

Recent Trends in Software Support for Online Communities for Teaching and Research Projects in Higher Education

Daniel Kadenbach and Carsten Kleiner

University of Applied Sciences and Arts Hannover, Germany
{daniel.kadenbach, carsten.kleiner}@fh-hannover.de

Abstract. This paper investigates the evolution of software project support for teaching and research projects in higher education. It therefore analyses the results of two surveys conducted with students of the department of computer science of the University of Applied Sciences and Arts Hannover in February 2009 and November 2010. The surveys focused on examining the experiences and requirements of the students. Beneath mere technical requirements the surveys also investigate the acceptance of social software elements in project support. Since containing a core of equal questions the answers to the two surveys allow to deduce trends. These trends towards more sophisticated supportive software tools and environments are further investigated not only to allow a better support for projects, but also to make projects more visible and valuable, preserving their achievements and making them accessible for others.

Keywords: software support, projects in higher education, online collaboration, social software.

1 Introduction and Motivation

The University of Applied Sciences and Arts Hannover offers bachelor and master's degree programs in applied computer science. Within these programs a multitude of different projects is conducted each semester and beyond. Similar to many institutions, in addition research projects of longer durations are also executed. These projects differ greatly in their type, contents, lifetime, the number of people involved and their goals. They vary from one person projects such as seminar works, comprehensive homework or final theses to those with many project members like capstone projects or scientific projects led by the faculty or staff.

In particular in the scope of software development projects, but also beyond that, a trend becomes more and more apparent to make use of sophisticated supporting software tools, which can provide an enormous benefit if a suitable set of tools is chosen for a project, accepted by the project team and used correctly. This paper analyses how project support by software in the department has changed over time and which possibilities this change offers, but also which challenges are involved with it.

By this other institutions shall be given the opportunity to have an insight into the gathered experiences and findings, especially by drafting trends in the usage of supportive software tools for projects, and to encourage an information exchange leading to an improved situation for projects in all higher education institutions.

1.1 Example: A “Traditional” Project

A software-development project for students without supportive software-systems shall serve as an example for this evolution. Such a project could begin with a few meetings of all project members in which the requirement analysis is conducted, the software-system is designed and further tasks and responsibilities are assigned to the participants.

The students henceforth will work on their own on their particular tasks between further team meetings; or they could use frequent meetings to work together in a room on the development of the software. The latter approach, although offering far better communication, will be more expensive in terms of time effort, because the students need time to reach the team’s working-room and because the meeting place usually lacks some comfort, so probably students will try to avoid the additional effort if possible. In addition it is sometimes difficult for the institutes to provide each project team with an individual lab room due to space constraints.

Finally the students will put together the modular developed parts of their application and in the best case will test the complete system extensively, and maybe even document it properly if enough time is left. The results of their work will be delivered and if applicable also presented in front of other students, but in most cases they will not be trackable even after a short time. Thus the value of the software developed itself as well as lessons learned during project execution will soon be lost. No sustainable project improvement process is achievable for student’s projects on one hand. On the other hand projects cannot build on previously implemented software in order to achieve higher goals.

1.2 Example: A Project with Software-Support

This process can be improved in many ways by supportive software systems for software development projects, which appeared mainly within the development of Open Source Software, such as version control systems, wikis, trackers and others.

Because they are used in nearly every real software development project nowadays, the knowledge of these different tools has become an absolute necessity in an up-to-date study program in applied computer science to improve student employability. In addition these tools also support the development project itself and thus may improve the results. That implies the knowledge of their operating mode, their advantages and disadvantages in different projects and their handling. In order to make the right choice in the upcoming projects, a student has to know about the qualities and features of a variety of supportive software systems.

In using the adequate supporting functions, totally new possibilities could arise in this exemplary software development project. To just name a few:

- The use of a distributed version control system like Subversion offers the possibility to work together on the whole source-code of the application from any place. Additionally, one can be informed about the results of each member of the team, avoid conflicts of different versions and make the results of the projects accessible and useful even after the run-time of the project.
- A wiki provides a central, up to date, documentation which matches the specific requirements of the project. On one hand the created content can also be accessed and

preserved beyond the project life-time. On the other hand it can always be changed and updated easily at all times. It is therefore remarkable how simple it is for all team members to manage and find project related information which increases the likelihood that a useful documentation is created.

The wiki can provide an external online presentation of the project for other students or even on the Internet as required. It increases the project's visibility and, if desired, can be a way for the project to receive feedback.

