

# From Collective Knowledge to Intelligence: Pre-Requirements Analysis of Large and Complex Systems

Peng Liang  
State Key Lab of Software Engineering  
Wuhan University, China  
liangp@sklse.org

Keqing He  
State Key Lab of Software Engineering  
Wuhan University, China  
hekeqing@sklse.org

Paris Avgeriou  
Department of Computing Science  
University of Groningen, The Netherlands  
paris@cs.rug.nl

Lai Xu  
Software Systems Research Centre  
Bournemouth University, UK  
laixu@acm.org

## ABSTRACT

Requirements engineering is essentially a social collaborative activity in which involved stakeholders have to closely work together to communicate, elicit, negotiate, define, confirm, and finally come up with the requirements for the system to be implemented or upgraded. In the development of large and complex systems, with a huge number of uncertain stakeholders, the requirements engineering process becomes a challenging task due to overwhelming and dynamic social interactions, tradeoffs, and collective decisions made by above stakeholders. Traditional approaches and techniques are deficient in supporting this kind of social interactions in requirements-related activities, and managing the evolving requirements and their traceability caused by the social interactions. This paper proposes to address the challenges in the pre-requirements analysis of large and complex systems by employing the techniques from collective intelligence based on Web 2.0 tools and technologies, which is composed of three steps: first, obtain collective requirements knowledge through collaborative tagging by stakeholders; second, transform collaborative requirement tags into requirement ontologies; third, support collective requirement decision-making (i.e., collective intelligence) based on the requirement ontologies through requirements reasoning.

## Categories and Subject Descriptors

D.2.1 [Software Engineering]: Requirements—*Methodologies*

## General Terms

Human Factors

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Web2SE'10, May 4, 2010, Cape Town, South Africa.

Copyright © 2010 ACM 978-1-60558-975-6/10/05...\$10.00.

## Keywords

Collective intelligence, pre-requirements analysis, collaborative tagging, requirements reasoning

## 1. INTRODUCTION

Computer-based systems are built for people and by people [7]. Requirements engineering (RE) of computer-based systems is essentially a social collaboration activity, in which involved stakeholders (e.g., customers and developers) have to closely work together to communicate, elicit, negotiate, define, confirm, and finally come up with the requirements (including functional and non-functional requirements) for the system to be implemented or upgraded [1].

Today software development is carried out by large distributed teams and concerned with a huge number of stakeholders [4]. The Ultra-Large-Scale (ULS) system, an extremely large-scale and highly complex software system, is a typical example [18]. The requirements of the ULS systems are inherently conflicting, unknowable, and diverse due to a wide variety of stakeholders with unavoidably different, conflicting, complex, and changing needs. For example, stakeholders may have different expectations of how the system is going to perform. The key challenge of the ULS system is that it's even impossible to gather the requirements [9]. As a metaphor, the construction of the ULS system is more like the evolution of the city, "*the form of a city is not defined in advance by specifying requirements; rather, a city emerges and changes over time through the loosely coordinated and regulated actions of many individuals*" [18].

In this context, some RE challenges have been identified in the development and maintenance of large-scale systems [13][21], including large number of customer requirements, changing technologies, distributed teams, creating and maintaining requirements traceability, and scoping change and creep. Traditional approaches to address these challenges focus on tool support (especially the requirements management tools) and RE process improvement guidelines and methods (e.g., WinWin approach for requirements negotiation). In this paper, we argue that people is actually the most important factor in the RE process. Their opinions, decisions [2], and interactions (e.g., collaboration and negotiation) are more critical to the final success of a project than how requirements are specified and managed. We propose to address these challenges in a social collaboration per-

