Cooperating with a Non-governmental Organization to Teach Gathering and Implementation of Requirements

Gregor Gabrysiak, Regina Hebig, Lukas Pirl, and Holger Giese
Hasso Plattner Institute at the University of Potsdam, Germany
{prename.surname@hpi.uni-potsdam.de}, Lukas.Pirl@student.hpi.uni-potsdam.de

Abstract

Teaching Requirements Engineering needs to be a realistic experience. Otherwise the students might not understand the repercussions of failing to gather requirements correctly. While simulated stakeholders are always a feasible option, only real stakeholders offer an authentic experience since only they are impacted by the system that is being specified. In this paper, we present our experiences of cooperating with the non-governmental organization (NGO) Wasserwacht. In a first requirements engineering course, nine graduate students elicited requirements by interviewing a dozen heterogeneous stakeholders. In a subsequent bachelor's project, four undergraduate students continued by implementing the software system based on these requirements. We discuss the authenticity of our requirements engineering setting, the influence of the collected requirements on the follow-up implementation project and how the Wasserwacht benefited from this cooperation.

1. Introduction

To successfully teach requirements engineering (RE), courses should include an epiphany for the students. While they are accustomed to hear that they have to listen to the demands of their stakeholders, only few of them have the chance to experience such a real interaction during their education. Having the opportunity, however, allows them to practically witness most of the common pitfalls firsthand. Some of these pitfalls might end an interview, others might even end a project. Therefore, Callele et al. [2] argued that RE education requires shocking moments to sensitize students to the importance of RE. According to the definition put forward by Glinz et al. [7], a stakeholder is someone who is impacted by the system, thus, more motivated to get what he wants or needs. To get a sense of what is at stake when they specify a software system for a real stakeholder, students need to interact with stakeholders who are impacted by the system. Thus, it is necessary to be able to interact with a stakeholder, to ensure that what is specified is what is required. However, the creation of such a setting is quite difficult as it needs to include stakeholders who are real or behave as authentic as possible. Only if these stakeholders have a real stake in what is elicited and specified, the students are confronted with the whole range of emotions.

Two obvious possibilities of providing access to stakeholders exist [9]; either the faculty can provide access to real stakeholders with real needs or the stakeholders, i.e. their motivation and needs, are simulated by other persons (virtual stakeholders [5]). As Callele and Makaroff point out, cooperating with real customers offers the best learning experience along with the highest risk [2]. As related risks they mention that real customers are rather uncontrollable, their needs may not match the learning objectives for the course, and their schedules are often too tight to be used in a classroom setting. Yusop et al. [16] further point out the high costs and achieving customer commitment. Still, while most RE courses rely on simulated stakeholders, there are also rare reports of cooperations with industrial partners [8]. As for computer science projects,
on the other hand, which rather focus on the implementation of software systems than their specification, multiple service-learning examples are documented [10, 11, 14, 15]. We circumvented most of the problems associated with real customers by cooperating with a non-governmental organization (NGO) – at first not for the implementation of a system, but instead the elicitation of requirements of such a system. They are usually quite focused in their area, thus, they can plan ahead for up to several years without the need to react to sudden influences of the economy. Also, some of their needs can be matched with what university students can offer, i.e. requirements specifications or even prototypical implementations [4]. We successfully realized a requirements engineering setting that we outlined in [4]. Thereby we cooperated with the NGO Wasserwacht (www.wasserwacht.de) of Berliner Rotes Kreuz (www.drk-berlin.de), a local chapter of the German Red Cross (DRK). Our setting tries to gain authenticity of real stakeholders with real needs without the hindrances one usually finds when trying to establish such a cooperation. Potential repercussions are limited, since the stakeholders themselves are volunteers volunteering some extra time to help their NGO to gain the anticipated benefit. Thus, although mistakes are still to be avoided at all costs, volunteering stakeholders are more understanding than economically constrained stakeholders would be. Scorse [11] pointed out that it is critical to define the deliverables and the time frame of such a cooperation. He also recommends that such a cooperation should run longer than one semester. In this paper we present our setting, which ran over a total of three semesters and two separate courses: a requirements engineering course in the summer term 2011 for specifying what the stakeholders require and a bachelor’s project for implementing the specified system in the two subsequent terms. Thus, by splitting up the requirements engineering from the actual implementation, we achieved a clear definition of what needed to be delivered.

