigorousness and have not been adequately validated in practice.

According to (Johnson et al., 2012) most of these theories are not subject to serious academic discussion; they are not evaluated or compared as regards traditional criteria of theoretical quality such as consistency, correctness, comprehensiveness, and precision.

As (Jacobson et al., 2012) stated *"Software Engineering is afflicted by a lack of credible experimental evaluation and validation"*. The behavior of the discipline is not that which is desired, thus motivating the need to transform it and to build theories around it, to understand it, and more importantly, generate proven knowledge.

The above fostered the origin of the Software Engineering Method and Theory (SEMAT) initiative in 2009, in the form of a call for action to refound Software Engineering. Many prominent members of the Software Engineering community become signatories and supported the process that would be used to base Software Engineering on *"solid theory, proven principles and best practices"* (Jacobson et al., 2009).

Since then, the call for action has been transformed into a standardization process that was backed by the Object Management Group (OMG) through the *Foundation for the Agile Creation and Enactment of Software Engineering Methods Request for Proposals* (FACESEM RFP) (OMG, 2011), and which two years later crystallized into a standard: ESSENCE – Kernel and Language for Software Engineering Methods (OMG, 2013).

In the context of FACESEM, our research group has actively responded to the RFP by creating a proposal named KUALI-BEH (Morales-Trujillo et al., 2014b), which later became part of the ESSENCE specification after completing the standardization process, which is presented in detail in (Morales-Trujillo et al., 2014c).

In order to validate KUALI-BEH and evaluate its usefulness we have, over the last two years, developed a collaborative workshop and three case studies. The objective of this paper is to present the process applied to validate KUALI-BEH.

This paper is organized as follows: Section 2 presents a general background to the research methods in Software Engineering, focusing on those which are most relevant for the purpose of this paper. Section 3 describes the research strategy applied to create and validate KUALI-BEH. Sections 4 and 5 show the Collaborative Workshop and the Family of three Case Studies. The lessons learned are presented in Section

6, while the paper concludes in Section 7.

2 RESEARCH METHODS IN SOFTWARE ENGINEERING

Software Engineering requires both theoretical and empirical research. The former focuses on foundations and basic theories of software engineering, whilst the latter concentrates on fundamental principles, tools/environments, and best practices (Wang, 2007).

As mentioned by (Harrison et al., 1999), Software Engineering should have strong foundations as a scientific and engineering discipline. This implies that validation processes must be sufficiently mature to provide evidence that will support its advances. One alternative that can be used to achieve this goal is that of empirical software engineering since, according to (Belady and Lehman, 1976), this type of research provides the opportunity to build and verify its theories.

In Software Engineering, the validation must always involve scaling up to practice, which means that successive tests take place under increasingly realistic conditions (Wieringa, 2014). No matter what its form, the essence of an empirical study is the attempt to learn something useful by comparing theory to reality and to improve our theories as a result (Perry et al., 2000).

According to (Wang, 2007) and (Genero et al., 2014), the primary methods used for empirical studies in Software Engineering encompass: Experiments (Wohlin et al., 2012), Surveys (Kitchenham and Pfleeger, 2008), Case Studies (Yin, 2009; Runeson et al., 2012), Action-Research (Medeiros and Horta-Travassos, 2011), Systematic Literature Reviews (Kitchenham and Charters, 2007) and Standardization.

More than one method was considered for the validation of KUALI-BEH, and for the purposes of this paper, the most relevant research methods are therefore described in the following subsections.

2.1 Technical-Action-Research Method

Technical-Action-Research (TAR) is an approach that is used to validate new artifacts under conditions of practice (Wieringa and Moralı, 2012). In TAR the researcher uses an artifact in a real world project to help a client, or gives the artifact to others to use them in a real world project (Engelsman and Wieringa, 2012).

TAR can be seen as a research method that starts from the opposite side of traditional research methods. TAR starts with an artifact, and then tests it under conditions of practice by using it to solve concrete problems (Wieringa and Moralı, 2012).

In this method the researcher first develops the artifact, then tests it in a hypothetical situation in a laboratory and later scales it to be tested in real world situations, from an idealized context to the real one.

In technical disciplines, prototypes of artifacts are first tested in the idealized conditions of the laboratory, and these conditions are gradually relaxed after each iteration until a realistic version of the artifact is tested in a realistic environment, thus allowing the technical scientist to develop knowledge about the behavior of artifacts in practice (Wieringa and Moralı, 2012).

These iterations are called *engineering cycles* (Wieringa and Moralı, 2012) or *regulative cycles* (Van Strien, 1997). In the engineering cycle an improvement problem is investigated, a treatment with which to solve the problem is then designed and validated, its improved version is later implemented, and finally the experience with the implementation is evaluated and the cycle is executed again (Wieringa and Moralı, 2012).

Each engineering cycle is composed of four steps:

1. **Problem Investigation:** during this step the stakeholders, their goals and the improvement problem are identified.

