Proceedings of FLOSSEdu

FLOSS education and computational thinking workshop. Open Source Systems Conference 2016. Gothenburg, Sweden.

Proceedings of FLOSS education and computational thinking workshop

Gregorio Robles, Terhi Kilamo, Jesús Moreno-León

Preface

This is a collection of the papers presented at the FLOSSEdu Workshop. The workshop was co-located with OSS 2016 and was held in Gothenburg, Sweeden, on June 2, 2016.

The purpose of this workshop was to bring together free software experts and educators to discuss challenges that we face in the educational world at present and and that we will face in the future and how they can be undertaken from a FLOSS perspective. The presence of FLOSS in education has not stopped growing in the last years both in K-12 and higher education, a trend that has benefited from using FLOSS to teach computer science and other disciplines, but also for teaching FLOSS as part of the curriculum. An example that can illustrate this situation is the teaching of computational thinking skills through computer programming, which is one of the latest trends in education. This field has been globally addressed almost exclusively with FLOSS technologies, both by using FLOSS platforms and programming languages, such as Scratch or Alice, but also by including in the curriculum the social aspects of software development that characterize FLOSS movements, like sharing and contributing to the community.

The program contained 3 technical presentations discussing topics related to education, from various angles and perspectives. For the selection of the presentations, authors had to submit contributions that have been peer reviewed by at least two members of the programme committee.

We would like to thank the members of the organizing committee and the program committee for their effort. We also appreciate the contribution of the authors of papers submitted. We are also very thankful with the OSS 2016 organizers, and especially with the local support received during the workshop.

Madrid, July 2016.

Gregorio Robles, Terhi Kilamo, Jesús Moreno-León.

Organizing Committee

- Gregorio Robles, Universidad Rey Juan Carlos, Spain.
- Terhi Kilamo, Tampere University of Technology, Finland.
- Jesús Moreno-León, Programamos.es & Universidad Rey Juan Carlos, Spain.

Programme Committee

- Yasemin Allsop, University of Roehampton (United Kingdom)
- Alessandro Bogliolo, University of Urbino (Italy)
- Jordi Freixenet, University of Girona (Spain)
- Petri Ihantola, Tampere University of Technology (Finland)
- Oystein Imsen, Lær Kidsa Koding (Norway)
- Terhi Kilamo, Tampere University of Technology (Finland)
- Simon Marsden, University of Portsmouth (United Kingdom)
- Joek van Montfort, Scratchweb Foundation (Netherlands)
- Jesús Moreno, Programamos.es (Spain)
- Eduard Muntaner, Inventors for Change (Spain)
- Peter Parnes, Luleå University of Technology (Sweden)
- Gregorio Robles, Universidad Rey Juan Carlos (Spain)

Sponsorship

This workshop was sponsored in part by the Region of Madrid under project "eMadrid - Investigación y Desarrollo de tecnologías para el e-learning en la Comunidad de Madrid" (S2013/ICE-2715).

Contents

Adopting Open Source IT Certification in Higher Education: Lessons from the Field

Stephen Murphy

School of Computing and Digital Technology, Birmingham City University, Birmingham, UK.
stephen.murphy@bcu.ac.uk

Abstract. This paper suggests areas of good practice and considerations based upon the experience of embedding an open source information technology (IT) certification into a UK higher education program. Academically, open source is used as a vehicle for teaching general academic skills and values, but also as a collection of marketable skills. IT certification is used to further develop and signpost these skills to employers. This paper critically reviews literature in the fields of open source software in education and IT certification. A case study then discusses the methods used to embed such certification at Birmingham City University in the UK. Key barriers are reviewed along with a summary of lessons learned for the benefit of those considering similar actions.

Keywords: open source software · IT certification · higher education

1 Introduction

Open source software (OSS) as defined by the Open Source Initiative [1] offers advantages over proprietary counterparts in areas such as acquisition cost [2], independence from vendors [3] and in some cases, improved reliability and performance [4, 5]. Such attributes are driving the increased popularity of OSS [6–8] into a 'second wave' of adoption beyond its traditional user base of computer enthusiasts [9]. OSS technologies now underpin many Internet delivery platforms [10] and are paving the way for the rapid expansion [11] of public and private cloud provision via technologies such as OpenStack [12]. However, this growth is not without problems. There is evidence to suggest demand for skilled practitioners is outstripping supply [11, 13]; applying extra pressure on the educational system to produce suitably qualified graduates to address this need.

