Open Source Requirements Engineering

Barbara Paech
University of Heidelberg
Institute of Computer Science
Germany - 69120 Heidelberg
paech@informatik.uni-heidelberg.de

Bernd Reuschenbach
University of Heidelberg
Psychological Institute
Germany - 69117 Heidelberg
reuschenbach@uni-hd.de

Abstract

This paper describes an experience in requirements engineering for an open source E-Learning tool selection. The process meets the challenges of software selection such as the intertwining of requirements acquisition and product evaluation, the level of detail or the information collection. In addition, it is user-driven and just-in-time. Despite their time constraints the stakeholders were able to contribute throughout the whole process. The decisions and the effort for preparing these decisions were postponed to the latest possible point in time. This process seems also suitable for open source software development

1. Introduction

Requirements Engineering (RE) for selection of standard software is different from requirements engineering during a complete software development project. As described in [1], for selection requirements acquisition can be intertwined with product evaluation. The challenges are the high number of potential product candidates, the balance between high-level and detailed requirements and test cases, the prioritization of the requirements and the information collection during product evaluation. Each requirement can be immediately verified against the products. Thus, on the one hand requirements documentation can be less detailed. But on the other hand all information collected during product evaluation must be captured carefully so that the final choice can be justified adequately.

This paper describes an experience in RE for the selection of a university-wide open source E-Learning software at the University of Heidelberg. The applied process has exemplified and met all the RE selection challenges described above.

We call the process *open source requirements engineering* for several reasons:

- The requirements concern open source software (OSS).
- The spirit of the RE effort was similar to OSS efforts: user-driven and just-in-time [2].
- The process also seems suitable for RE during OSS development

The rest of the paper outlines the process, illustrates it by means of the specific case study, and provides some lessons learned. The conclusion discusses the application of the process for RE of OSS development.

2. The process

A stakeholder group representing management, developers and users from different departments carries out the process. Developers in this sense install and administrate the OSS and possibly contribute some code to it. The selection of the stakeholders is based on expertise and motivation. The users should have experience with similar software. They should all be able to contribute to an electronic discussion. On one hand the stakeholders – as usual – have very little time for the RE process, on the other hand the RE process has a fixed deadline. So the process needs to be very easy to accomplish.

The process consists of five steps:

Setting the stage: In the Kick-Off meeting the stakeholders introduce each other. A moderator is chosen out of the group. The stakeholders determine the general goal of the process (selection of an OSS). Furthermore, the next process steps need to be agreed on. As mentioned in the introduction the process challenges are

- coping with the *high number* of product candidates
- *intertwining* of RE and product evaluation
- the *level of detail* of the collected requirements
- the *precision* of requirement *prioritization* and
- the *recording of the rationale* for the final choice.

Requirements acquisition and product pre-selection and evaluation are performed concurrently. The pre-selection is based on the literature and personal experiences of the stakeholder. Due to the direct involvement of the stakeholders and their experience with similar systems, *the written requirements can be high-level*. The requirements can be gained through brainstorming. This brainstorming should already yield a first grouping to ease further processing. As the stakeholders have and represent different interests an *open and constructive communication* is essential to take care of the "Human Factor" [3]. Thus, at this stage all proposed requirements should be collected, but not discussed in detail. Acquisition of requirements details is postponed to OSS evaluation. This provides the benefits of scenario-based RE without the effort to produce detailed scenario descriptions.

As the stakeholders have very diverse backgrounds, it is important to *develop a common language*. Sorting requirements into different categories helps to develop a common "big picture". Furthermore, because of the diversity, prioritization is important *to make explicit different needs*. So priorities should be captured from the very beginning, but – as explained later - not emphasized too much.

Information collection is done on the white-board. Digital cameras allow capturing the information on the white-board and distributing the results to all stakeholders.

