

KommGame: A Reputation Environment for Teaching Open Source Software

Veerakishore Goduguluri, Terhi Kilamo, and Imed Hammouda

Department of Software Systems, Tampere University of Technology
Korkeakoulunkatu 1, FI-33720 Tampere, Finland
`{firstname.lastname}@tut.fi`

Abstract. The importance of teaching open source software in universities is increasing with the advent of open source as a development and business model. A novel, student centric approach of teaching open source was tried out at Tampere University of Technology where a new environment called KommGame was introduced to assist in teaching open source development. This environment includes a reputation system to motivate learners to participate. In this paper, we present our approach of teaching open source and how the KommGame environment was employed to teach open source software.

1 Introduction

With the advent of open source software (OSS) as a development and business model, the number of job vacancies valuing open source knowledge and experience has been rising on a regular basis. This in turn has motivated many universities and professional schools to introduce new courses and programmes related to teaching OSS principles and practices (e.g. [1], [3]). So far OSS teaching has mostly been organized in a traditional lecture course format, for example taking the form of a seminar where students present specific OSS related topics. Other attempts rely on sending students out into real open source projects and communities (e.g. [2]).

Such approaches to teaching open source software face two major challenges. First, classical teaching methods may not fully convey all the special aspects involved in OSS development such as community collaboration, peer review, and co-creation. Second, students may find it hard to participate in real OSS project as a first experience. This is because OSS projects typically have own principles, practices, processes, and tools.

A more attractive approach is to provide a learning environment for OSS where students could collaborate collectively to achieve a common goal. Such constructivist approach [4] to learning allows students to generate new knowledge through the interaction of the group's past experience and new ideas. A constructivist learning method however needs individual's active participation, which from the OSS perspective means student contribution to the community. An important

question is, therefore, how to keep students' motivation high for the purpose of learning OSS concepts through active contribution. It has been argued that reputation systems could play an important role in maintaining student motivation [5].

In this paper, we argue that reputation systems can be applied in a learning environment for open source software. Our approach is also inspired by the experiences of using reputation systems to reward and recognize developers in OSS communities such as Qt [6]. Towards this aim, we present an example reputation model and a concrete reputation environment known as KommGame that mimics real open source projects. The environment has successfully been tested at Tampere University of Technology (TUT) to introduce OSS concepts and practices to software engineering students.

The remaining of this paper is structured as follows: Section 2 reputation systems for teaching open source. Section 3 presents the KommGame environment. Finally we conclude in Section 4.

2 Reputation Model for Teaching Open Source Software

Reputation systems are used to measure the contribution of individuals in an online community; they are also applied in different fields such as e-commerce, search engines, and social news. As reputation systems are applied for measuring online activities one can see that reputation systems can be applied for e-learning in the educational context where most of the activities happen online. In [3] Farmer has explained about different reputation models. It is discussed in [7] that reputation systems suites a small group of young participants; they have high competitive spirit which makes learning more active and motivated.

In OSS development all kinds of contribution are treated as equally important and there is no good metric with which to compare or quantify different types of contribution with each other. This is the reason why most of the open source communities have not adopted a reputation system. In an educational context, however, the course moderator may decide which types of contribution should be emphasized. A reputation model can be designed accordingly.

We argue that the karma reputation model fits well the activities and the nature of OSS communities, where the object subjected to reputation is human. The final karma value of the participants is the sum of weight times of each contribution. The universal karma model can be written as

$$Karma = \sum_{k=1}^n (f_k(contribution_k)) + f(Favorites) + g(Weekly\ Quality\ Tokens) \quad (3.1)$$

Here n corresponds to the total number of contributions. f_k is the weight function corresponding to contribution type. "Favorites" is the number of like bookmarks a content author gets. "Weekly Quality Tokens" corresponds to the number of time the particular participant was selected as the best quality contributor of the week by the rest of the members of community.

For example, a sample karma model which covers activities related to bugs, features, improvements and wiki is given below. In the formula each activities is multiplied with its associated weight. Total karma is sum of all karmas from each activity.

$$\begin{aligned} \text{Karma} = & 6 * \sqrt{(\text{number of bugs reported})} + 3 * \sqrt{(\text{number of bug comments})} + \\ & 2 * (\text{number of bugs closed}) + 4 * \sqrt{(\text{number of feature requests})} + 3 * \sqrt{(\text{number of bug comments})} + \\ & 2 * (\text{number of closed new features}) + 4 * \sqrt{(\text{number of request})} + \\ & 3 * \sqrt{(\text{number of improvement comments})} + 2 * \sqrt{(\text{number of closed improvements})} + \\ & 4 * \sqrt{(\text{number of edits})} + 4 * \sqrt{(\text{number of likes})} + 4 * \sqrt{(\text{number of weekly quality tokens})} \end{aligned}$$

3 KommGame environment

We have developed an OSS learning environment based on the reputation model presented earlier. The learning environment, called KommGame [8], maintains karma values as a motivational factor for a community of learners. The KommGame environment forms an infrastructure required for collaborative and student centric learning.



Fig. 1. Karma reporting interface.

The KommGame infrastructure has been developed to mimic the infrastructure of a real open source community. The environment has features to add and edit open content, a user management system to manage users of the community, a system to track user activities, a communications channel, a bug management system, a source code base to maintain source code of the project, a reputation system to calculate the karma of each community member and an user interface to publish karma values.

Figure 1 shows the KommGame interface for karma value reports. The graph shows different categories of users (i.e. committers and reporters), illustrated using

different colors. Each vertical bar in the graph represents the score of each user of the system. Each vertical bar has two parts with different colors, the bottom part indicates the score of the previous weeks. The upper part indicates the score of the current week. The hat icons, shown in some bars, indicate the best contributors of the week.

4 Conclusions

The approach of KommGame for OSS education allows students to practice OSS project in safe and realistic OSS environment. The KommGame motivates the students to make more contribution to the OSS project and thus give them a valuable OSS project experience. This kind of realistic setting gives the students a good starting point to work in real OSS development.

The future plans for KommGame are to research how this can be applied in traditional programming courses where, students have to collaborate and participate in programming exercises. Future work includes applying the karma model to other courses and using KommGame as a standard system to issue certificates for OSS learners.

References

1. Megías, D., Serra, J., Macau, R.: An International Master Programme in Free Software in the European Higher Education Space. In *Proceedings of the First International Conference on Open Source Systems*, pages 349–352, July (2005)
2. Lundell, B., Persson, A., Lings, B.: Learning Through Practical Involvement in the OSS Ecosystem: Experiences from a Masters Assignment. In *Open Source Development, Adoption and Innovation*, volume 234 of *IFIP International Federation for Information Processing*, pages 289–294. Springer (2007)
3. German, M. D.: Experience teaching a graduate course in Open Source Software Engineering. In *Proceedings of the First International Conference on Open Source Systems (OSS 2005)*, Genova, Italy, pages. 326–328, July 11–15 (2005)
4. Piaget, J.: *The Child’s Conception of the World*. Rowman and Allenheld, New York (1960)
5. Temperini, M., Sterbini, A.: Learning from Peers: Motivating Students through Reputation Systems, *International Symposium on Applications and the Internet*, 305–308 (2008)
6. Qt developers network reputation system, <http://developer.qt.nokia.com/ranks>, last visited on March, 2011.
7. Temperini, M., Sterbini, A.: Learning from Peers: Motivating Students through Reputation Systems, *International Symposium on Applications and the Internet*, pages 305–308, (2008)
8. OSS Learning environment at TUT, http://osscourse.cs.tut.fi/mantis/login_page.php last visited on March, 2011