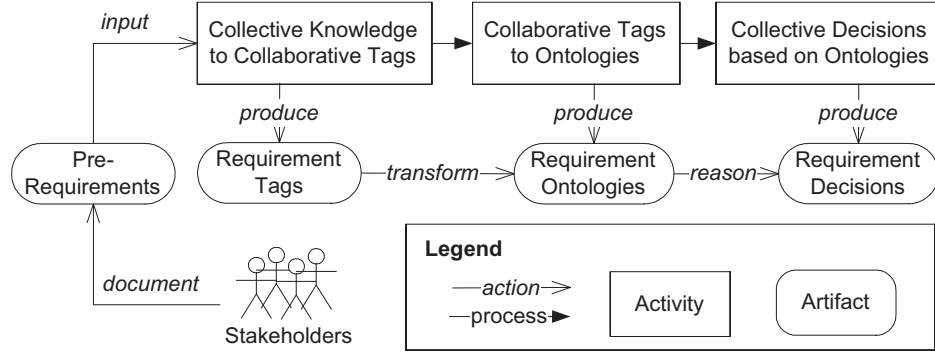


Figure 1: The process from collective requirements knowledge to intelligence.

spective. We present a method framework to support and improve the pre-requirements analysis of large and complex systems by employing the techniques from collective intelligence based on several popular Web 2.0 tools and technologies, including wikis, tags (folksonomies), and semantic web (ontologies and reasoning). The combination of these Web 2.0 tools and technologies is especially useful to refine the collective requirements knowledge obtained from massive and uncertain stakeholders into collective requirements intelligence (i.e., requirement decisions). The major benefit of the proposed method is that it lowers the barrier of active/effective stakeholders' participation, and facilitate collaborative decision-making process in pre-requirements analysis. This method framework can also deal with the continuous evolution and change of requirements during the whole requirements lifecycle in an uncertain world<sup>1</sup>.

The rest of this paper is organized as follows. The challenges of pre-requirements analysis in large and complex systems development are presented in Section 2. The proposed method framework in three steps is described in Section 3 with examples. The effectiveness of our method framework against the challenges is discussed in Section 4. Related work on collective intelligence and social requirements engineering is reviewed in Section 5. The paper concludes in Section 6.

## 2. CHALLENGES ADDRESSED

Pre-requirements analysis is about requirements' life prior to inclusion in the requirements specification (RS) [8]. In this turbulent phase, initial stakeholders<sup>2</sup> are not convinced about what the system should look like, and the requirement statements from diverse sources and stakeholders may eventually be integrated into a single requirement in the RS. Continuous iterations take place when changes in this phase need to be re-worked into the RS. This section details the challenges on pre-requirements analysis that are selected from [13][21], and that can be addressed by our proposed method framework:

**Requirements communication:** With markets globalization, more large and complex projects are running in a geographically distributed environment by distributed teams and stakeholders. Recent studies have shown that this dis-

tribution often leads to issues, such as coordination and communication challenges of requirements.

**Requirements negotiation** in large and complex projects with massive stakeholders is particularly difficult since various stakeholders always have different interests and perspectives about the system. The requirements negotiation task becomes even challenging when the number of stakeholders increases dramatically, and sometimes it is even impossible to come up with a requirement decision because of incomplete, contradictory, and changing requirement statements, which is termed as "wicked problems" [3]. Supporting method and tools should be provided in requirements negotiation activity in order to help stakeholders to find out compromised requirement options, and reach an agreement (i.e., requirement decision).

**Requirements traceability:** The pre-requirements traceability depends on the ability to trace requirements from, and back to, their originating statement(s) [8]. Creating and maintaining the pre-requirements traces for large and complex systems is a challenging task, due to the diverse sources and massive stakeholders, and subtle interrelationships that exist between requirements early on. Huge amount of time and manual effort are required for requirements traceability management in traditional approaches [8][20].

**Requirements change control:** In pre-requirements analysis, an early phase in RE process, stakeholders frequently change their mind and requirement statements. These changes may break the existing requirements balance (i.e., the current requirements set agreed and compromised by involved stakeholders). It is quite challenging to detect how and to what extent the requirement changes will affect the other requirements, including the propagation effect from the changed requirements, and to reach a new requirements balance.

We present a method framework to address these challenges in the next section, and discuss the effectiveness of the proposed method framework in Section 4.

## 3. FROM COLLECTIVE REQUIREMENTS KNOWLEDGE TO INTELLIGENCE

To address the challenges above, we propose to employ the techniques from collective intelligence based on popular Web 2.0 tools and technologies, including wikis, tags, and semantic web. Requirement statements issued by stakeholders are essentially the knowledge of these stakeholders, since

<sup>1</sup>But in this paper, we focus on the issues in pre-requirements analysis.