- Issue-tracking systems can be used to keep track of bugs, feature requests, or even the planned tasks, forming milestones of packages of this tasks. Team members can be attached to every issue as responsible persons for it. Also the progress of the issues and milestones and therefore of the project itself can easily be tracked by other developers, instructors or even customers.

The motivation of the students to reach a high quality in every aspect of the project can be increased through a greater visibility of the project and therefore a much higher possible value and benefit of the project outcomes. This effect increases the more the project generates a value for any type of customer, be it a research project at the school, a company or some non-profit organization.

Especially the last aspect allows to use the mechanisms and the potential of social software. In social software system the users are often inherently rewarded by gaining reputation when they create content, which is useful for others (see [12]). Thus the motivation of project participants for providing and preserving quality information and improving existing content can be increased.

In section 2 we will be reviewing related publications on software support for student projects. In section 3 the analysis of the conducted surveys follows, starting with a general description of the surveys, describing their methodology and summing up results from which trends are deduced. Also further observations of the daily project support practise will be enclosed at the end of this section. Section 4 sums up the results and deduces conclusions from them. Finally section 5 will draft future work and the focus of future investigations.

2 Related Work

Stanfill et al. document their experiences with version control systems and project management tools like Trac [3] for the use in capstone projects for product design teams in [8]. Also the use of wikis such as the MediaWiki is investigated. Problems and solutions with these tools are described. The acceptance of these tools and their benefit is analysed. The importance of learning effective collaboration techniques with these tools is stated, which will also be the case in this paper. Also Radermacher et al. investigate the benefits of using Subversion and Trac to improve capstone projects [9]. Munson investigates the use of wikis to create an organizational memory of projects which is also a subgoal of our efforts [7]. Grudin et al. explore the challenges in using wikis in scientific and engineering organizations, and at least a part of the results also apply in our environment [4].

Denzer describes the importance of teaching students communication skills and tools, which also applies for the supportive software systems described within this paper [1].

Another interesting aspect in which the use of collaborative, supportive online tools can greatly improve the results is their use in distributed student projects across multiple universities, which is investigated in [2] [10].

Also this paper partially continues to investigate and substantiate some directions of our previous one in [6], which focused more on the aspect of social software.

3 Analysis

The following considerations are quantitatively based on two conducted surveys, but also qualitative experiences which were gained in two years of daily project support practise are finally shared.

In these two surveys (the first one carried out in February 2009, the second one in November 2010) all students of the department were invited to give their answers to an online questionnaire¹. The aim was to gather the experiences of the students with supportive systems for their projects and their requirements for the support they desire for their future projects to meet their needs as good as possible. Of course it had to be considered, that a probably significant part of the students will not be used to many supportive systems and therefore will not be able to estimate their possible value.

Nevertheless both surveys allow to take a look at the level of awareness of the students in this matter, and at their demanded requirements for project support. The first insight allows to further optimize the project offerings of the department, the big advantage of the second one is that knowing which systems the students want to use themselves means to know which system will reach the highest acceptance among them. Therefore they will be used with the greatest motivation. But also this is highly dependent on the knowledge of the students about the advantages and disadvantages of different system components, so they are able to make a good choice at all.

The surveys however exceeded mere technical requirements. So they also investigated the form of administration the students demanded for their project support. They were asked if some administrators should set up functions for a project on inquiry or if there should be a system where the students could do that autonomously. The acceptance of social software was also analyzed. The students were asked if they are willing to use social functions in their project support – thus generating content for other students through their project work and offering more ways of interaction between projects – and also how deeply they are interested in accessing the contents of other projects.

Furthermore because the core of the questions in both surveys was identical, it is possible to conclude developments and trends in the answers by comparing the results. So the summed up results will be discussed, right after a short look at the methodology of the surveys has been taken.

3.1 Methodology

Both surveys were carried out online with the open-source tool limesurvey [11]. Therefore the students were invited through mailing-lists to take part in the particular survey.

¹ The results can be found at

<https://proanvil.inform.fh-hannover.de/surveys>

The first survey consisted of 30, the second of 32 questions, each within the categories: Experiences of the students, desired supportive systems, acceptance of social functions, administration of the software support and miscellaneous questions (like for example the usage of virtualisation to provide server instances for projects).

Most questions of the survey could be answered with yes, no or abstention. Sometimes the students could choose between a set of predefined answers (for example when asked which version control system they prefer), but were also able to add further answers themselves.

Of the about 250 active students in the two programs 60 participated in the first survey and 94 in the second one. This return rate is pretty good as the majority of students is enrolled in the first semesters of the program where no projects are executed. Thus among the students active in projects the return rate is significantly above 50%.