While Section 2 presents the setting of how the requirements were gathered and then implemented, Section 3 illustrates how it was conducted. Then, Section 4 presents an evaluation of this setting from the perspective of the students participating in the elicitation. Section 5 discusses the impact of the collected requirements on a typically self-contained bachelor’s project, in which the software system was realized. Finally, Sections 6 and 7 close with lessons learned and conclusions.

2. Foundation: Settings as Planned

This section summarizes how we envisioned the requirements engineering setting [4], i.e. the first five steps illustrated in Fig. 1, the learning objective of both courses, and the practice of bachelor’s projects as they are conducted at our university.

![Figure 1: Setup of the presented course and the follow-up project (extended from [4])](image-url)
1. **Negotiate Value:** Only by negotiating what all involved parties expect, it can be ensured that a) the lecturers obtain authentic stakeholders who require a software system and b) the NGO gains a software system which is worthwhile for them. Stone and Madigan [14] stated, that this is a crucial point in setting up a successful cooperation. Only by managing the expectations of the cooperation partner and aligning these with course goals as well as the students’ skill level it can be ensured that the cooperation is mutually beneficial.

2. **Communicate Vision:** By inviting representatives of the NGO to present their overall vision of the project’s outcome at the first seminar session, all involved participants get an idea of what it is they have to specify.

3. **Elicitation, Documentation and Validation of Requirements:** After theory lectures, groups of three students each get access to distinct stakeholders who they have to schedule interview and validation sessions with.

4. **Create Requirements Specification:** Then, each group of students needs to hand in a specification of the requirements they elicit from the stakeholders they were assigned.

5. **Handover Results:** Since such a technical specification is by itself not useful for an NGO, a subsequent bachelor’s project is conducted in which undergraduate students have to implement a software system which satisfies the stakeholders’ needs.

**Learning Objectives:** Students in the requirements engineering course should learn how interviews are conducted, how stakeholders needs are documented in form of requirements specifications and how great quantities of such requirements can be managed for analysis and validation.

The learning objective of a bachelor’s project, on the other hand, is that students gain experience with practical projects and team work. A typical bachelor’s project is supervised by a professor and consists of four to eight undergraduate students who work together to either realize a desired software system or to extend an existing one in cooperation with a project partner. Project partners are always from technical industries, require software systems and, thus, rely on bachelor’s projects to implement and test new concepts prototypically. This is done over nine months: for the first three months, the students get to know the domain and the needs of their industry partner (approximately 10 hours per week). Then, the students start to realize their software system as a full-time project. The result is then presented and handed over to the cooperation partner. In these projects, regular meetings with the project partner and an agile development process are encouraged.

3. **Running the Setting**

In a course, it is rarely possible to create realistic experiences with real stakeholders, since once something is at stake 1) it needs to be achieved, 2) in an economic fashion, and 3) without upsetting anybody related to the project. Working with real stakeholders in a course requires that constraints on time and resources are not too strong. While such setup should lead to benefits for stakeholders, possible damage or loss of resources should be limited, as failure can never be ruled out completely. To achieve such a setting we chose the NGO Wasserwacht, which is a water patrol rescue service and relies on volunteers who work on weekends during the water sport season (between May and September). While Wasserwacht fulfills all properties of a real company (real needs, real processes, real restrictions on what needs to be done and real differences on how it is handled), a failure of the project comes with reduced risks, as volunteering stakeholders are more understanding than economically constrained stakeholders would be. Still, volunteers in Wasserwacht would be happy to use a prototype in the upcoming water sport season. Thus, we expected stakeholders to be highly motivated and drive the interviews to maximize the potential gains from the expected system.

Due to the prior arrangement, it was important to ensure that the assignment is self-contained to restrict the scope of the assignment for the students specifically by assigning a strict number of stakeholders to interview. Our setup is explained along the structure sketched in Fig. 1.
3.1. Negotiate Value

To support their communication operatives the Wasserwacht decided that a custom-made software system for transcribing radio transmissions and supporting mission (reports) should be built. Due to a lack of funds for software procurement, a specification of such a system by itself would not be beneficial for them [4]. We agreed that a seminar would be conducted to collect the requirements for such a system and a bachelor’s project during which students will build a prototype.