2. **Treatment Design:** during this step the researcher designs the artifact that will be used as a treatment in the problem context.

3. **Design Validation:** after the artifact has been designed, the researcher states the expected effects, expected value, trade-offs and sensitivity.

4. **Treatment Implementation and Evaluation:** this step involves the insertion of the artifact into the real world. The artifact has to be validated and the researcher has to evaluate the resulting effects.

TAR is based on the assumption that what the researcher learns in this particular case will provide lessons learned that will be usable in the next case (Wieringa and Moralı, 2012). The expected result from TAR is the validation of the proposed artifact in the specified context, thus satisfying the stakeholders' needs and giving it value.

2.2 Case Study Research Method

A case study is an intensive investigation and analysis of a particular technology, project, organization, or environment based on information obtained from a variety of sources such as interviews, surveys, documents, test or trial results, and archival records (Wang,

2007). Wang establishes that case studies link a theory to practice, which allows conclusions to be drawn about the suitability of a given method for real world problems at industrial scales (Wang, 2007).

A case study is, according to (Yin, 2009) *an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident.*

The essence of a case study, the central tendency among all types of case study, is that it tries to illuminate a decision or set of decisions: why they were taken, how they were implemented, and with what result (Schramm, 1971).

Case studies may be used to validate a theory or method by means of empirical tests. They are also useful as regards providing a counter instance for a generally accepted principle. However, the drawback of case studies as an empirical method in Software Engineering is the difficulties of data collection and the generalization of findings via limited cases, particularly when they are positive but non-exhaustive (Wang, 2007).

According to (Runeson et al., 2012) the differences between Software Engineering study objects rely on the fact that: (i) they are entities that *develop* software rather than *using* it; (ii) they are *project-oriented* rather than *function oriented* organizations; and (iii) the work studied is *advanced* rather than *routine* engineering. The generic phases of a case study described by (Runeson et al., 2012) are:

1. **Case Study Design:** objectives are defined and the case study is planned.

2. **Preparation for Data Collection:** procedures and protocols for data collection are defined.

3. **Collecting Evidence:** data collection procedures are executed on the studied case.

4. **Analysis of Collected Data:** data analysis procedures are applied to the data.

5. **Reporting:** the study and its conclusions are packaged in feasible formats for reporting.

3 RESEARCH STRATEGY APPLIED

The research strategy used to validate KUALI-BEH was guided by the integration of TAR and Case Study methods. The TAR method was applied as a framework in which the evaluation during engineering cycles was carried out by applying the Case Study method.

The research process consisted of five engineering cycles (see Figure 1). The validation of these engineering cycles was carried out by means of case studies primarily developed in real world software development organizations. The artifact was also inserted in circles formed of practitioners and experts from academy and industry, such as a collaborative academy-industry workshop and the OMG Analysis and Design Task Force in charge of developing IT standards.

The first engineering cycle (from August 2011 through February 2012) consisted of identifying the common concepts used in the context of a software project and was carried out using a literature review.

During the second engineering cycle (from March through August 2012), an initial version of the artifact was released and validated by means of a collaborative workshop, whose participants were active Software Engineering practitioners, Master's degree students and researchers from the discipline.

The third engineering cycle was developed from September through December 2012. A first case study was carried out during these months in which an improved version of the artifact was validated in a Mexican enterprise in charge of software development and hardware construction. The artifact was also evaluated by the OMG task force.

During the fourth cycle (January through July 2013), improvements and lessons learned from the previous cycle were applied to the artifact. It was also used in a second case study, which took place in an entity that specializes in requirements specification and software projects design.

During the fifth and the last engineering cycle, which took place between August 2013 and April 2014, the artifact was enriched and an Authoring Extension was designed, which now belongs to the ESSENCE OMG formal specification. At that point the artifact was newly validated by the OMG task force, while a third case study was simultaneously carried out in a very small entity in charge of software development.

A detailed and in-depth description of the validation of the artifact is presented in Sections 4 and 5.

4 COLLABORATIVE WORKSHOP

KUALI-BEH was validated by means of a collaborative workshop attended by active practitioners from industry and academy. This section presents a detailed description of this workshop.
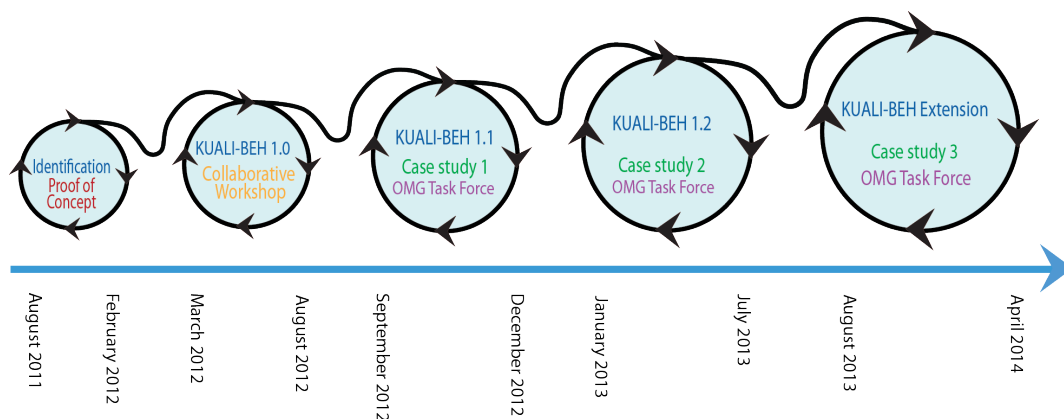
Figure 1: Engineering cycles of the applied research process.