3.2 Results of the Surveys

Detailed results for all of the questions cannot be presented due to space constraints. Thus we focus on a summary of the most interesting results here:

Basic Acceptance. The surveys showed that 100% of the participants desire supportive software systems for their projects, so that such systems can expect a high acceptance. However it can be assumed that some students who would refuse supportive systems for their projects have not taken part in the surveys in the first place but their number is probably rather small compared to the ones who just did not care about it enough.

But this is not really a drawback of the surveys, because supportive systems should always only be an offering for the students, they can choose to use them if they want. They should never be forced to use them in their projects, because this would very likely far decrease the acceptance of the students and render the supportive systems useless. Students should be convinced with arguments to use the provided systems autonomously.

Version Control Systems. In both surveys all students voted for the use of version control systems (VCS). In the second survey the questions regarding the usage of VCS were more precise which showed that 96% of the participants even suggested the use of VCS for small projects with only one person like final theses.

Furthermore it was investigated which VCS the students preferred, which is shown in table 1. The acceptance of Subversion increased while there is a slight trend towards the more sophisticated Git.

Table 1. Survey: Preferred VCS

Type	First Survey	Second Survey
Subversion	56%	74%
CVS	23%	–
Git	19%	22%

Wikis. The students were also questioned, if they would recommend the use of wikis for the documentation of projects. The acceptance rate in the first survey was at 48%, in the second at 50%. In both cases the other half of the participants abstained from voting.

Again the second survey questioned deeper and the students were asked about their experiences with wikis in their previous project work. It could be seen that just the 50% who suggested the use of wikis had already worked with them in their projects, and that the other half which abstained had not gained any experiences with their use. This also illustrates how important it is that students get to know the different supportive systems, so that they can choose the right ones for their future projects.

Tracker. The use of issue-tracking systems like the open-source Mantis or Bugzilla, or integrated as parts of other systems like in Trac or Redmine [5], was suggested by 66% of the students in the first survey. The acceptance increased to 85% in the second survey.

Mailing-Lists. Automatically generated mailing-lists for projects were favoured by 72% in the first and 80% of the participants in the second survey.

Personal Homepages. Personal homepages are the only point in which the demand was decreasing between the two surveys, it sank from 53% to 30%, which shows the lower importance of this more traditional function.

Administration. The results of both surveys indicated (71% and 90%), that the students would prefer a system where they can independently choose and establish supportive systems for their projects – like creating Subversion repositories or wiki instances on demand.

Usage for Personal Projects. A growing part of the students (from 69% to 93%) would appreciate it if they could use supportive tools for their own small projects, for example using a VCS and a wiki to support them in their final thesis.

Social Functions. An Internet portal page, where all projects are listed, and thus which offers the possibility to find, look at and learn from projects of other students and which therefore increases the visibility and sustainability of their own projects alike, was favoured by 71% of the participants of the first and 80% of the second survey.

The surveys additionally not only investigated if the students would like to see the projects of others but also if they are willing to share the contents of their own projects with other students. The positive answers increased from 61% up to 82% in the second survey.

Similar high acceptance rates were reached by functions such as the following:

- The use of instant messengers to support team communication.
- The commenting of content of other projects and the receiving of comments for own projects in return.
- The creation of user-profiles with interests and abilities to find project-partners and help.
- The creation of a central knowledge base with solutions to frequently arising problems.

All in all the surveys show a trend towards more experiences of the students and a greater acceptance of and demand for supportive systems as can be seen in figure 1.

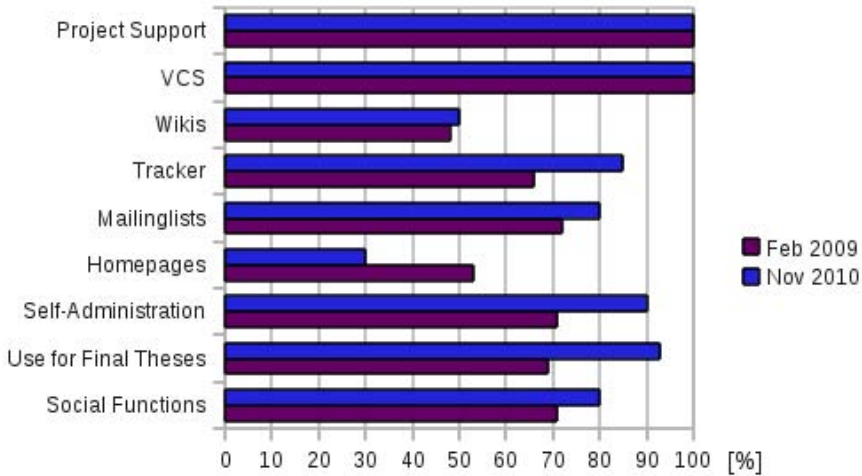


Fig. 1. Trends of the Surveys

Also an increasing willingness and wish to create a sustainable overvalue with their project work for other students and to profit from other students projects likewise was identified.