With this backdrop in mind, this paper critically reviews literature regarding Information Technology (IT) certification and the use of OSS in education, highlighting a gap where these intersect. It then delivers a case study discussing the main issues surrounding the adoption of the Linux Professional Institute (LPI) Linux Essentials (LE) [14] certification as a valued-added extension to an academic

curriculum at Birmingham City University (BCU) in the UK. It will then go on to identify lessons learned from this experience and areas where work is still needed.

2 Literature Review

2.1 The Role of Certification in Higher Education

Higher Education (HE) worldwide is coming under greater scrutiny than ever before [15–19]. Taking the UK as an example, there have been concerns [17] over the effectiveness of degrees generally [20] and the employment prospects for computing graduates in particular [21]. This has led to a focus on general employability and industry-relevant practical skills [22]. One method to signpost programs containing such skills is professional body accreditation through institutions such as the British Computer Society or the Institution of Engineering and Technology. However, such bodies may have limited international currency and while they are a broad assurance of overall 'quality' [23], they do not signpost any specific skill to an employer. Additionally, it has also been argued that such accreditation schemes can stifle innovative program design and have limited currency with students [24].

While accreditation can signpost employable skills, it cannot foster them. Industry placements are an effective approach to developing such skills [25], but the number of computing students taking up such opportunities has declined [26]. Suggested reasons include students not appreciating the benefits and practical issues relating to obtaining and attending placement employment [27]. As this decline limits the utility of placements, parallel methods need to be sought [28], one such being IT certification (henceforth certification) [29].

Certification comes in two broad forms, vendor-specific and vendor-neutral [30]. Vendor-specific certifications are developed and managed by the provider of that technology (e.g. Microsoft and Cisco). Vendor-neutral certifications are commonly offered by independent third parties, with content that is generic to many providers' technologies (e.g. CompTIA, Linux Professional Institute). Both types of certification are valued when hiring IT practitioners [13, 31, 32], so much so that they may be a prerequisite for a position [22]. When not mandated, they may be used to differentiate [33] or quickly short-list suitable candidates [22]. Anecdotal evidence from BCU and elsewhere [32] suggests that even where certification is not officially sought by an employer, it can act a point of positive discussion during the selection process.

2.2 Challenges to Embedding Certification

While there appear to be employability benefits to the inclusion of certification within academic programs [34], few institutions have done this. Some do not consider it the job of HE [35], or consider it beneath the academic rigor of university-level study [34]. This poor perception of certification amongst academics

in particular may partly be explained by their own lack of understanding, with few holding these qualifications [36]. This negative view contrasts with disciplines such as law and accountancy where external certification is widely accepted [37]. In fact, there is evidence to suggest that HE is the correct place to embed certification. HE provides the theoretical underpinnings to scaffold the practical skills fostered by certification [38], leading to a greater depth of understanding [39] than either type of study alone.

Benefits notwithstanding, the content of some certifications has come under scrutiny. Some vendor-specific exams are narrow and product focused [38], and in some cases have been likened to propaganda [32]; a poor match for academic environments that promote critical thinking. Other exams may rely heavily on the reproduction of facts [40], leading to concerns that candidates may not know how to apply these skills in practice (so called 'paper certifications' [41]). This narrowness and focus on minutiae may contribute to the limited ‘shelf life’ of the skills taught [20].

As with the adoption of any innovation [42], the process of integrating certification can be challenging [32]. Academics may need to gain the certification themselves, a time-consuming and costly exercise [43]. This is commonly followed by a cycle of re-certification in the following years [39] to maintain qualified status. Changes may be required to the schedule of academic delivery, content, teaching style [29] and possibly even teaching staff. Where substantial commitment is not present, adoption may be reversed [29], leading to dissatisfaction amongst staff and students.