From an economic perspective, it would be perfectly reasonable to have only one team of students elicit, document, and validate all related requirements. From a teaching perspective, however, we wanted as many students as possible to attend. First, some of the students gather requirements for the first time. Consequently, at least minor errors are to be expected – errors which are unlikely to be repeated by other groups. Second, while Wasserwacht does not have funds to spent on procuring software, they have another valuable resource which all of their volunteering members already provide: time. Thus, as long as no other kinds of costs occur, the Wasserwacht only has to ask their members for volunteers who can spare time for two sessions of up to three hours each. Third, the members of the Wasserwacht form a heterogeneous group of stakeholders. Some of them are on active duty at least one day each week, while others are only available twice a year. Consequently, this variety of stakeholders had to be covered in more detailed since power-users have more sophisticated requirements, e.g. keyboard shortcuts, than other users. All of them are volunteering their time for the Wasserwacht – if the resulting system would be too demanding to use, it might reduce the stakeholders’ willingness to volunteer. As a reasonable trade-off between how many students would attend and how many stakeholders could be enlisted, we agreed on three teams. These negotiations started in 2010 and only required three meetings.

3.2. Communicate Vision

We invited representatives of the Wasserwacht to present their vision of the project’s outcome in the first session of the requirements elicitation seminar: they envisioned custom software to be built after their requirements which should reduce the required resources to handle and document all communications and to optimize the flow of information while exploiting the available features of the digital radio network which is currently introduced.

3.3. Elicitation, Documentation and Validation of Requirements

After six sessions of theoretical preparation, the nine participating students formed three groups. Each groups received names, roles, cell phone numbers and email addresses of four distinct stakeholders they had to interview and validate their requirements with. Next, the students were sent out to interview stakeholders and validate the requirements they documented afterwards with the stakeholders. The students were expected to go through one iteration of elicitation and validation in two weeks for each of the four stakeholders they were assigned to. Additionally, an elicitation and a validation session with the technician in charge was arranged to get detailed information about technical restrictions which might affect existing requirements or add new ones.

3.4. Create Requirements Specification

Subsequent, each group had to hand in a requirements specification complying with a template provided by the lecturer. Apart from a general motivation of the problem, all identified (groups of) stakeholders had to be described, the requirements needed to be specified using scenarios and use case templates, and reflections as well as recommendations for the implementations were required.

3.5. Handover Results (and Prepare Implementation)

To introduce the students of the bachelor’s project into the domain of the Wasserwacht and the overall vision, an initial meeting between the students from the requirements engineering seminar
and the four undergraduate students was arranged. Also, this meeting successfully concluded the requirements engineering seminar and kicked off the subsequent bachelor’s project.

Meeting the graduate students supported the students of the bachelor’s project to acquire the fine-grained requirements in the three documents more easily and allowed a steep learning curve. Further, it was neither required for the bachelor’s project students to do time-intensive interviews with the stakeholders, nor to translate the requirements into the IT domain as this was done during the requirements engineering seminar. Since the three documents were created based on the requirements of different roles in the Wasserwacht, the undergraduate students achieved a broad view on the requirements. This detailed knowledge enabled these students to differentiate between important and unimportant requirements, as well as to better identify contradicting or incomplete requirements.

3.6. Synthesize and Prioritize Requirements

Due to the heterogeneous form and style of the three requirements documents, one of the first tasks for the bachelor’s project was to unify all specified requirements. Naturally, these requirements were not identical throughout these documents. Although equivalent core requirements were found in all documents, some of them were present in two or in only one of the documents. Next, the requirements were transferred in a consistent form, so that they were uniformly accessibly and integrated into the software development process of the bachelor’s project. Thereby, the requirements were split into chunks of work with estimable effort, i.e. user stories, and were composed to components of the software. The amount and variety of these requirements did not allow for an implementation of all system details within the bachelor’s project. Therefore, the requirements were collaboratively prioritized with volunteers of the Wasserwacht in a separate meeting.

While requirements primarily define the problem to be solved, other software artifacts focus on the behavior of the proposed system [3]. To ensure that the subjective perception of the prospective users of the software is met, e.g. requiring an intuitive way of interacting with the interface, the students of the bachelor’s project developed paper prototypes of the software according to the core requirements. As shown in Fig. 2a, such prototypes were tested interactively with members of the Wasserwacht – the corresponding functionality was simulated by the students. This ensured further validation of the unified and prioritized requirements in a tangible manner and provided a good communication medium between the students and their stakeholders. The heterogeneity of the volunteers of the Wasserwacht was once again reflected in their differing workflows, experience with computers and, thereby, preference of interaction design. This led to the challenge of embracing and integrating the entire feedback without introducing conflicts to existing requirements.