<sup>2</sup>The scope of stakeholders is also in a turbulent status.

they have the knowledge about the meaning of these requirement statements, and why they are beneficial for them. In traditional RE approaches, stakeholders communicate the requirements through requirements documentation, which works for a small and fixed number of stakeholders, but is insufficient when the scope of the stakeholders of a system are unpredictable (e.g., users of web-based systems). To effectively collect and manage the requirements knowledge issued by massive and uncertain stakeholders, we propose to use collaborative tagging method to tag the requirement statements by stakeholders themselves, and then these collaborative tags (collective knowledge) is transformed into requirement ontologies with formal semantics and richer expressivity. In the end, the reasoning techniques are employed to obtain requirement decisions (collective intelligence) over the formal requirement ontologies and reasoning rules (constraints).

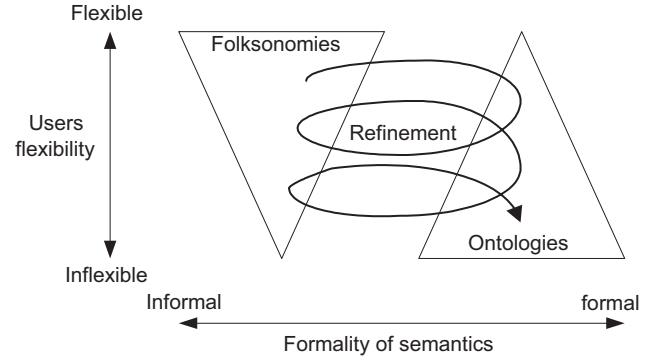
Figure 1 presents the process from collective requirements knowledge to collective intelligence (requirement decisions), including three activities (in rectangle) and four artifacts (in round rectangle) produced by activities and actions. These activities and artifacts will be further described in following subsections. Note that, wiki as a popular collaborative editing tool has already been introduced and used in distributed requirements documentation e.g., in [5], so the first step in Figure 1 (stakeholders document pre-requirements using wiki) is not covered in the following descriptions of activities.

### 3.1 Collective Knowledge to Collaborative Tags

In this activity, we ask stakeholders to tag their requirement statements using whatever tags they want to use based on their personal understanding. For example, stakeholders can tag a requirement statement “*The organizer shall be able to send invitation to candidate participants by email.*” with various tags e.g., *organizer*, *invitation*, *email*, *sending email*, *functional requirement*, and *high-priority*. In these tags, *organizer*, *invitation*, *email*, and *sending email* are tags that are directly concerned with the content of the requirement statement, and we call them **content-tag**; while *functional requirement* and *high-priority* are tags that are used to describe the properties of the requirement statement, and we name them as **meta-tag**. The **content-tag** and **meta-tag** are two kinds of tags that are most frequently used in tagging requirement statements by stakeholders. The benefit of collaborative tagging (i.e., folksonomies) is that it retains the individual understanding of various stakeholders, while the drawback is that folksonomies (results of collaborative tagging) suffer from inconsistency, ambiguity, and redundancy, for example, another stakeholder may tag this requirement statement with a tag *invitation letter*, which is redundant to previous tagging *invitation*. Collaborative tags (folksonomies) has been used in identifying and organizing concerns in pre-requirements analysis [19]. We make a further step in our research that these folksonomies should be refined and transformed into formal ontologies to support the pre-requirements analysis.

### 3.2 Collaborative Tags to Ontologies

Figure 2 shows an inverse two peaks model on how folksonomies can be transformed into ontologies through refinement. The number of tags in folksonomies is quite large initially. After the transformation (refinement) from folk-