Requirements consolidation: Since a brainstorming result is rarely complete, and since the white-board-result is typically only understandable to the participants of the session, it is important to consolidate the requirements. This can be done by each stakeholder between meetings. Thus, a table relating the brainstormed requirements to stakeholder priorities is distributed electronically. The requirements in the table should be arranged according to the groups identified in the brainstorming and given a unique identifier. The stakeholders can comment on existing requirements or add new ones. They can also exchange any kind of electronic documents (e.g. in the Web) explaining the requirement. Furthermore, each stakeholder is asked to prioritize the requirements on a scale of 1 (very important) to 3 (not so important). Rating of importance typically leads to a "ceiling effect": Every dimension is "very important". Therefore, prioritization is given an operational interpretation: *it reflects the promised effort for evaluation of this requirement*. If the stakeholder rates the requirement as very important, s/he is committed to an intense evaluation of the requirement.

At this step, one could also think of using a Wiki for requirements discussion. However, then discussions would need further consolidation to allow prioritization. Furthermore, as the requirements are high-level the discussions would either be high-level (and thus, not of much value) or go into detail. As the details have been postponed to the evaluation, it is recommended *to also postpone the intense requirements discussion to the evaluation step*.

Concurrently with the requirements consolidation, the potential OSS candidates are pre-selected guided by the high-level requirements, recommendations in the literature, experience with same problems in other institutions and experience of the stakeholders. These candidates are proposed and collected also in the electronic discussions.

Determination of the evaluation process: In a second meeting the consolidated requirements are shortly discussed to explain all the added requirements. Then the stakeholders determine the OSS to be evaluated. To reduce effort, but not constrain creativity, at least 2 and not more than 4 OSS should be evaluated. The job of the moderator is to avoid early jumping to conclusions and to control the impact of "opinion leaders". At the end of the meeting the details of the evaluation process are fixed.

All stakeholders should be able to evaluate at least part of each OSS. The evaluation has to be carried out as homework. Stakeholders should concentrate on the requirements important to them. The results can be captured in the table again, where each stakeholder can rate the requirement for each OSS. Excel-features are very important at this point allowing all stakeholders to view all ratings of each stakeholder at one glance and to view the ratings of all stakeholders for one OSS at once. The rating should be more finely grained than the prioritization, for example marks 1 to 6. Obviously, even the detailed ratings are not sufficient to understand the details of the evaluation. Therefore, *the detailed thoughts of the stakeholders during the evaluation should be captured*. In particular, this provides details to the high-level requirements. This can be supported e.g. through a Wiki, so that stakeholders can discuss their experiences. If the OSS supports collaboration, it should be arranged that several stakeholders evaluate the collaboration features jointly.

Distributed evaluation: Again the table is distributed electronically. The selected OSS are installed. Stakeholders evaluate the OSS and capture the ratings in the table. Discussions and ratings should be broadcasted to all stakeholders so that they are *motivated to join and contribute their evaluations*.

Final selection: In a last meeting the results of the evaluation are discussed. At first, each stakeholder should describe her/his overall impression. Then the table should be analyzed: Has everybody rated the requirements that are important to him/her? Are there required product features which have a very negative or very positive rating by all stakeholders? Which requirements have very diverse ratings? These conflicts should be discussed and resolved jointly. If necessary, the requirements could be directly evaluated by using the OSS in the meeting. This usage could help demonstrate features, which may have been overlooked by some of the stakeholders. Based on the individual prioritization of the requirements and the joined rating of the requirements fulfillment of each OSS, each stakeholder can select the OSS, which fits best to her/his needs. Then a joined selection can be based on the individual selections.

Clearly this selection can be quite controversial. Based on direct assessment of the OSS, the resolution of fulfillment rating conflicts should be not too hard. However, if the prioritization of the requirements differs very much for different stakeholders, it is necessary to prioritize the stakeholders. This step could have already been done during requirements consolidation. However, *this would destroy the spirit of the whole joined effort* (as some stakeholders are characterized as more valuable than others). So, this step should be postponed to the last possible point of time. If at all, the priorities of the stakeholders should be determined jointly.

Evaluation of product usage: After the roll-out of the selected OSS, the stakeholder group could meet again, to discuss their experiences. This could lead to further requirements which the developers can try to realize within the chosen OSS. This step has not been performed in the case study and thus is not considered part of the process described in the paper.

3. The Case Study

This section describes how the process sketched above was performed for the E-learning OSS selection at the University of Heidelberg..

The stakeholder group consisted of 15 persons with different interests und competencies within E-learning.