3.3 Further Observations

Other observations were made next to the surveys in the daily project support work. So figure 2 shows the number of projects which were supported with software systems (most of them with VCS, wikis and issue tracker, or integrated systems like Trac or Redmine). A slightly more than proportional increase can be investigated over the years.

Figure 3 shows the accumulated frequency of commits in the version control systems of all projects, which fluctuates depending on the current projects and project phases.

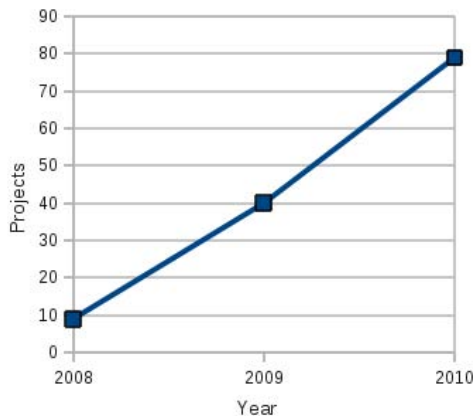


Fig. 2. Number of Projects

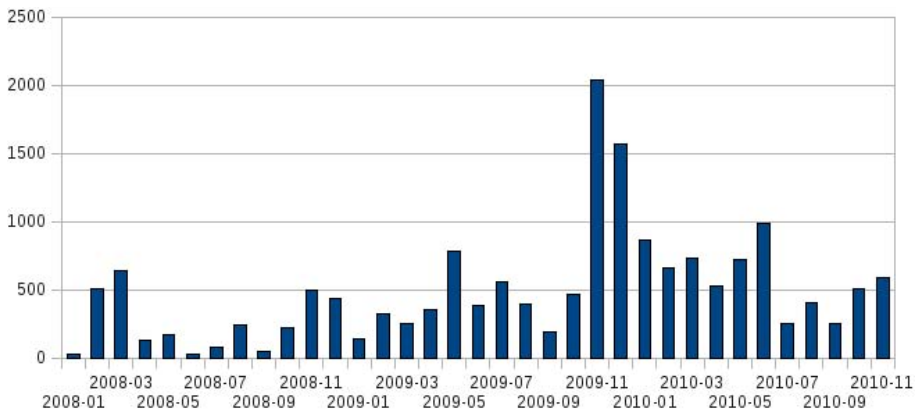


Fig. 3. Overview of Monthly Subversion Commits

It can be seen that project work takes place continuously, even in semester breaks, and that the systems are used extensively. A similar trend can be seen in the wiki usage.

In the last year there was a stronger demand for supportive systems which integrate multiple functions like VCS, wiki, tracker, shared file storage and more in one user-interface like Trac and Redmine. Also an increased interest in Git from students and staff members has been observed, which also showed slightly in the surveys.

4 Summary and Conclusions

All in all, there is a trend towards more sophisticated software support for projects. This can not only be proved by the increased number of participants in the second survey and their increased level of awareness of available tools and their handling. Also the higher acceptance of these tools by students and staff members in their daily project work demonstrates this trend.

The results of the survey point out how important it is to make the students familiar with the possibilities of recent supportive tools. They do not only have to be able to handle these tools efficiently but also, which may be even more important, have to have the knowledge and experience to decide what kind of tool is the best to use in a certain situation.

The survey also showed a trend towards tools such as Trac and Redmine for example. Both integrate multiple supportive functions in a single user-interface. The acceptance of Subversion is enhanced within the version control systems. Surprisingly, there was a slight increase in the interest in Git, which offers a far more sophisticated use as Subversion for example with different workflow models. Therefore, its utilization should be absolutely encouraged.

Also important and carrying a great potential is the higher acceptance of social functions in project support and the increased demand for a central portal page for projects.

If the projects can be presented there in an easy way to browse and search, their value will be increased beyond their development time because other students might be inspired. They could learn from conducted projects or even interact with currently carried out projects, giving comments, advice, or criticism. Therefore, through the higher visibility and possible higher benefit of a project for others, the students would be more encouraged to aim for high quality and reusability of their project artifacts, especially the documentation. In using these tools students are supported furthermore in this effort. The overall sustainability of both the project results as well as the process to achieve them is significantly increased.