**Figure 2: The inverse two peaks model from folksonomies to ontologies.**

sonomies into formal ontologies (e.g., the tags *invitation* and *invitation letter* are merged into an ontology *invitation letter*, which is a sub-ontology of the ontology *letter*), the number of tags decreases, while accordingly the number of ontologies increases along with the refinement process. The ontologies have more formal semantics (e.g., constraints on and relationships between ontologies), and the refined ontologies can be fed back into the collaborative tagging activity as candidate tags for folksonomies (the spiral line back to folksonomies in Figure 2). The feedbacks of ontologies into folksonomies can dramatically alleviate the problems of inconsistency, ambiguity, and redundancy of tags. Some mature work and results on building and maintaining ontologies from folksonomies have been reported in [15]. We can investigate and employ these techniques in the transformation/refinement from requirement tags into ontologies as well. Note that, the refinement process from collaborative tags to ontologies deals with different kinds of tags respectively (e.g., the **content-tag** and **meta-tag** as described in Section 3.1) since they target to the pre-requirements analysis at different levels. The **content-tags** focus on the content analysis of requirement statements, which are domain-specific (e.g., inconsistency of concerns), and the **meta-tags** are mainly used for the analysis of the properties and relationships of requirements, which are normally domain-independent (e.g., inconsistency of requirement properties). The grouping of tags (e.g., into **content-tag** and **meta-tag**) will be done with tool support (e.g., word analyzer) and human intervention.

### 3.3 Collective Decisions based on Ontologies

Ontology is defined as a formal and explicit specification of a shared conceptualization [10]. In the perspective of collective intelligence, ontology can also be regarded as refined knowledge that is distilled from collaborative tags. The purpose of constructing formal requirement ontologies is two-fold: to retrieve distilled requirements knowledge from massive stakeholders, and to support the group decision-making of requirements (collective intelligence) through requirements reasoning with formal ontologies. The requirement decisions include all kinds of decisions about requirements, e.g., decisions on requirement priorities, and conflicting requirements. For example, at the **content-tag** level, if the number of stakeholders' taggings by *invitation letter*, *sending* and *email* is greater than those by *invitation letter*,

sending and SMS (Short Message Service), then the requirement decision is that: the requirement “(The organizer shall be able to) send invitation letter by email” has a higher priority than the requirement “send invitation letter by SMS”. At the **meta-tag** level, if two tags *high-priority* and *low-priority* are both tagged with a requirement statement, then negotiation should be done to resolve the conflicting tags<sup>3</sup>. For tagging of non-functional requirements (NFRs), there are always NFR category tags like *security*, *performance*, and *scalability*, etc. being used at the **meta-tag** level to denote the categories that a requirement statement belongs to. A list of patterns<sup>4</sup> of potentially-conflicting NFR categories (e.g., *security* with *performance*) can be used to recommend potential conflict of concerns for stakeholders’ consideration and negotiation. Note that, the reasoning rules are defined by requirements engineers according to the best practices in pre-requirements analysis, and these rules can be readily described by semantic web rule language (SWRL) [12]. Some sample reasoning rules based on Requirements Rationale Model (a formal requirement ontology) has been presented in our previous work on requirements reasoning for distributed requirements analysis [14]. The requirement decisions (collective intelligence) are the final product of the proposed process. These decisions can automatically evolve based on the requirement traces generated during the process, and runtime requirements reasoning.

## 4. DISCUSSIONS

We discuss in this section how the proposed method framework can address the challenges presented in Section 2, while concrete validations should be performed to justify these arguments in our future work.

**Requirements communication:** The requirement tags generated by collaborative tagging of massive stakeholders are natural resources for requirements communication among stakeholders, which is quite similar to tagging web pages (e.g., del.icio.us) and pictures (e.g., flickr) for web content communication. The requirement ontologies distilled from collaborative requirement tags act as a formal communication tool (concepts, relationships, and constraints) for requirements, and promote the mutual understanding through a wider agreement of ontologies among stakeholders.

**Requirements negotiation:** First, requirements ontologies provide a common understanding for requirements negotiation. Second, the requirement decisions obtained by requirements reasoning over collective requirements knowledge (e.g., decisions on requirement priorities) provide supporting information or alternative solutions with sound rationale (why these decisions are made) for requirements negotiation.

**Requirements traceability:** Requirement traces from stakeholders to pre-requirements, and further to requirement decisions can be generated during the three steps of the proposed process, including the traces from pre-requirements to requirement tags, from requirement tags to requirement ontologies, and from requirement decisions to requirements ontologies and reasoning rules employed. All these traces

<sup>3</sup>Besides negotiation, the decision on requirement priority can also be made based on the number of taggings (e.g., if the number of taggings with *high-priority* is greater than that of *low-priority*, we can make decision that this is a requirement statement with *high-priority*).