The process was performed in 3 meetings within 8 weeks. Each meeting took between 2 to 3.5 hours. Additional time for individual testing and evaluation had to be added.

Setting the stage: The reason to start the OSS selection was the replacement of the currently applied E-Learning system .LRN. Problems regarding support and licenses necessitated a change. An invitation to participate arouse among a group of 15 persons from different departments. Students were not part of this group, but involved later in the evaluation.

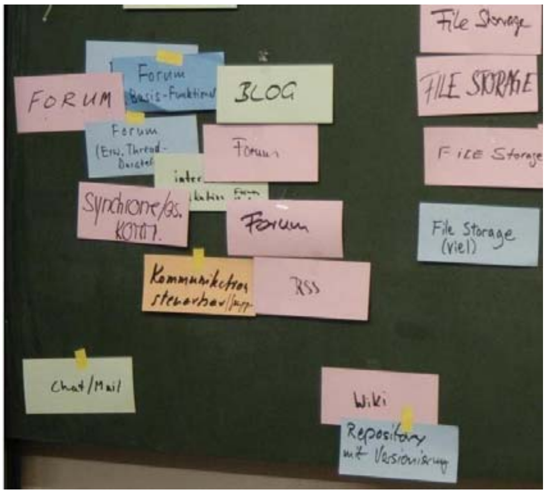


Figure 1: Part of the brainstorming result

brain-storming	solution technology	literature comments	require-ments
1.Kommunikation (DozentIn und Studierende)			
synchronous /asynchronous communication	RSS		1.2. Chat
			1.3. Mail
			1.4. Forum
			1.4.1. flexible structuring of discussions
			1.5. Calendar
			1.6. Wiki
			1.7. Blog

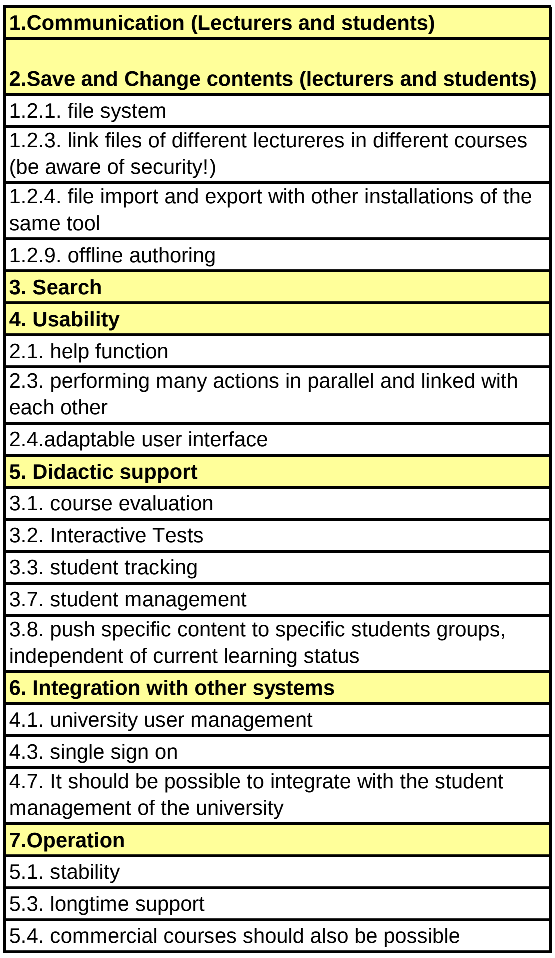
Figure 2: Part of the consolidated requirements

A psychologist who at first was not aware of the presence of a requirements engineer moderated the first meeting. He used an open communication style to ensure participation of every stakeholder from the beginning and started by a round of introduction. Everybody had to express his/her expectation and experience with E-learning as well as his/her role in the group. Then visions of further steps of the process were determined. The moderator proposed the steps to be performed in the meeting, but the details of the further meetings were left open. It was consensus to establish a list of individual important requirements through brainstorming. As can be expected from the different interests and experiences of the stakeholders, the requirements collected varied considerably in the level of detail. After 3 hours meeting, it became clear that there was not enough time for a more detailed requirements specification or joined prioritization. So prioritization was assigned as homework. The requirements engineer took responsibility to consolidate the requirements.