Finally the results of the survey show, that supporting software systems for projects can expect the highest acceptance when the students have the freedom to choose them themselves and are not forced to use a predefined set of tools or to use them at all. This increased responsibility certainly requires that the students are aware of a profound knowledge of the different types of tools available. They need to know about their use, possibilities, strengths and weaknesses. Then they can choose the right tools for their work. This implies that these tools, their attributes and handling early have to be a part of the curriculum in courses and exercises, in particular because the students can profit from their usage already very soon in their studies.

5 Future Work

With these results the project support at our department shall be improved to better fit the requirements of the students and staff members. This includes giving them the facility to easily create supportive software systems like wikis, VCS, trackers and others by themselves using a web interface based on a central user-management. This also means building a portal for the projects and establishing functions of social software.

All this will cause great efforts to be accomplished. Therefore it is planned to further investigate how other institutes are satisfying this needs and to work together with them striving for solutions which will be generally applicable and an enrichment for all.

Additionally future investigation will be made to keep track of the always changing requirements and to gain a finer understanding which types of projects can benefit the most from which type of supportive systems. Finally an eye has to be kept open to look at other collaborative, supportive tools, new versions of used tools and newly developed ones.

Acknowledgments

This paper owes various persons a debt of gratitude, particularly the students and staff of the department of computer science at the University of Applied Sciences and Arts Hannover, who participated in the surveys, stated their requirements and wishes, evaluated and used supportive functions and gave valuable feedback. Especially Oliver Bruns and Noomi Müller helped to improve this paper. Also the department itself which invests into the challenging research and development of a preferably optimal support for its projects earns gratitude.

References

1. Denzer, A.: Long-Distance Multidisciplinary Collaboration: Some Lessons Learned. In: Proceedings of the National Capstone Design Conference (2010), <http://capstoneconf.org/resources/2010%20Proceedings/index.html>
2. Brereton, O.P., Lees, S., Bedson, R., Boldgreff, C., Drummond, S., Layzell, P., Macaulay, L., Young, R.: Student group working across universities: a case study in software engineering. In: IEEE Transactions on Education, p. 394. IEEE Education Society, Los Alamitos (2000)
3. Edgewall Software: Trac, <http://trac.edgewall.org/> (visited, December 2010)
4. Grudin, J., Poole, E.S.: Wikis at work: success factors and challenges for sustainability of enterprise Wikis. In: Proceedings of the 6th International Symposium on Wikis and Open Collaboration, WikiSym 2010, pp. 5:1–5:8. ACM, New York (2010), <http://doi.acm.org/10.1145/1832772.1832780>
5. Lang, J.-P.: Redmine, <http://www.redmine.org/projects/redmine> (visited, December 2010)
6. Kadenbach, D., Kleiner, C.: Benefits and Challenges of Using Collaborative Development Environments with Social Software in Higher Computer Science Education. In: Ozok, A.A., Zaphiris, P. (eds.) OCSC 2009. LNCS, vol. 5621, pp. 479–487. Springer, Heidelberg (2009)
7. Munson, S.A.: Motivating and enabling organizational memory with a workgroup wiki. In: Proceedings of the 4th International Symposium on Wikis, WikiSym 2008, pp. 18:1–18:5. ACM, New York (2008), <http://doi.acm.org/10.1145/1822258.1822283>
8. Keith Stanfill, R., Blackwelder, E.I.: Adapting Lightweight Source Control and Project Management Software for Use by Multidisciplinary Product Design Teams. In: Proceedings of the National Capstone Design Conference (2010), <http://capstoneconf.org/resources/2010%20Proceedings/index.html>
9. Radermacher, A., Helsene, A., Knudson, D.: Improving Capstone Courses with Content Management Systems and Virtualization. In: Proceedings of the National Capstone Design Conference (2010), <http://capstoneconf.org/resources/2010%20Proceedings/index.html>
10. Tabrizi, M.H., Collins, C.B., Kalamkar, V.: An international collaboration in software engineering. In: Proceedings of the 40th ACM Technical Symposium on Computer Science Education, SIGCSE 2009, pp. 306–310. ACM Press, New York (2009), <http://doi.acm.org/10.1145/1508865.1508976>
11. The Lime Survey Project Team: Limesurvey, <http://www.limesurvey.org> (visited December, 2010)
12. Tørning, K.: Position Paper: CSCW and the Web 2.0. In: Workshop of the Tenth European Conference on Computer-Supported Cooperative Work (2007)