4.1 Problem Investigation

The propositions of the study were focused on the pertinence, appropriateness and proficiency of the KUALI-BEH common concepts. These propositions were evaluated by asking the following questions:

- Are the common concepts pertinent?
- Are the definitions of the common concepts appropriate?
- Are the definitions of the common concepts similar to real world usage?
- Are the common concepts proficient?

These questions helped us to support the decisions made during the acceptance/rejection process carried out during the identification phase.

4.2 Artifact Design

The artifact to be validated by means of the workshop was the first version of KUALI-BEH, which was the output of the Identification phase of this research, and corresponded to its first engineering cycle. This version was base lined on February 20th, 2012.

4.3 Design and Implementation

The workshop was divided into eight on-site and online sessions. The theoretical components of KUALI-BEH were presented as follows:

- **Session 1:** Motivation, background and overview of the framework.
- **Session 2:** The Static view, the definitions of its 20 common concepts and its relationships and the graphical representation.
- **Session 3:** Method properties and the common concept templates.

- **Session 4:** The Operational view, the Method Enactment and the Practice Instance Lifecycle.
- **Session 5:** The adaptation operations.
- **Session 6:** The Operational boards.
- **Session 7:** Putting everything together.
- **Session 8:** Analysis of results and closure.

During the workshop, three approaches were used in order to obtain feedback from participants: application of surveys, direct interaction with participants during sessions, which included Q&A runs, and direct observation of activities. "Homework" activities also helped analyze the understanding of the proposal.

4.4 Evaluation

The data collected during the workshop resulted in a set of suggestions that were analyzed and classified by the researchers. After this process, the main improvements made to KUALI-BEH concerned the following issues: (i) The definitions of terms were enhanced; (ii) New definitions and operations were added; and (iii) Operational rules were improved.

4.5 Results

We developed a collaborative workshop to which organizations and active software engineers were invited. During the workshop we validated the first version of KUALI-BEH and we had the chance to try out its elements one by one.

The feedback received was extremely valuable input with which to improve KUALI-BEH and bridge the gap between theory and practice. We should also mention the participants' active contribution and involvement which helped establish a confident environment in which to share opinions and criticize KUALI-BEH.

This collaborative workshop was the closure to the second engineering cycle and the suggestions obtained served as a first step toward developing an improved version of KUALI-BEH.

5 FAMILY OF CASE STUDIES

In order to prove the usefulness and sufficiency of the framework and its common concepts, KUALI-BEH was validated using three case studies.

During each case study the practitioners defined their actual ways of working using KUALI-BEH, which is referred to as an authoring project. This section provides a detailed description of the case studies and their respective results.

5.1 General Considerations

The research question for the case studies was defined as follows:

Is KUALI-BEH suitable as regards defining practitioners' ways of working during software projects?

Two more questions also proved to be relevant for the research:

1. Is the effort of applying KUALI-BEH suitable when carrying out an authoring project?

2. What is the value obtained by the organization after having defined its own practices and method?

These questions were used as a basis to define the objectives (Os) that are common to the three case studies, as follows:

- **O1:** To demonstrate the sufficiency of the KUALI-BEH elements in describing practitioners' ways of working.

- **O2:** To measure the feasibility of using the concept of Practice to express practitioners' tacit practices.

- **O3:** To identify the value obtained by the organization as a consequence of defining its own method composed of its own practices.

In order to decide whether or not the objective had been achieved, the following indicators (Is) were collected

- **I1:** Associated with O1, which was collected applying two surveys to practitioners.

- **I2:** Associated with O2, which was obtained by measuring the effort required by practitioners to document their ways of working.

- **I3:** Associated with O3, which was obtained by means of an interview during the Feedback step, in which practitioners expressed the benefits and drawbacks of structuring their ways of working using KUALI-BEH.

Each of the case studies is described in more detail in the following subsections.

5.2 Case Study 1: InfoBLOCK

The first case study (CS1) was carried out at InfoBLOCK[1], a Mexican organization founded in 1997. InfoBLOCK's main activities involve hardware construction and software development.

Two of InfoBLOCK's general managers and two programmers participated in this case study. The programmers had spent between 1 and 3 years as active practitioners and reported having played the roles of Analyst, Tester, Project Manager and Programmer.