All in all 36 high-level requirements were elicited. Part of the brainstorming outcome is shown in Fig. 1. This part was consolidated into the Excel-sheet shown in Fig. 2. This figure shows the overall organization of the Excel-sheet and very high-level requirements. Together with the requirements we captured typical solution technologies (if already known to the stakeholders) and comments from literature. In Fig. 3 the other categories and some examples for requirements in these categories are shown. Note that numbering of the categories was added after the individual numbers of the requirements were determined. Clearly, these requirements are quite high-level. All of them were discussed in detail during the session, but these details were only captured later during the evaluation.

Requirements consolidation: In this phase new requirements were rarely added, as the most important ones had been successfully elicited during brainstorming. 13 of 15 stakeholders contributed their priorities, however, some with explicit disclaimers on the corresponding evaluation effort. One stakeholder left the team as the department decided to stay with the current tool (.LRN), because the effort to switch to a new one was deemed too high.

Determination of the evaluation process: While there are many E-learning OSS, there are 2 quite obvious choices nowadays: Ilias and Moodle. Publicity, commonness and positive experiences with the OSS in similar institutions were relevant factors for pre-selection. As the stakeholders were keen on a quick decision, no other OSS were evaluated.



1.Communication (Lecturers and students)
2.Save and Change contents (lecturers and students)
1.2.1. file system
1.2.3. link files of different lectureres in different courses (be aware of security!)
1.2.4. file import and export with other installations of the same tool
1.2.9. offline authoring
3. Search
4. Usability
2.1. help function
2.3. performing many actions in parallel and linked with each other
2.4.adaptable user interface
5. Didactic support
3.1. course evaluation
3.2. Interactive Tests
3.3. student tracking
3.7. student management
3.8. push specific content to specific students groups, independent of current learning status
6. Integration with other systems
4.1. university user management
4.3. single sign on
4.7. It should be possible to integrate with the student management of the university
7.Operation
5.1. stability
5.3. longtime support
5.4. commercial courses should also be possible

Figure 3: Further requirements examples

Because of time constraints and different focuses of the stakeholders the evaluation of the requirements was divided to four groups which evaluated different parts of the requirements for both OSS. Beyond the determined evaluation each stakeholder was free to casually rate the other requirements. As both OSS support collaboration, it was obvious to agree on commonly evaluating these features. Furthermore, instead of a separate Wiki the requirements discussion was to be captured in the OSS themselves. So the requirements discussion could help to evaluate the collaboration requirements in detail. This meeting only lasted 2 hours.

Distributed evaluation: Due to the holiday season over Christmas the evaluation did not start right away. In the week before the final session, discussions and evaluations were intensified. At this point stakeholders encouraged their students to evaluate the 2 OSS. Two student groups worked intensively for some sessions with the tools and contributed their evaluation. Again, almost all stakeholders contributed their ratings. Some of them provided detailed discussions. Others provided summaries of their evaluation thoughts. Only in these discussions and thoughts details of the requirements were explicitly captured.

Final selection: In this case the choice was very obvious. All stakeholders agreed that one system was much easier to use than the other and both satisfied the minimum requirements. None of the OSS met the requirements by 100% and some relevant requirements were difficult to evaluate by the stakeholders. However the decision for the more usable OSS was clear. Many stakeholders stated that their final evaluation was congruent with the first impression during the evaluation. Interestingly enough, usability had been an important issue right from the beginning of the process. However, not until this meeting it became clear to all that ease of use is the most essential requirement (for all stakeholders), as the OSS should be adopted by as many teachers and students as possible (to support wide-spread adoption of E-learning). So, the need for this adoption only became important when faced with a very difficult to use OSS.

4. Lessons Learned

The whole process was very efficient. All stakeholders were quite satisfied with the process and the result. In particular, it was an important insight for them that requirements engineering competence (provided by the moderators) significantly supported the process.

Compared with [1] the process was clearly much less thorough. But this seems adequate for the purpose.

The process was *user-driven* because stakeholders really perceived the need for the system and had the expertise to define requirements and to evaluate the OSS [4]. Participation in every step, including the requirements management is important and assures success.