Given the rapid pace of technological change, educational programs delivering certification can date rapidly. The differing pace of change for academic and certification curricula can lead to conflicts [29], and may place additional pressures on those delivering synergized programs to be pro-active in adopting new technologies. Vendor-neutral alternatives have some advantages in this area [30], especially where the certified technologies are relatively stable. Linux may be one such example. Commands familiar to a UNIX administrator of 30 years ago can still be used on the latest distributions of Linux, bringing an element of continuity. When changes do occur, they are usually evolutionary, not revolutionary. This promotes stability in certification content, making them easier to integrate with academic programs.

2.3 Open Source in Education

OSS is increasingly used for teaching and learning [44]; both for pedagogic and infrastructure purposes. Often, as in industry, OSS is a pragmatic choice. The absence of a license fee permits easy trial [45], experimentation and deployment without a lengthy administrative overhead. This ease of ‘trialability’ is positively associated with the adoption of many technologies [42]. Pedagogically, OSS is commonly associated with software engineering [46], where such tools are frequently used in industry. There is also preliminary evidence to suggest that

participation in OSS development communities can foster employable skills such as communication and team working in a way that complements a formal education [47]. Participation in OSS projects has also been cited as a method to give students realistic challenges within their studies [48] or even as a 'virtual placement' in lieu of formal work experience [49].

There are numerous degree programs now teaching OSS principles and practices [50], and some have implemented 'self-certification' of OSS skills on an institutional basis [46]. However, unlike third party certification, this could suffer from limited currency in the job marketplace [32]. The author can find no work where OSS-specific skills have been verified via third-party certification, perhaps partially explaining the apparent lack of skilled practitioners in the eyes of some employers. To partly fill this gap, the following case study outlines how OSS vendor-neutral certification was integrated into a UK degree program and the lessons learned from that process.

3 Case Study

Two academic modules (discrete study elements) are the basis for this case study at BCU in the UK. Both modules supply similar outcomes on different computing degree programs, aiming to teach theoretical and practical skills relating to systems administration, computer networking, OSS concepts and transferable problem solving skills. The modules reference other subjects of study to allow synthesis of concepts and deep learning [51].

Fig 1. Module delivery pattern

Figure 1 shows the pattern of delivery utilized in the 2014/15 academic year. The academic portion of the modules are delivered via a one-hour slot in a traditional lecture theater and two hours in a computer lab, every week, for 12 weeks. The lecture session makes use of audience participation and flipped learning principles [52]. Students investigate OSS culture and projects outside of class and feed-back to the group during the lecture slot. The lab sessions develop practical skills by utilizing Linux virtual machines on an OSS virtualization platform, allowing students to easily re-create the lab environment on their own computers. Students are encouraged to work in small groups and peer-teach. Model solutions are provided in written and video format to allow self [53] and peer-assessment.

For practical administrative reasons, as well as the academic reasons previously discussed, the certification exam is not used for academic assessment. Summative assessment for the modules utilizes a combined theory and problem solving practical exam, delivered using the University's virtual learning environment (VLE). An

evaluative coursework exercise has also been used in the past in one of the modules to enhance the critical analysis skills of students.

The module content aligns to the LPI LE [14] certification program. This is a vendor-neutral, entry-level certification covering Linux system administration and general OSS skills. The certified material forms a subset of the material taught on the academic modules, with certification used as external verification of OSS skills. To facilitate the process of certification, students optionally attend a certification 'bootcamp' that follows on from the academic element of the module. The bootcamp program revises key exam topics and focuses on the further developing the skills needed to pass the certification exam. A variety of methods are used to prepare students for certification [54], but emphasis is placed on practical exercises and formative feedback via the use of practice questions. This means the face-to-face session are used as troubleshooting surgeries, rather than opportunities to 'deliver' content. In the penultimate week, students attempt a mock exam with the final week being the certification exam itself, facilitated by the University's own LPI test center.

