How to Use Open Source Software in Education

Judith Bishop
Microsoft Research
Redmond, WA, USA
jbishop@microsoft.com

Carlos Jensen
Oregon State University
Corvallis, OR, USA
jensenca@eecs.oregonstate.edu

Walt Scacchi
University of California, Irvine
Irvine, CA, USA
wscacchi@ics.uci.edu

Arfon Smith
GitHub Inc
San Francisco, CA, USA
arfon@github.com

Keywords

Open source software; industrial collaboration; games; FOSS; capstone projects

1. SUMMARY

There are many papers about projects that release their software as open source. Our panel looks at why open source is a good idea for education, and how it can best be leveraged. The panel represents industry and academia and its members have years of experience in the field. Specific questions that panelists will address, in addition to their position statements, are:

- What makes open source particularly attractive for educators and students?
- What are barriers to use?
- What areas of computer science and what levels of study are most appropriate for introducing open source?

Finally, this panel takes a step back and looks at the broad issues associated with using that software in education. We will mention the specific details of releasing, curating, licensing and using open source software. But most of all this panel wants to explore how, for educators, open source works. We believe it is because open source fosters an ecosystem of collaborators who can pool resources and expertise to build new collaborative ways of solving hard research projects. We look forward to the contributions from the audience and energetic debate.

NOTE: There are many definitions of Open source software (OSS) and Free/Open Source Software (FOSS). OSS emphasizes community development and sharing, but the software might not be free. FOSS ensures free code, but does not guarantee shared development.

2. JUDITH BISHOP (MODERATOR)

OSS is a powerful way of advancing software development. The free aspect of FSS has always been attractive to academics who wish to put diverse software in the hands of their students. Software companies such as Microsoft are now routinely putting research projects, as well as key software, in open source repositories for both community development and free use. An example can be seen at http://research.microsoft.com/opensource . Academia has long wanted access to industrial projects, but now that it is a reality, what exactly can be done and by whom? There are several options, notably:

- instructors can use the software as illustration in their classes [10]
- capstone projects can be based on OSS [13];
- software can be extended for research purposes [5]
- in hackathons, often in conjunction with industry [12].

It is expected that students will have difficulties in working with other people's code [13], but the learning benefits of getting acquainted with industrial strength software are deemed to be worth it. What we are now seeing is that the growing diversity of available OSS projects presents an even greater hurdle for educators who are faced with selecting suitable projects for integration into a course [5] [10].

With quality open source software being released by industry daily, educators face a real dichotomy between staying with what they know and moving onto the next new thing. In order to help with the choice, curated hubs have been become popular. Openhub.net has some categorization and reflects which projects are popular. OSSPAL.org and Microsoft Research's portal**Error! Bookmark not defined.** concentrate on classifying software into categories.

Another problem for educators is that raw dumps on repositories such as GitHub can be meaningless to students without some additional guidance and back-up. Even at the research level, software on its own is not enough for a successful transfer of ideas, or the establishment of an engagement. As discussed in [12], even Hackathons need careful planning. I shall pose some solutions to these problems.

3. CARLOS JENSEN

Free/Open Source Software (FOSS) is an increasingly important part of the computing eco-system. Teaching students how to participate in FOSS not only gives students meaningful and highly marketable skills [11], it presents novel opportunities for recruiting and motivating both traditional and non-traditional [3][4], and can have a lasting positive effect beyond the classroom. That said, figuring out how to successfully contribute to FOSS is a daunting challenge for students, and often fraught with dangers [1], especially for women [2]. New tools must often be used, and students and instructors must both learn to navigate new social and technical processes, large code-bases and deal with active and distributed developer communities [4]. How then can instructors help students navigate the joining process and successfully incorporate FOSS in the classroom, where student success and inclusion are top concerns? I will address these issues, as well as ethical and logistic hurdles to using and incorporating FOSS development in the classroom.

SIGCSE'16, March 2–5, 2016, Memphis, TN, USA.
ACM ISBN: 978-1-4503-3685-7/16/03.
DOI: http://dx.doi.org/10.1145/2839509.2844665

4. WALT SCACCHI

FOSS is now a widespread approach to the development of software systems by groups of loosely-coupled developers working in geographically and temporally distributed settings [9]. FOSS development projects are now responsible for major software systems that underlie the ongoing operation of World-Wide Web servers and browsers, modern search engines, social media, e-commerce, global financial services, military defense systems, and more including operating systems, programming language compilers and interpreters, database management systems, networking, computer graphics, interactive development environments, public source code repositories, and other technologies central to the theory and practice of Computer Science [7][9]. Computer games are software applications that now represent among the most complex and most comprehensive embodiment of CS knowledge to date. Computer games have also become a global software industry, with large shares of computer users, especially those in the 8-30+ year old demographic, frequent game players. My interest is in examining how, where, when, and by whom do these two worlds of software come together, and to what ends [6][8]? I will identify what lies at the intersection and union of these two worlds of software, and what such knowledge may offer to CS educators who are looking for new ways and means to engage and challenge the next generation of undergraduate CS students.