The process was *just-in-time*, as all questions were solved as late as possible:

- The requirement details were not determined until evaluation.
- The prioritization of the requirements was associated with the necessary evaluation effort.
- The OSS systems were chosen after the high-level requirements had been determined.
- The stakeholders' prioritization was postponed to the final selection.

Going into more details of the process we observed the following issues:

The table listing the requirements with the categories provided an overview, but it did not guarantee that all requirements were evaluated thoroughly. It mainly helped to make the process transparent, so that every stakeholder was assured that his/her needs were taken seriously. Furthermore, it helped to develop a common vocabulary and served as a checklist for the stakeholders who had less experience than the others.

The table with the priorities was important to focus the discussions and to point out the different goals of the stakeholders. However, it was never used to calculate any overall priorities or ratings. The requirements prioritization was controversial, but this was never discussed in detail. Before evaluation there was no need for that and after looking at the selected OSS it did not matter any more.

Given the low number of requirements the distributed consolidation worked very well. For more complex systems we suspect that another consolidation meeting is necessary to discuss all added requirements.

The stakeholders were quite willing to contribute, but still it was important that the requirements engineer explicitly requested the contributions several times.

Participation of the stakeholders and a constructive communication culture is an important contribution to a good selection. Attention to the “human factor” in requirement engineering assures satisfaction and commitment. [3]

The availability of the software focused the whole process much more than expected. It is well-known that early prototyping helps to elicit the detailed requirements, so this was anticipated also for the selection process. However, it was surprising how much the availability also helped to reduce the effort for requirements prioritization. Maybe we were just lucky, as the final choice was so unanimous. But based on this experience we also expect that even in more controversial situations the prioritization effort will be reduced strongly.

The table and the documented evaluation discussions serve as the rationale for the selection.

This rationale is mainly for the stakeholders involved. Other people will have difficulties to understand the details of the decision. In this case, this was not necessary. Only a summary for the management was requested. In general, we suspect that the table and the discussions have to be merged and consolidated to serve as long-term documentation of the RE process.

5. Conclusion

We have presented a successful process for open source requirements engineering. Success is measured in terms of stakeholder satisfaction with and commitment to the process. The selected OSS has just been rolled-out, so satisfaction with the OSS in operation cannot be measured.

As mentioned in the introduction, we believe that in addition to OSS *selection* this process could also be used during OSS *development*. As described in [5] OSS requirements arise from discussions on the web and are available for open review and consolidation. They emerge during the use of the already implemented software. Sometimes they are only collected retrospectively. We believe that our process could easily be applied whenever a new call for contributions to the OSS is to be issued. All meetings would be held electronically. So during **setting the stage** the high-level requirements can be collected based on existing discussion threads and a virtual brainstorming session. The **consolidation** with the table is one specific form of the usual requirements review and consolidation. It could be of help to focus the contributions directly, avoiding too widespread discussion threads. A goal-oriented leadership of the open process will be useful. The **evaluation step** could be performed twice. In a *first* round, the requirements are detailed by looking at the “holes” in the existing software. Then the requirements are published and contributions are asked for. The selection of the best contribution can then be performed as described in our process. So the evaluation step is carried out a *second* time and at the same time the details of the requirements and evaluations are discussed electronically. The selection can proceed as described here. The user-driven and just-in-time spirit of the process seems to fit well to OSS development.

Acknowledgements

We thank all stakeholders involved in the E-Learning OSS selection process for their contributions which made this a really worthwhile experience.

References

[1] Maiden NA, Ncube C (1998) “Acquiring COTS software selection requirements”, IEEE Software March/April, pp. 46-56

[2] Berglund E, Priestley M (2001) “Open-Source Documentation: In search of user-driven, just-in-time writing“, SIGDOC; pp. 132-141,ACM Press

[3] Viller S, Bowers, J, Rodden, T (1999) “Human factors in requirements engineering: A survey of human sciences literature relevant to the improvement of dependable systems development processes“, Interacting with Computers, 11 (9), 665-698.

[4] Thomas D, Hunt A (2004) “Open Source Ecosystems”, IEEE Software, July/August, pp. 89-91.

[5] Scacchi, W (2004) “Free and open source development practices in the game community”, IEEE Software January/February, pp. 59-66