The students were then able to define development runs (sprints) with exact requirements that had to be completed after each run. The approval of the correct realization of the specific requirements was done by members of the Wasserwacht in dedicated meetings and programmatically wherever possible. Through this iterative software development process with a high integration of the stakeholders, misunderstood or changed requirements were identified early and existing requirements were refined accordingly. During implementation, further questions arose which had to be clarified or decided on by an authority. Otherwise, such ambiguities would have blocked the development process. Thus, a dedicated contact person within the cooperating NGO was very valuable and allowed the students to get instant feedback (within 24 hours).

In conclusion, the requirements documents provided a solid base for the students in the bachelor’s project. When they initially received the extensive documents, the students in the bachelor’s project assumed a certain verbosity instead of a concise listing of requirements. During implementation, it turned out that the documents actually provided an excellent coverage of the requirements as well as some specialized, low priority and non-trivial requirements for specific roles.
3.7. Handover Software System

Before the actual deployment, it is essential to test software as a whole in a preferably realistic setting. This avoids unforeseen unfitness to certain situations. The students that developed the software prototype during the bachelor’s project went to the lifeguard station of the Wasserwacht at Wannsee with computers, having the software prototype installed and ready to test. The main components of this prototype are showed in Fig. 2b–2d. This minimized the effort for the members of the Wasserwacht and ensured a correctly configured software system. The students set up the workspace as it is planned for the future and let the users test the software parallel and non-interfering with the conventional work at the lifeguard station.

While evaluating the satisfaction of the users concerning the functionality and usability of the software prototype, it turned out that the satisfaction of a specific user seemed to be related to his or her individual experiences and priority of requirements. Nevertheless, no major discrepancies between the implemented requirements and the actual requirements were discovered. Due to the individual character of perception of intuitive interaction, some minor usability flaws arose during testing with different types of users. Most of the concretized requirements could be implemented instantly and improved the testing experience for the testees.

4. Evaluating the Stakeholder Authenticity Within the RE Course

Follow-up questionnaires were filled out by eight of the nine students from the requirements engineering course. Thereby, we used 7-point Likert scales (1 for strong disagreement and 7 being strong agreement) and open-ended questions. The goal was to assess how the students perceived the authenticity of the seminar as well as how satisfied they were with the overall setting.

4.1. Goals (and Expectations on a Realistic Experience)

We defined a set of expectations on an authentic setting to evaluate whether the described setup leads to realistic experiences. These expectations were evaluated using a questionnaire that the participating students filled out after they had handed in their specifications. Requirements engineers encounter aspects such as inconsistent statements by stakeholders, synonyms for the same concepts, and having to get to know the domain of the stakeholders. These expectations ($E$) are:
**E1** Students experience a barrier in terminology when talking to stakeholders.

**E2** Students experience inconsistencies between terminology used by different stakeholders.

**E3** Students experience a difference between the expectations of different stakeholders.

Also, since stakeholders are impacted by the system that is being specified (directly or indirectly), they will try very hard to get it specifically suited to their needs above all others – something that naturally will less likely occur in a simulated setting where the specification is only created for the sake of the specification. Thus, to rate the stakeholders’ authenticity as perceived by the students, we expected the following:

**E4** Students perceive the stakeholders as highly engaged and motivated.

**E5** Stakeholders who are engaged tend to forget the time when talking about the system.

**E6** Engaged stakeholders are anxious to represent their personal perspective.

**E7** Engaged stakeholders are likely to interrupt other stakeholders, to discuss or argue facts.

It was agreed with the Wasserwacht representatives, that students should visit stakeholders on-site at the station the software would be deployed to. Thus, in such an authentic setting the environment supports stakeholders, as outlined by Seyff et al. [12]:

**E8** Environmental stimuli enable stakeholders to remember details they would omit otherwise.

### 4.2. Meeting the Stakeholders

The interviews were conducted in three different locations. Either the students met the stakeholder at the Hasso Plattner Institute (in an informal seminar room or in a coffee shop in the same building) in Potsdam, in an office space of the DRK in Berlin, or directly at the lifeguard station of the Wasserwacht at Wannsee. While meetings in the evening during the week were usually either in the HPI or in the DRK office space, weekend sessions were conducted at the Wannsee station. Seyff et al. [12] argued about the importance of contextual triggers – environmental impressions which lead to the discovery of requirements which were omitted otherwise. According to the students, the Wasserwacht station offered the most stimuli to help stakeholders to remember details. In the HPI, no specific triggers were available ($\bar{x}=2.43, s^2=1.4$). The DRK office ($\bar{x}=3.87, s^2=0.99$), on the other hand, is used to prepare the service hours at the Wannsee station. While some essential documents or maps can easily be located or are obvious in plain sight, the station was ranked highest among the students who actually interviewed stakeholders in the station ($\bar{x}=5.43, s^2=1.4$, cf. Fig. 3), supporting expectation **E8**.