3.1 Discussion of Key Adoption Challenges

Choosing a Certification – Philosophy and Practicality

Choosing the right certification to adopt is a critical decision. Factors such as currency in the employment market need to be considered, but one of the most important is the match between the philosophy of the certification and the academic program. It is rare that curricula are designed from scratch, so there needs to be some synergy between the existing academic approach and the certification; or a willingness to make this the case. Unsuitable certification cannot be successfully 'bolted on' to an academic program.

At BCU, a number of 'big-name' Linux certifications were investigated, but all were found to be too expensive to run or prescriptive in their approach at that time. They also offered a 'vendor's eye view' of the world, which was at odds with the focus of the academic curricula. At the time, the only vendor-neutral solution offering progression beyond one certification was the LPI program, and this is the scheme that was adopted. Two important aspects of the LPI offering are that their certifications are developed using open source principles and that a new certification was being designed with the academic environment in mind (Linux Essentials). The latter point was important, as this led to lower exam costs and aided the University's facilitation of the certification process. The LPI also had a pragmatic approach to what preparation resources could be used and had clear certification life cycles [55], ensuring flexibility of academic delivery could be maintained.

Content Alignment and Difficult Choices

Students are busier than ever, with the pressure on their finances leading many to work long hours to support themselves through their studies [56]. With this in mind, additional work, over and above their compulsory academic studies may be viewed

with skepticism and will need to be visibly advantageous. This makes aligning the academic and certification content one of the main objectives to foster participation and success.

For the example presented here, the content of the modules is well-aligned to that of the LE certification, but there are differences. Academic degrees should provide high-level cognitive skills and it is felt that they should be distinct from training courses for specific technologies [39]. One of the major issues with offering certification in academic programs is also one of its major benefits, namely the detailed knowledge required to pass a certification exam. Much of this knowledge may be at an academically low level on Bloom's taxonomy [57], making it hard to justify the use of such exams as part of a HE academic assessment process. Because of this, the decision was taken to separate the certification process from the academic assessment completely. While this may be detrimental to promoting participation in certification, it does resolve some concerns regarding both the rigor [58] and security [59] of certification exams.

The Right Support at the Right Time

When BCU initially trialed the LE exam in 2011/12, no structured support was offered for certification. Students who sat the exam found it challenging, something identified by others who have embedded certification [54]. This may be due to the disparity between the format and expectations of such exams and academic assessments. Whatever the reason, it was clear that additional support would be needed.

In industry, certification training is traditionally delivered via short, highly focused courses [43], sometimes called bootcamps. While the students already had much of the required skills and knowledge needed, the idea of using a bootcamp was compelling. However, deciding the timing and duration of this bootcamp proved challenging, and various formats were considered. One model that was quickly rejected due to resourcing and educational veracity concerns was the high intensity one week 'burst'. Between the initial bootcamp in 2012/13 and the latest in 2014/15, both the time within the academic calendar and duration have been adjusted.

Duration proved the easiest problem to solve. During the first iteration in 2012/13, a three-session bootcamp was offered, but students indicated that they would like additional support. A further two weeks were added, and this format has been used since then. The positioning within the academic calendar proved more problematic. Term-time offerings were initially popular, but led to large drop-outs due to conflicts with academic assessment schedules. The availability of both staff and labs was also limited due to academic commitments. Bootcamps in the summer period seemed ideal, but some students reported issues with the availability of accommodation and conflicts with family and work commitments. After much debate, the summer option was retained, as participation in the certification exam seemed to be higher at this time of year.

Participation

While the idea of certification appeals to students, participation in the final exam is an area where improvement is needed. Those that engage with the post-course certification are generally successful, with 95% of those who attempted the exam achieving certification in 2015. However, the numbers attempting the exam are limited. In 2015, around 190 students were eligible to attend the bootcamp, but only around 11% sat the certification exam. While this may seem low, this compares favorably with published participation rates [54], suggesting this is a generic rather than local issue.

One major barrier to participation is the cost of sitting a certification exam [60]. Certification is not currently considered a core part of the students' learning at the university, so is not covered in their tuition fees. One of the factors leading to the selection of the LE certification was its low cost, but any cost to the student is still a barrier to participation. This has been partially ameliorated by students competing for funded exam vouchers, but these are limited in number. New funding models and incentives are being investigated with a view to increasing the numbers of potential participants.