<sup>4</sup>The pattern of potentially-conflicting NFR categories is a kind of reasoning rule for pre-requirements analysis.

are generated either as a side-product of an activity (e.g., requirements tagging) or automatically (e.g., from requirement decisions to requirement ontologies and reasoning rule employed) without much human effort.

**Requirements change control:** When stakeholders add or change their requirement statements, they are prompted to re-tag these new/updated statements, and those tags for tagging deprecated requirements are checked and removed if no requirement statement is tagged with these tags. The update of requirement tags results in the evolution of requirement ontologies, and further leads to the update on requirement decisions based on reasoning over requirements ontologies.

## 5. RELATED WORK

Collective intelligence and social requirements engineering are most related work with this paper. Collective intelligence is a shared or group intelligence that emerges from the collaboration and competition of many individuals. Gruber exploits the collective intelligence by combining the social web and semantic web, and proposes to unlock the “collective intelligence” of the social web with knowledge representation and reasoning techniques of the semantic web, and shows some concrete applications in [11]. Our method framework is heavily based on this idea that is applied in RE for large and complex systems.

To our knowledge, Goguen is the first author who posed the social issues in RE [7]. He mainly discussed the social issues on the organization of client and requirements team, and how these social issues negatively affected the RE activities and requirements quality with candidate solutions.

Ossher et al. implemented a prototype tool BITKit (part of IBM Requirements Composer tool suite) for tagging requirement concerns [19]. Their work primarily focuses on the first step in our proposed process, i.e., from pre-requirements in business analysis phase to requirements tags, in order to identify and organize stakeholders’ concerns.

Lohmann et al. implemented a web-based platform SoftWiki, which provides social RE features and support collaboration and knowledge sharing for larger groups of stakeholders in collection, discussion, development, and structuring of requirements [16]. The SoftWiki platform focuses on semantic annotation and sharing of requirements artifacts using a fixed ontology model - SWORE. The SWORE ontology is defined by RE experts, which is not flexible and cannot be easily extended. This makes SWORE insufficient for requirements tagging in an open and uncertain world.

Group decision support in RE with recommendation systems is also a kind of collective intelligence support in RE activities [17]. Felfernig et al. proposed and designed a system to support the decision-making process concerned with group stakeholders in RE on the basis of different types of recommendation algorithms [6]. These recommendation algorithms can be fairly integrated into our method framework, e.g., in Step 3 of requirements reasoning with specific constraints, in order to improve the results of requirement decisions or recommend better requirement alternatives.

## 6. CONCLUSIONS

In this paper, we present a method framework for the pre-requirements analysis of large and complex systems by employing collective intelligence based on Web 2.0 tools and

technologies. The major contributions are two points: (1) lower the barrier of stakeholders' participation and facilitate the collaborative decision-making process in pre-requirements analysis with Web 2.0, which is critical for large and complex systems development, e.g., ULS systems and software ecosystems; and (2) propose a roadmap (the process in three steps) and working agenda to address the challenges in pre-requirements analysis of large and complex systems. It's worth noting that the proposed method framework, from collective requirements knowledge to intelligence, is not a substitute to mature methods currently used in requirements analysis, but complementary to reduce human effort and promote social collaboration in pre-requirements analysis of large and complex systems.

This is currently a method framework without covering much technical details, for example, how to solve the inconsistency and ambiguity in collaborative tagging; what tagging language is used; how to transform tags (folksonomies) into ontologies, etc. The reason is that these technical issues have been fairly addressed in related research fields (e.g., social tagging systems), and these technical solutions will be further investigated/evaluated in the next step when implementing this method with validations of concrete cases.

## 7. ACKNOWLEDGMENTS

This research has been partially sponsored by the Natural Science Foundation of China (NSFC) under Grant No. 60950110352, STAND: Semantic-enabled collaboration Towards Analysis, Negotiation and Documentation on distributed requirements engineering, the National Basic Research Program of China (973) under Grant No. 2007CB310801, RE4ULS: Requirements Engineering: Fundamentals for Ultra-Large-Scale Systems, and NSFC under Grant No. 60970017.