5.2.1 Background

Despite the soundness and constant growth of the organization, its managers were concerned about controlling the progress of projects in a more detailed way, but they did not have a defined development process. The needs expressed by the organization were consequently:

- To define the actual software development process followed in the organization.

- To be aware of what is being done by the work team at a particular moment during the development process.

The artifact to be validated using CS1 was the second version of KUALI-BEH, which was the output of the second engineering cycle, and had passed through the Collaborative Workshop presented in Section 4.

5.2.2 Design and Execution

The case study was designed as a sequence of the following seven steps:

1. **Presentation of the Case Study**

   The research team presented the objectives of the case study to the InfoBLOCK team, after which KUALI-BEH and its elements were explained. This step took 90 minutes.

2. **Description of Practitioners' Way of Working**

   This step was carried out in one work session, during which the researcher guided the practitioners

---
[1] http://www.infoblock.com.mx/

as regards the documentation of their first practice using the KUALI-BEH Practice template.

This also served to gain an insight into the organization and identify the individual goals and daily responsibilities of the practitioners involved in the case study.

After ensuring that the practitioners understood how to describe their ways of working using the concept of practice, the researcher assigned them activities that they should perform independently during the next step. This step took 120 minutes.

3. **Authoring of Practices**

The InfoBLOCK team began the authoring of practices and the running of an internal project simultaneously. Throughout this phase, the communication between the team and the researcher was carried out by videoconferencing and email.

The team completed the following tasks:

(a) The practitioners documented their way of working using the Practice template. This was done before they executed the practice in the internal project.

(b) The researcher then checked aspects of consistency between the KUALI-BEH concepts and what the team interpreted.

(c) The researcher later suggested improvements that could be made to the use of concepts but not the way of working itself.

(d) Finally, the practitioners discussed the suggestions and then generated the 1.0 version of the practice.

4. **Implementation of Documented Practices**

Owing to the fact that the case study was developed alongside a real project, the practitioners performed the practices almost immediately after documenting them. This resulted in adjustments being made to the initial version of the practices, which were carried out by following the same tasks developed in the previous step. During the three weeks of the project, 13 practices were identified, documented, implemented and adjusted by practitioners.

5. **Composition of a Method**

After the practices had been documented, executed and adapted, the practitioners proceeded to compose a method that represented their actual way of working using the 13 practices created.

The resulting method was then modified to achieve properties of coherency, consistency and sufficiency, thus attaining a well-formed method.

The practitioners used the method template and the graphical representation of KUALI-BEH to complete this step.

6. **Presentation of Results and Feedback Interview**

This step consisted of a session in which the documented practices and the composed method were presented to the organization's members.

During this meeting the researchers collected and registered orally-expressed lessons learned from this particular case study, such as suggestions for improvement, and the benefits and drawbacks of KUALI-BEH. The InfoBLOCK team also had the opportunity to think about their actual way of working and the improvements to be made in future projects.

7. **Closure**

At the phase of closure, the products eventually generated and the case study report were delivered to the InfoBLOCK team.

5.2.3 Analysis

At the end of this case study a method composed of 13 practices using KUALI-BEH was documented. According to the initial objectives of this case study the following results were obtained:

- **O1.** Based on the data collected by means of surveys and practitioners opinions, it was concluded that the elements of which KUALI-BEH consists were sufficient to describe the InfoBLOCK practitioners' way of working during the real project selected.

  Moreover, it was possible to observe that the common concepts were everyday concepts for practitioners, who used them in a natural and straightforward manner.

- **O2.** After considering the time and effort required by practitioners to understand KUALI-BEH, we were able to conclude that making practitioners' ways of working explicit in a short period of time was feasible. Except for the first practice, which required the researcher's support in documenting it, 12 more practices were expressed by the practitioners themselves without having to rethink the common concepts.

  The total effort required was 10 hours, resulting in an average of less than 50 minutes per practice. It was also observed that a practice contained an average of between 3 and 4 activities.

  Finally, documenting their way of working allowed the practitioners to identify work products

that were generated during the project. A total of 28 work products were identified and grouped into 9 categories.

- **O3.** The organization obtained the first version of its actual software development method, allowed its managers to share it with others, and also planned to use it as the means to train new employees. More importantly, the managers now have a means to manage progress and control the projects using the authored practices, thus making it easier to distribute and measure the work.

5.2.4 Feedback and Improvements

The most important improvement made to the KUALI-BEH proposal was the adjustment of the practice template in order to make it more legible, and these changes were focused on the visual organization of its elements.

5.3 Case Study 2: Entia

The second case study (CS2) was developed at Entia[2], a Mexican IT company that has been present in the industry since 2003 and has 20 employees. With the motivation to increase its projects' success rate, Entia developed ActiveAction, a game-centered intensive workshop which allows the organization to carry out the inception phase of its projects.

The general manager and a coach from Entia and two assistants from the research team took part in this case study.

5.3.1 Background

The game nature of ActiveAction and the constant on-the-fly adaptations had led to a rapid evolution of the workshop, one of whose consequences was the lack of documentation related to the workshop, while all the knowledge and techniques belonged to the people in charge and not to the organization.