3.2 Summary of Lessons Learned

The literature shows that some attempts to embed certification have been unsuccessful [29] due to practical issues and philosophical objections. Some of these have been discussed in detail above, but others are also worthy of consideration and are summarized below:

1. Certification is challenging, so commitment is needed from the institution and staff to maximize the chance of student success. Half-hearted commitment will lead to dissatisfied staff and students. Those teaching and advising students need to have a good understanding of the benefits and practices of certification and ideally need to hold the qualification themselves. This will require a program of staff and resource development prior to offering certification to students.
2. Ensure that the academic and certification syllabuses are well-aligned. The greater this disparity, the harder it will be for students to succeed.
3. ‘Sell’ the benefits of participation to students. Able students may see the benefits for themselves, but others may need convincing. Consider using alumni to assist in this activity.
4. Check that the chosen certification provider does not mandate or restrict what resources must be used during preparation. Many vendor and third-party materials utilize a didactic approach to content delivery, which students familiar with a more participatory approach, may find unappealing. The ability to customize delivery and select appropriate resources is therefore essential.
5. Preparation for certification is not just a class based activity, personal study

is a necessity. Books, software, labs and practice questions need to be accessible as and when needed by students.

6. Offer coaching in certification exam techniques, as well as content. This may form part of the academic program itself, or be add-on support, such as a bootcamp. Many students will not be familiar with the customs and practices of certification and will need help to perform at their best.
7. Time delivery of support such that the students can focus their attention on it sufficiently. While there is no perfect time, some times are better than others and this will need to be identified on a local basis.
8. Look at factors surrounding the academic environment such as availability of student accommodation outside term time, student workloads and resource availability. These are issues that can impinge upon participation.
9. If certification is optional, offer incentives to participate (free exam vouchers, prizes, etc.). While the benefits of participation should be enough by themselves, pressures on modern students mean they may not be sufficiently tempting on their own.

4 Conclusions and Future Work

Despite being an imperfect solution, certification can form part of a valuable employability package for HE students. It offers the dual benefit of fostering and signposting the skills needed by employers, making it a useful method of embedding employable skills into an academic program. This paper has outlined some of the methods by which BCU has successfully embedded and aligned the LE certification as part of its students' learning journey.

In this institution (as elsewhere [29]), the separation of certification from the academic element via means of a bootcamp has been found to aid in the certification process. This has helped to clarify student priorities and ensure that academic rigor and high-level skills are maintained. However, the alignment between academic and certification syllabuses is essential to ensure students can achieve certification success with a modest outlay of additional effort. As elsewhere [61], factors such as the timing of support, staff commitment and qualifications as well as the academic and social environment have been found to be significant in ensuring successful outcomes. However, challenges remain surrounding certification costs and participation rates, underlining that they should form part of a suite of employability measures, and not be the only solution.

To tackle some of these issues, we encourage further work to understand and mitigate barriers to student participation in certification. It would also prove instructive to have a better picture of whether it is the content of the certification itself that appeals to employers or the fact that the student has extended their studies beyond the required minimum, and whether there are generic factors that differentiate certifications in the eyes of an employer.