## 8. REFERENCES

- [1] N. Ahmadi, M. Jazayeri, F. Lelli, and S. Nesic. A Survey on Social Software Engineering. In *Proceedings of the 1st Workshop on Social Software Engineering and Applications (SoSEA)*, pages 1–12, 2008.
- [2] A. Aurum and C. Wohlin. The Fundamental Nature of Requirements Engineering Activities as a Decision-making Process. *Information and Software Technology*, 45(14):945–954, 2003.
- [3] C. Churchman. Wicked Problems. *Management Science*, 14(4):141–142, 1967.
- [4] D. Damian and D. Moitra. Global Software Development: How Far Have We Come? *IEEE Software*, 23(5):17–19, 2006.
- [5] B. Decker, E. Ras, J. Rech, P. Jaubert, and M. Rieth. Wiki-based Stakeholder Participation in Requirements Engineering. *IEEE Software*, 24(2):28–35, 2007.
- [6] A. Felfernig, W. Maalej, M. Schubert, M. Mandl, and F. Ricci. Recommendation and Decision Technologies for Requirements Engineering. In *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering (RSSE)*, 2010.
- [7] J. Goguen. Social Issues in Requirements Engineering. In *Proceedings of the 1st IEEE International Symposium on Requirements Engineering (RE)*, pages 194–195, 1993.
- [8] O. Gotel and A. Finkelstein. An Analysis of the Requirements Traceability Problem. In *Proceedings of the 1st International Conference on Requirements Engineering (RE)*, pages 94–101, 1994.
- [9] G. Goth. Ultralarge Systems: Redefining Software Engineering? *IEEE Software*, 25(3):91–94, 2008.
- [10] T. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
- [11] T. Gruber. Collective Knowledge Systems: Where the Social Web Meets the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):4–13, 2008.
- [12] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Groszof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *W3C Member submission*, 21, 2004.
- [13] S. Konrad and M. Gall. Requirements Engineering in the Development of Large-scale Systems. In *Proceedings of the 16th IEEE International Requirements Engineering Conference (RE)*, pages 217–222, 2008.
- [14] P. Liang, P. Avgeriou, and V. Clerc. Requirements Reasoning for Distributed Requirements Analysis using Semantic Wiki. In *Proceedings of the 4th IEEE International Conference on Global Software Engineering (ICGSE)*, pages 388–393, 2009.
- [15] F. Limpens, M. Buffa, and F. Gandon. Bridging Ontologies and Folksonomies to Leverage Knowledge Sharing on the Social Web: a Brief Survey. In *Proceedings of the 1st Workshop on Social Software Engineering and Applications (SoSEA)*, pages 13–18, 2008.
- [16] S. Lohmann, S. Dietzold, P. Heim, and N. Heino. A Web Platform for Social Requirements Engineering. In *Proceedings of Software Engineering (SE)*, pages 309–316, 2009.
- [17] W. Maalej and A. K. Thurimella. Towards a Research Agenda for Recommendation Systems in Requirements Engineering. In *Proceedings of the 2nd International Workshop on Managing Requirements Knowledge (MaRK)*, 2009.
- [18] L. Northrop, P. Feiler, R. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, D. Schmidt, and K. Sullivan. *Ultra-Large-Scale Systems: The Software Challenge of the Future*. Software Engineering Institute, 2006.
- [19] H. Ossher, D. Amid, A. Anaby-Tavor, R. Bellamy, M. Callery, M. Desmond, J. D. Vries, A. Fisher, S. Krasikov, I. Simmonds, and C. Swart. Using Tagging to Identify and Organize Concerns during Pre-Requirements Analysis. In *Proceedings of Workshop on Early Aspects at ICSE (EA)*, pages 25–30, 2009.
- [20] B. Ramesh. Factors Influencing Requirements Traceability Practice. *Communications of the ACM*, 41(12):37–44, 1998.
- [21] B. Regnell, R. Svensson, and K. Wnuk. Can We Beat the Complexity of Very Large-Scale Requirements Engineering? In *Proceedings of the 14th International Working Conference Requirements Engineering: Foundation for Software Quality (REFSQ)*, pages 123–128, 2008.