The objective defined by the Entia general manager was therefore:

- To document each step of the workshop activities.

The artifact to be validated using CS2 was the third version of KUALI-BEH, the output of the third engineering cycle, which involved CS1.

5.3.2 Design and Execution

The generic design of the family of case studies steps was adapted according to Entia's context. This case study was therefore divided into the following six steps:

1. **On-site Observation of the Workshop**

   An on-site observation was conducted in order to understand the new game-based technique, its purpose and execution.

   It is important to mention that an ActiveAction workshop takes between 10 and 12 hours, signifying that this step took 720 minutes.

2. **Presentation of the Case Study**

   The objectives of the case study and KUALI-BEH were explained to the participants. This step took 60 minutes.

3. **Description of Practitioners' Way of Working**

   This step was identical to that in CS1, with the difference that the researcher, the general manager and the two assistants participated in it. This step took 60 minutes.

4. **Authoring and Adaptation of Practices**

   During this phase the assistants identified the practices involved in the workshop, completing the same tasks defined in Step 3 of CS1, with the only difference that the assistants presented the practices to the general manager, who suggested modifications, which were applied and then generated using the 1.0 version of the practice. During the three months of the project, 19 practices were expressed by the assistants and were agreed upon by the game expert.

   The challenge of this step was to identify the inputs and results of each practice, since Entia used to manage all of them as one work product: a mind map.

5. **Composition of a Method**

   Having documented the practices, the method composition step was a relatively easy task, because it resulted in the practices being ordered one after another, like a waterfall approach.

6. **Presentation of Results and Closure**

   This step consisted of a session in which the documented practices and the composed method were presented to the Entia general manager.

   The main result of this case study was the "unusual" method and its practices, which was presented as a research paper (Morales-Trujillo et al., 2014a) during the 9th International Conference on Evaluation of Novel Approaches to Software Engineering 2014 (ENASE14) which took place in Lisbon, Portugal.

---

[2]http://www.entia.com.mx/

5.3.3 Analysis

At the end of this case study a method composed of 19 practices was documented using KUALI-BEH. According to the initial objectives the following results were obtained:

- **O1.** The data collected by means of the surveys and the assistants' opinions allowed the researchers to conclude that the elements of which KUALI-BEH is composed were sufficient to describe the game-based method.

- **O2.** In this case study the required authoring effort was 26 hours, resulting in an average of 82 minutes per practice. It was concluded that expressing the workshop practices in a short period of time was feasible.

- **O3.** The only objective defined by Entia's general manager was that of documenting each step of the workshop activities, and this was fully achieved.

  Moreover, documenting the method reduced the possibility of variation and allowed Entia to identify ways in which to improve it and, more importantly, replicate it in affiliates.

5.3.4 Feedback and Improvements

On the one hand, this case study benefited Entia and allowed it to achieve important business goals in order to remain competitive. By publishing the paper at ENASE'14 Entia became more visible.

On the other hand, the KUALI-BEH proposal was improved and its usefulness was demonstrated in a context in which software development was not the main purpose.

5.4 Case Study 3: Tic-Tac

The third case study (CS3) was developed at San Luis Potosi Superior Tech Institute (ITSSLP) in a software development entity called Tic-Tac. Tic-Tac was part of the Tech business incubator program. Two professors and two recently graduated systems engineers participated in this case study.

5.4.1 Background

The software development entity was carrying out projects without having a defined method or process that could be followed. The objectives defined by the professors in charge of coordinating the entity were therefore:

- To document their software development process.

- To train new work team members using the process defined.

The artifact to be validated using CS3 was the KUALI-BEH Authoring Extension, which was base lined on November 12th, 2012.

5.4.2 Design and Execution

In order to satisfy the needs defined by the ITSSLP team, this case study added another objective to the three previously defined in section 5.1. The new objective was established as follows:

- **O4:** To measure the effort required to train a new work team member using the method defined.

The indicator that would demonstrate whether or not the objective had been achieved was the following:

- **I4:** Associated with O4, which was obtained by measuring the effort required by the practitioners to train a new work team member.

1. **Presentation of the Case Study**

   The objectives of the case study and KUALI-BEH were explained. This step took 60 minutes.

2. **Description of Practitioners' Way of Working**

   The researcher and the ITSSLP team documented its first practice, as had occurred in CS1 and CS2. This step took 60 minutes.

3. **Authoring of Practices**

   The ITSSLP team carried out this step in the same way as that defined in CS1. This step resulted in 23 documented practices.

4. **Composition of a Method**

   After the practices had been documented, executed and adapted, the Tic-Tac method was composed. It will be noted that, unlike the other case studies, the ITSSLP proposed that their method be divided into phases.

5. **Adaptation of the Method**

   In order improve and adjust the method, some adaptation operations were applied. The ITSSLP team adjusted its method using the concatenation and combination operations defined in KUALI-BEH.