### 4.3. Authenticity of Stakeholders

A goal of the lecture setup was to reach an authentic experience for the students. Although the students’ questionnaires do not allow a quantitative proof, the given answers provide strong indicators, which qualitatively confirm the authenticity. As we expect for a realistic experience, students had to deal with barriers and inconsistencies in terminology (**E1 & E2**). They reported examples for which it took them up to half an hour to realize and resolve a misunderstanding. Further, misunderstandings occurred in both directions – also stakeholders had problems to deal with terms from the computer domain. Students disagreed with the statement that all members of Wasserwacht have

![Figure 3: E8: Students’ agreement on “Stimuli from the environment enabled stakeholders to remember details they would have omitted otherwise.” (n=7)](image-url)
a consistent understanding of domain specific terms ($\bar{x}=2.86$, $s^2=2.27$). Examples were mainly differences in terms for documents (e.g. ‘Matrix’ = ‘Alarmplan’) or even roles.

Moreover, students actually experienced differences between the expectations of stakeholders they interviewed ($E3$). Five of seven students disagreed with the statement that all stakeholders have the same expectations concerning the system ($\bar{x}=3.57$, $s^2=1.72$). An example was a disagreement, where a manager requested a statistical component for fuel consumption, while the boatmen were opposed. Generally, such differences concerned the actual extent of the functionality.

Students perceived the stakeholders as motivated and highly engaged ($E4$). They agreed that stakeholders were motivated ($\bar{x}=6.25$, $s^2=0.71$ in Fig. 4) and interested in results of the project ($\bar{x}=6.13$, $s^2=1.13$ in Fig. 4). Further, the students also observed that individual stakeholders forgot the time and were surprised about how much time actually elapsed in at least two of the interview sessions (supporting $E5$). They indicated the huge amount of information to collect as main reason ($\bar{x}=5.25$, $s^2=2.12$), while they did not agree to alternative explanations such as small talk or talking at cross purposes for taking longer than expected. Further, students rated stakeholders as anxious to represent personal perspectives ($\bar{x}=6.13$, $s^2=0.83$ for $E6$, cf. Fig. 4). Six out of eight students reported that their stakeholder sessions were interrupted by other stakeholders (supporting expectation $E7$). Thereby, interrupting stakeholders contributed documents, made additional comments, or gave insights by making jokes. Some of them expressed a different opinion than the interviewed stakeholder or suggested additional features for the system. In one case, an interrupting stakeholder even illustrated reasons for refusing such a system. As a final indicator for the authenticity of the experience, the students agreed that they felt responsible for a good solution. They acknowledged that (their) errors may quickly lead to a system which would not be used ($\bar{x}=5.86$, $s^2=0.69$).

5. Impact on the Bachelor’s Project

After the requirements engineering seminar ended, four bachelor students applied to participate in the follow-up project to realize the desired software system. We observed that the integration into the presented setting led to several changes for the bachelor’s project. Here we want to report about and discuss these experiences. As discussed in Section 2, a bachelor’s project is conducted with an industry partner which is, due to personnel costs, normally represented by one or two contact persons, only. These two contact persons act as the only stakeholders for the project. Thus, heterogeneous or conflicting requirements that have to be reconciled can only rarely be experienced by the participating students. Due to the requirements documents provided by the seminar students, this homogeneous character changed. Instead of two stakeholders, the bachelor students now had access to requirement specifications from 13 stakeholders. Thus, they experienced the task of balancing different requirements. Further, they now experienced responsibility for also questioning and discussing the contact persons’ point of view, if it contradicted the requirements specifications.