5 References

1. Open Source Initiative: The Open Source Definition (Annotated), https://opensource.org/osd-annotated.
2. Williams van Rooij, S.: Adopting Open-Source Software Applications in U.S. Higher Education: A Cross-Disciplinary Review of the Literature. Rev. Educ. Res. 79, 682–701 (2009).
3. Kuechler, V., Jensen, C., Bryant, D.: Misconceptions and Barriers to Adoption of FOSS in the US Energy Industry. In: Petrinja, E., Succi, G., El Ioini, N., and Sillitti, A. (eds.) Open Source Software: Quality Verification. pp. 232–244. Springer, Berlin (2013).
4. Raymond, E.S.: The Cathedral and the Bazaar. First Monday. 3, (1998).
5. Boulanger, A.: Open-Source Versus Proprietary Software: Is One More Reliable and Secure Than the Other? IBM Syst. J. 44, 239–248 (2005).
6. Gartner: User Survey Analysis: Open-Source Software, Worldwide. Gartner, Stamford (2008).
7. Gartner: Gartner Survey Reveals More than Half of Respondents Have Adopted Open-Source Software Solutions as Part of IT Strategy, http://www.gartner.com/newsroom/id/1541414.
8. Black Duck Software: The Ninth Annual Future of Open Source Survey, https://www.blackducksoftware.com/future-of-open-source.
9. Choi, N., Chengalur-Smith, I.: An Exploratory Study on the Two New Trends in Open Source Software: End-Users and Service. In: Sprague Jr, R.H. (ed.) 42nd Hawaii International Conference on System Sciences, 2009. HICSS'09. pp. 1–10. IEEE (2009).
10. Netcraft: November 2014 Web Server Survey, http://news.netcraft.com/archives/2014/11/19/november-2014-web-server-survey.html.
11. Rightscale: State of the Cloud Report. Rightscale, Santa Barbara (2016).
12. Openstack: User Stories » OpenStack Open Source Cloud Computing Software, http://www.openstack.org/user-stories/.
13. Linux Foundation: Linux Jobs Report 2015. Linux Foundation, San Francisco (2015).
14. Linux Professional Institute: Linux Essentials - Linux Professional Institute, https://www.lpi.org/certification/linux-essentials/.
15. The Economist: Not What It Used to Be, http://www.economist.com/news/united-states/21567373-american-universities-represent-declining-value-money-their-students-not-what-it, (2012).
16. HM Government: Teaching at the Heart of the System - Speeches - Gov.uk. , Universities UK, Woburn House, Tavistock Square, London (2015).
17. Dearing, R.: Dearing Report 1997 - Full Text. HM Government, London (1997).
18. Douglas, J.A., Douglas, A., McClelland, R.J., Davies, J.: Understanding Student Satisfaction and Dissatisfaction: An Interpretive Study in the UK Higher Education Context. Stud. High. Educ. 40, 329–349 (2015).

19. Higher Education Policy Institute, Higher Education Academy: The HEPI-HEA 2015 Student Academic Experience Survey. Higher Education Policy Institute and Higher Education Academy, York (2015).
20. Archer, W., Davison, J.: Graduate Employability. The Council for Industry and Higher Education, London (2008).
21. Hetrick, S.: What's Wrong with Computer Scientists?, http://www.software.ac.uk/blog/2013-10-31-whats-wrong-computer-scientists.
22. Robin, G.J.: Do Companies Look for Education, Certifications or Experience: A Quantitative Analysis. In: Proceedings of the 49th SIGMIS Annual Conference on Computer Personnel Research. pp. 1–5. ACM, San Antonio (2011).
23. Reichgelt, H., Yaverbaum, G.: Accountability and Accreditation: Putting Information Systems Accreditation into Perspective. Commun. Assoc. Inf. Syst. 20, 27 (2007).
24. Sommerville, I.: BCS Accreditation Considered Harmful, http://iansommerville.com/systems-software-and-technology/bcs-accreditation-of-computer-science-degree-courses/.
25. Patel, N., Brinkman, W.-P., Coughlan, J.: Work Placements and Academic Achievement: Undergraduate Computing Students. Educ. Train. 54, 523–533 (2012).
26. Education for Engineering: Sandwich Courses in Higher Education - a Report on Current Provision and Analysis of Barriers to Increasing Participation. Education for Engineering, London (2011).
27. Banga, K., Lancaster, T.: Addressing the Challenges Computing Students Face in Completing a Placement Year. Presented at the HEA STEM Conference , Birmingham (2013).
28. Hepworth, S., Beaumont, C., Halligan, D., Allanson, M.: Strategic Enhancement Programmes Embedding Employability into the Curriculum: An Evaluation of Practice in 3 Departments at Edge Hill University. Higher Education Academy, York (2015).
29. Rob, M.A.: IT Certification: Demand, Characteristics and Integration into Traditional University MIS Curriculum. Commun. IIMA. 14, 2 (2014).
30. Bird, D.: Certified Paths to Success, http://www.informit.com/articles/article.aspx?p=23281.
31. Wireschen, D., Zhang, G.: Information Technology Certification Value: An Initial Response from Employers. J. Int. Technol. Inf. Manag. 19, 89–109 (2010).
32. Koziniec, T., Dixon, M.W.: ICT Industry Certification: Integration Issues for Post-Secondary Educational Institutions in Australia. In: Informing Science. pp. 831–838. The USENIX Association, Cork (2002).
33. Prabhakar, B., Litecky, C.R., Arnett, K.: IT Skills in a Tough Job Market. Commun. ACM. 48, 91–94 (2005).
34. Schlichting, C., Mason, J.: Certification Training and the Academy. J. Comput. Sci. Coll. 20, 157–167 (2004).