6. **Presentation of Results and Feedback Interview**

   During this step, the documented practices and the composed method were presented to the ITSSLP authorities.

   The researchers used a videoconference to collect and register orally-expressed lessons learned from

this particular case study, such as suggestions for improvement, along with the benefits and drawbacks of KUALI-BEH.

7. **Closure**

The products generated and the report of the case study were delivered to the ITSSLP team.

5.4.3 Analysis

At the end of this case study a method that represented the ITSSLP software development entity's way of working was composed. The 23 authored practices and a list of required work products was also generated. According to the objectives of this case study, the following results were obtained:

- **O1.** The sufficiency of KUALI-BEH elements as regards describing the practitioners' way of working was confirmed. The results collected by the surveys demonstrated that it was possible and appropriate for practitioners to express their tacit practices using KUALI-BEH, and that its elements are understandable and useful.

- **O2.** In this case study the required effort was 44 hours, resulting in an average of 115 minutes per practice. The professors in charge of the entity expressed satisfaction with the effort made.

- **O3.** A software development method was documented and served as basis for the execution of the new software projects carried out by the entity. At that moment, two new projects were successfully developed and two new members were trained and integrated into the team.

- **O4.** The ITSSLP followed a seven-step strategy in order to incorporate two new members, and the time investment was 7 hours. According to the professors, the strategy's success was primarily related to the participants' knowledge and skills.

5.4.4 Feedback and Improvements

This case study experience allowed us to improve KUALI-BEH and its Authoring Extension, and it more specifically permitted us to better define the states and transitions of the Practice Authoring and Method Authoring Alphas.

We additionally confirmed that two of the adaptation operations are suitable and have real meaning for practitioners, although it is still necessary to verify their usefulness in other projects.

The results of the case study, expressed by the work team and professors during the feedback interview, were:

- *"We also think that KUALI-BEH guides the practitioner and causes his/her better participation in a project"*.

- *"In addition, it will serve to generate a repository of available methods and practices for different projects"*.

- *"Team members can think and rethink about how they do things"*.

5.5 Results

Conducting the case studies allowed us to learn many lessons, thus permitting us to improve KUALI-BEH and its validation process. The main lessons learned from these three case studies and the collaborative workshop concern the following issues:

A Tool is Required to Support the Practice-authoring Process. The use of the *Track changes* function of the word processor permitted the come-and-go of authored practices between practitioners and researchers to become a valuable learning process. However, managing the control version strategy of each practice, making adaptations and sharing practices became more difficult processes. At this point the need for a tool that would automate this process became essential

Support from Managers and Commitment from Practitioners is Necessary. The validation process is a process that involves many variables that must be controlled. On the one hand, it requires the organization's interest in the proposal and an understanding of how it will deal with its particular needs. On the other hand, the people in charge of applying the proposal must see benefits reflected in their daily work if bias and reluctance are to be avoided. A two-level support from managers and practitioners is therefore mandatory, and if either of these levels is not committed, the case study will not be possible.

Finding Suitable Projects in Order to Validate the Proposal. It is clear that support and enthusiasm are not the only ingredients needed to carry out a case study. The proposal needs to be validated in a real context and in a suitable project. While the research team is almost always available and open to the idea of conducting case studies, the organization needs to estimate the time, effort and risks of being involved beforehand, which may lead to time matching problems between the researcher and the organization's project.

Identifying Improvements and Adjustments to Artifact. The case studies have allowed us to improve many aspects of KUALI-BEH, thus making it a solid and accepted proposal that now belongs to an international standard.

We were also able to establish that the effort made (see Table 1) permitted a high ROI for both parties: practitioners and researchers

Table 1: Effort by step in each case study.

	CS1	CS2	CS3
On-site observation	-	720	-
Presentation of the CS	90	60	60
Way of working description	120	60	60
Authoring of practices		600	
Implementation of practices	390	-	2520
Composition of a method		120	
Training new members	-	-	420
Results and Feedback	60	60	60
Closure	30	30	30
Total effort (in hours)	**11.5**	**27.5**	**52.5**
Practices authored	13	19	23
Authoring effort (in hours)	**10**	**26**	**44**
Average effort per practice	46.2	82.1	114.8

5.6 Threats to Validity and Limitations

In order to avoid threats to validity, various factors were considered in the collaborative workshop and the three case studies.

- **Construct Validity:** Multiple data sources were used in order to provide evidence and respond to the research question. The sources used were interviews, direct observations, surveys and work artifacts, thus covering the three degrees defined by (Runeson et al., 2012).

  Moreover, the validity of the construct (or artifact) was ensured since it was created following the mandatory requirements requested by the OMG in the FACESEM RFP. It was also validated many times by the OMG Analysis and Design Task Force.

- **Internal Validity:** The case studies' results demonstrated that the objective for which KUALI-BEH was created was achieved, thus allowing us to accomplish our goals and the participants' needs.