In bachelor’s projects, all participating students are at the same level of training. They have no access to more experienced students or even to students who are familiar with the industrial partner’s domain. Students have to bridge the gap between computer science and this domain by themselves. This restricts possible cooperations to technically oriented industry partners. Due
to the combination with the preceding requirements engineering course, the new setting enabled students in the bachelor’s project to profit from the more experienced graduate students. They provided not only the requirements specifications, but further participated in two sessions where they shared their personnel insights on the problem to solve and provided feedback on the bachelor’s project’s progress. Even more important, the setting allowed us to perform the bachelor’s project with an industry partner that is not technically oriented. Thus, using results of an requirements engineering course, enabled the students to bridge a broader gap between domains.

6. Lessons Learned

Some obstacles had to be overcome during this setting. First, it was not always possible for the students to reach stakeholders for making appointments for interviews. The students spontaneously recruited Wasserwacht members who worked nearby and were interested in the topic during their scheduled interviews. Based on these experiences and student feedback, we learned that such ad-hoc replacements are worthwhile and should be encouraged. Second, there is always a risk involved when the responsibility for a project is handed over to another team. In our setting, however, this did not lead to problems. We experienced positive effects of the shared presentation between the courses as well as the fact that some participants of the seminar were also reachable for follow-up questions. Third, due to the given setting, there was a longer period of time between the elicitation of the requirements and the delivery of the software. The risk of the Wasserwacht loosing interest or getting dissatisfied was successfully addressed through regular meetings with the contact persons.

7. Conclusions

In [4], we asked how lecturers can make time and energy of real stakeholders committed to an RE course worthwhile. After the requirements had been handed in and the specifications was prototypically implemented, we are able to reflect on how our setup of employing real stakeholders in an RE course worked out for the participating students as well as for the cooperating NGO.

During the seminar, many of the expected indicators of authenticity were observed. Further, the students’ feedback strongly indicated that our expectations for realistic stakeholder contact were fulfilled. Subsuming, we were able to evaluate the hypothesis that the course setup can provide the students with a realistic requirements engineering experience.

As argued by Madigan [13], the communication between students and their stakeholders needs to be monitored to ensure success. The students agreed quite uniformly, that the supervision during the seminar was good ($\bar{x}=6$, $s^2=0.82$). Nevertheless, when asked how the supervision can be improved, they asked for more detailed evaluations concerning individual interview sessions and more structured feedback sessions in-between. While we accompanied 10 of the 24 sessions and provided feedback after these sessions, the students indicated that they wished for more feedback. However, increased supervision and more feedback generally decreases the feasibility of such a course setting from the lecturer’s point of view. This challenge was solved by incorporating experienced students who participated in a similar requirements engineering seminar beforehand [6].

The implementation of the required software system in the second course was uniformly considered successful. In qualitative stakeholder interviews, four prospective users of the system the students developed praised the usability and were generally content with the scope of the system.

Only few of the Wasserwacht participants are capable of developing software themselves which obstructs a handover as proposed by Brügge and Gluchow [1]. As argued by Scorce [11], this might lead to disadvantageous commitment of the university for the maintenance of the software system. Thus, the software system will be published under the GNU General Public License on the open-source platform Github\(^1\) to enable other software developers to contribute to the system.

\(^1\) https://github.com/wasserwacht-berlin/ww – The source code will be available in summer 2013.
The described setting of employing two separate courses as part of the cooperation with an NGO can be summarized as successful and insightful. From the first course, the Wasserwacht gained valuable insights into the heterogeneity of their processes and how they are executed. After the second course, volunteers working for the Wasserwacht obtained a customized software prototype to support them in their work—a prototype which saved them “five-figure development costs”, as they estimated. Consequently, they emphasized that they would like to continue this cooperation in follow-up projects. Thus, we can acknowledge that they considered the time and energy that they committed to our teaching endeavors as worthwhile.

We initially described our setting for recruiting authentic stakeholders to participate in requirements engineering sessions in [4]. This requirements gathering setting was successfully tested with an NGO as described in this paper and since then repeated with a software startup [6] focusing on scalability issues concerning the supervision of the students during their stakeholder interactions. Although, the repeated execution of the setting with different NGOs would be a prerequisite for generalization of the results, the run reported here shows that such a cooperation can be successful.

Acknowledgment
The authors would like to thank Christoph Kühnl and Andreas Seibel for supervising some of the elicitation sessions. Also, we are grateful for the participation of all volunteers of the Wasserwacht, especially Steffen Rönnecke and Peter Hennig. Further, we greatly appreciate the time and effort of the participating graduate students and Matthias Bastian, Catharina Hahnfeld, and Christoph Matthies.

References