5. ARFON SMITH

Whether consuming, contributing or producing, open source software for many industries is the 'new normal'. It's therefore critical that our students are exposed early to the open source ecosystem and learn how to become active members of the community. I'll concentrate on outlining some strategies for open source newcomers wishing to either release their first open source software or contribute to an existing project. In addition, I'll devote some time to talk about a new tool GitHub has developed to reduce the barrier to entry for teachers wishing to use GitHub in a classroom setting including creating assignments, inviting groups of students to a course and keeping on top of their progress.

6. ABOUT THE PANELLISTS

Judith Bishop is Director of Computer Science at Microsoft Research. She is leader of the Open Source Software Initiative which brings together projects from Microsoft and Microsoft Research that are of interest to academics.

Carlos Jensen is associate professor at Oregon State University. He has studied the barriers for entry into Open Source, and finding ways we can get more people, including students, involved in Open Source.

Walt Scacchi is Senior Research Scientist, Research Faculty, Institute for Software Research, and Research Director, Institute for Virtual Environments and Computer Games. His research includes organizational studies of FOSS development and the acquisition of OSS (including governance, procurement, licensing, software engineering, deployment, operations and field support).

Arfon Smith is a Chief Scientist at GitHub Inc, the world's largest host of software on the planet. He supports researchers who are using the GitHub platform for capturing the process of scientific discovery and seeks to elevate the role of research software within academia at large.

7. REFERENCES

[1] King, S., Kuechler, V., Jensen, C.: Joining Free/Open Source Software Communities: An Analysis of Newbies' First Interactions on Project Mailing Lists. 44th Hawaii International Conference on System Sciences (2011).

[2] Kuechler, V., Gilbertson, C., Jensen, C.: Gender Differences in Early Free and Open Source Software Joining Process. Open Source Syst. Long-Term Sustain. 78–93 (2012).

[3] Morelli, R., Tucker, A., Danner, N., De Lanerolle, T.R., Ellis, H.J., Izmirli, O., Krizanc, D., Parker, G.: Revitalizing computing education through free and open source software for humanity. Commun. ACM. 52, 67–75 (2009).

[4] Morgan, B., Jensen, C: Lessons Learned from Teaching Open Source Software Development. OSS, 133-142 (2014)

[5] Nascimento, D.M.; Cox, K.; Almeida, T.; Sampaio, W.; Almeida Bittencourt, R.; Souza, R.; Chavez, C., Using Open Source Projects in software engineering education: A systematic mapping study, in *Frontiers in Education Conference, IEEE* , 1837-1843 (2013)

[6] Scacchi, W., Modding as an Open Source Approach to Extending Computer Game Systems, *Intern. J. Open Source Software and Processes*, 3(3), 36-47, (2011). Reprinted in S. Koch (Ed.), *Open Source Software Dynamics, Processes, and Applications*, 177-188, Information Science Reference, IGI Global.

[7] Scacchi, W. and Alspaugh, T.A. Understanding the Role of Licenses and Evolution in Open Architecture Software Ecosystems, *Journal of Systems and Software*, 85(7), 1479-1494 (2012)

[8] Scacchi, W, and Cooper, K.M. Research Challenges at the Intersection of Computer Games and Software Engineering, *Conf. Foundations of Digital Games* (2015)

[9] Scacchi, W, Crowston, K, Jensen, C, et al. Towards a Science of Free/Open Source Systems, FOSS Workshop Final Report, Prepared for the Computing Community Consortium, Fall. (2010)

[10] Therese Mary Smith, Robert McCartney, Swapna S. Gokhale, and Lisa C. Kaczmarczyk. Selecting open source software projects to teach software engineering. In *Proceedings of the 45th ACM technical symposium on Computer science education*, SIGCSE, 397-402. (2014)

[11] Trapasso, E., Vujanic, A.: Accenture Newsroom: Investment in Open Source Software Set to Rise, Accenture Survey Finds. (2010).

[12] Trainer, E.H., Chaihirunkarn, C., Kalyanasundaram, A., Herbsleb, J.D., Community Code Engagements: Summer of Code & Hackathons for Community Building in Scientific Software. *ACM Conference on Supporting Groupwork (GROUP)*, (2014).

[13] Jari Vanhanen, Timo O. A. Lehtinen, and Casper Lassenius. 2012. Teaching real-world software engineering through a capstone project course with industrial customers. In *Proceedings of the First International Workshop on Software Engineering Education Based on Real-World Experiences* EduRex, 29-32 (2012)