  Different causal relations were examined:

  - The surveys' trustworthiness. The data collected with surveys is closely related to the practitioners' experience. In these case studies, the participants' experience covered the "juniors" through "seniors" classification.

  - The number of participants. Although the number of participants is not large, we can state that the sample is representative of the profile of practitioners working in small software organizations.

- The participants' age, education and experience. These factors could have affected whether they were in favor or against KUALI-BEH. However, the sample of participants was diverse.

After having analyzed these factors, all of them were disallowed, and as a result we have been able to determine that the implementation of KUALI-BEH in the case studies allowed us to achieve the case studies' objectives.

- **External Validity:** In spite of the limited number of organizations that participated in the case studies, we can state that each organization can be categorized as a typical software developer entity. The three participating organizations shared the main characteristics of very small entities in charge of software development endeavors, which is the target audience of the KUALI-BEH proposal.

  The selection of the participants was not intentional: the organizations that participated in the case studies expressed their interest in participating.

  It is important to highlight that the methods that were expressed using KUALI-BEH were the actual ways of working of active practitioners involved in real projects.

  We consequently believe that the sample and the context of this research were representative. What is more, thanks to the scopes of OMG and SEMAT, the results obtained were contrasted with researchers and practitioners from other countries who backed it up.

  We therefore conclude that the results obtained make it possible to generalize about the subject being researched.

- **Reliability:** The case studies were carried out by two researchers and the results were constantly triangulated to third parties, such as colleagues and members of the research group. The participants were additionally informed of the results and lessons learned, which were and extensively disseminated in each of the case study reports delivered to them.

  A detailed plan that served as guidance for the case studies is also available, thus making its replication possible.

In order to improve the validity of the case studies, it was compared with the checklists provided by (Runeson et al., 2012).

The following approaches were also taken into account:

- The *prolonged involvement*, mainly in CS2 and CS3 (3 and 6 months respectively), which allowed us to develop a trusting relationship with the participants, thus making the collection of data an unobstructed process and, in general, to develop the case studies in a pleasant atmosphere.

- During CS2 we had two assistants who participated in the data collection process, thus allowing us to analyze different data sources, such as interviews, surveys and direct observations. This circumstance made *triangulation* possible.

- *Peer debriefing* took place in all three case studies, owing to the fact that the case studies were carried out by two researchers. In addition, the findings and results were periodically discussed with the other members of the research group.

- The work products and documents generated during each case study were given to the participants so that they could review them. This took place at minimum after the finalization of each step of the study, but the member checking action was carried out at any moment if deemed to be necessary.

- A version control strategy was defined from the very beginning of each case study. Given the technological background of all the participants, the *audit trail* mechanism was easy to follow and its application was very successful.

Finally, the limitations of the case studies can be summarized in two points:

- The sample size is small, which therefore limits the power of generalization. It is necessary to replicate the case studies with bigger populations.

- Bias in the case studies could have occurred and been related to the participants' feeling of being observed and evaluated. This may have led to an alteration in their actual way of working.

6 LESSONS LEARNED

After three years of continuous work, we can conclude that the TAR is a valuable research method for the purposes of this research. Its application, in combination with case studies, allowed us to validate and improve KUALI-BEH after each engineering cycle.

The iterative approach of TAR allowed us to obtain continuous feedback about and validation of the artifact in a real context and these are, in our opinion, the main advantages of the TAR.

As researchers we had the opportunity to develop the three roles identified by Wieringa: Designer,

Helper and Researcher. Starting as designers, we created an artifact whose objective was to resolve a type of problems present in the industry. During the engineering cycles we inserted the artifact into organizations which were affected by this type of problem. In order to apply the proposed treatment, and so acting as helpers, we used the artifact and assessed its functioning.

As researchers we later took advantage of the lessons learned and analyzed the resulting effects, the value achieved and the trade-offs obtained with the objective of adjusting and improving the artifact.

Validating the artifact in a real context allowed us to gain experience and generate knowledge through the lessons learned which, according to (Endres and Rombach, 2003), are the principal means of obtaining knowledge in the Software Engineering discipline

TAR also provided the organizations involved in this research with benefits. At the end of the case studies we were able to establish that:

- Each organization had achieved the stated objectives.

- The practitioners had been trained in a new technology, in this case KUALI-BEH, at no "extra" cost.

- The case study results were used to achieve business goals, signifying that the artifact was useful and applicable to their particular contexts.

- Collaboration and partnership ties were promoted between the university and organizations.

Finally, we can define TAR as a research method which gives researchers a valuable opportunity to learn and obtain knowledge by applying theories in practice in order *to solve real problems by solving real problems*.

7 CONCLUSIONS

The research strategy presented in this paper is the result of the integration of TAR and Case Study methods. We took advantage of the engineering cycles, which were used as the means to create and validate the artifact in a real context, thus allowing us to minimize the disadvantages of a full empirical study which can, according to (Harrison et al., 1999), be very disruptive and time-consuming for the company involved

On the one hand, making the artifact available to critical reference groups (mainly OMG and SEMAT) formed by theoreticians and practitioners provided us with fruitful feedback and a wide range of points of view. That knowledge became lessons learned, which

were applied during the treatment design phase of each engineering cycle.

On the other hand, the application of case studies was meaningful as regards establishing the sensitivity of the artifact and whether or not it satisfied the stakeholders' expected value and the hypothesis of the present research in a real world scenario.

This research strategy allowed us to generate the KUALI-BEH framework, guided us through the validation process and helped us achieve the objective and goals set for this research.

Moreover, valuable lessons related to the OMG standardization process were learned and reported in (Morales-Trujillo et al., 2014c).

Finally, we can conclude that the combination of TAR and Case Study research methods was a successful experience, allowing us to validate and improve KUALI-BEH in several ways and making us realize that TAR is a powerful means to bridge the gap between academy and industry.

ACKNOWLEDGEMENTS

REFERENCES

Belady, L. and Lehman, M. (1976). A model of large program development. *IBM Syst. J.*, 15(3):225–252.

Bjørnson, F. O. and Dingsøyr, T. (2008). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Inform. Software Tech.*, 50(11):1055 – 1068.

Edwards, J. S. (2003). Managing software engineers and their knowledge. In *Managing software engineering knowledge*, pages 5–27. Springer.

Endres, A. and Rombach, D. (2003). *A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories*. Fraunhofer IESE series on software engineering. Pearson/Addison Wesley.

Engelsman, W. and Wieringa, R. (2012). Goal-oriented requirements engineering and enterprise architecture:

Two case studies and some lessons learned. In *Proceedings of the REFSQ'12*, volume 7195 of *LNCS*, pages 306–320. Springer-Verlag.

Genero, M., Cruz-Lemus, J., and Piattini, M. (2014). *Métodos de Investigación en Ingeniería del Software*. RA-MA Editorial.

Harrison, R., Badoo, N., Barry, E., Biffl, S., Parra, A., Winter, B., and Wst, J. (1999). Directions and methodologies for empirical software engineering research. *Empirical Software Engineering*, 4(4):405–410.

Jacobson, I., Meyer, B., and Soley, R. (2009). The SEMAT initiative: A call for action.

Jacobson, I., Ng, P.-W., McMahon, P., Spence, I., and Lidman, S. (2012). The essence of software engineering: The SEMAT kernel. *Queue*, 10(10):40–51.

Johnson, P., Ekstedt, M., and Jacobson, I. (2012). Where's the theory for software engineering? *IEEE Softw.*, 29(5):96.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, EBSE-2007-01.

Kitchenham, B. and Pfleeger, S. (2008). Personal opinion surveys. In Shull, F., Singer, J., and Sjberg, D., editors, *Guide to Advanced Empirical Software Engineering*, pages 63–92. Springer London.

Medeiros, P. and Horta-Travassos, G. (2011). Action research can swing the balance in experimental software engineering. *Advances in Computers*, pages 205–276.

Morales-Trujillo, M., Oktaba, H., and González, J. (2014a). Improving software projects inception phase using games: Activeaction workshop. In *Proceedings of the ENASE'14*, pages 180–187.

Morales-Trujillo, M., Oktaba, H., and Piattini, M. (2014b). Bottom-up authoring of software engineering methods and practices. *IET Software*. Submitted.

Morales-Trujillo, M., Oktaba, H., and Piattini, M. (2014c). The making-of an OMG standard. *Computer Standards & Interfaces*. Submitted.

OMG (2011). A foundation for the agile creation and enactment of software engineering methods RFP. Technical report, Object Management Group, Needham, USA.

OMG (2013). ESSENCE – kernel and language for software engineering methods. Technical report, Object Management Group, Needham, USA.

Perry, D., Porter, A., and Votta, L. (2000). Empirical studies of software engineering: A roadmap. In *Proceedings of the ICSE'00*, pages 345–355. ACM.

Runeson, P., Host, M., Rainer, A., and Regnell, B. (2012). *Case Study Research in Software Engineering: Guidelines and Examples*. Wiley.

Schramm, W. (1971). *Notes on Case Studies of Instructional Media Projects [microform] / Wilbur Schramm*. ERIC Clearinghouse.

Van Strien, P. (1997). Towards a methodology of psychological practice the regulative cycle. *Theory & Psychology*, 7(5):683–700.

Wang, Y. (2007). *Software Engineering Foundations: A Software Science Perspective*. Auerbach Publications, Boston, MA, USA, 1st edition.

Wieringa, R. (2014). Empirical research methods for technology validation: Scaling up to practice. *J. Syst. Softw.*, 95:19–31.

Wieringa, R. and Moralı, A. (2012). Technical action research as a validation method in information systems design science. In *Proceedings of the DESRIST'12*, pages 220–238. Springer-Verlag.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering: An Introduction*. Springer-Verlag.

Yin, R. (2009). *Case Study Research: Design and Methods*. Applied Social Research Methods. SAGE Publications, 4th edition.