35. Schlichting, C., Mason, J.: The Computer Curriculum and Certification: A Proposal. J. Comput. Sci. Coll. 20, 84–91 (2005).
36. Coelho, J.V.P.: Competence Certification as a Driver for Professional Development: An IT-Related Case-Study. In: EDUCON 2010 – Annual Global Engineering Education Conference The Future of Global Learning in Engineering Education. pp. 81–88. EDUCON, Madrid (2010).
37. Ray, C., M., McCoy, R.: Why Certification in Information Systems? Inf. Technol. Learn. Perform. 18, 1–4 (2000).
38. Randall, M.H., Zirkle, C.J.: Information Technology Student-Based Certification in Formal Education Settings: Who Benefits and What Is Needed. J. Inf. Technol. Educ. 4, 287–306 (2005).
39. Ortiz, A.: Preparing Undergraduate Students for Java Certification. In: Companion of the 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications. pp. 178–183. ACM, Anaheim (2003).
40. French, M.: Why IT Certification Matters, http://www.techcentral.co.za/why-it-certification-matters/13743/.
41. Zadik, J.M., Dittman, K.: Employers Beware: Degrees and Certifications Don't Guarantee the Quality of an Information Technology Applicant. Rev. Bus. Inf. Syst. RBIS. 13, (2011).
42. Rogers, E.M.: Diffusion of Innovations. Free Press, New York (2003).
43. Frank, C.E., Werner, L.: The Benefit of the CSSLP Certification for Educators and Professionals. J. Comput. Sci. Coll. 26, 49–55 (2010).
44. Williams van Rooij, S.: Higher Education Sub-Cultures and Open Source Adoption. Comput. Educ. 57, 1171–1183 (2011).
45. Fitzgerald, B.: Open Source Software Adoption: Anatomy of Success and Failure. In: Koch, S. (ed.) Multi-Disciplinary Advancement in Open Source Software and Processes. pp. 1–23. IGI Global, Hershey (2011).
46. Papadopoulos, P.M., Stamelos, I.G., Cerone, A.: Using Open Source Projects in Higher Education: A Two-Way Certification Framework. In: Information Technology and Open Source: Applications for Education, Innovation, and Sustainability. pp. 274–280. Springer, Berlin, Heidelberg (2012).
47. Fernandes, S., Cerone, A., Barbosa, L.S.: A Preliminary Analysis of Learning Awareness in FLOSS Projects. In: Information Technology and Open Source: Applications for Education, Innovation, and Sustainability. pp. 133–139. Springer, Berlin, Heidelberg (2012).
48. 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, 2013 IEEE. pp. 1837–1843. IEEE (2013).
49. García-Peñalvo, F.J., Cruz-Benito, J., Conde, M.Á., Griffiths, D.: Virtual Placements for Informatics Students in Open Source Business Across Europe. In: 2014 IEEE Frontiers in Education Conference (FIE) Proceedings. pp. 1–5. IEEE, Madrid (2014).

50. Montes León, S.R., Robles, G., González-Barahona, J.M., Sánchez C., L.E.: Considerations Regarding the Creation of a Post-graduate Master's Degree in Free Software. In: Corral, L., Sillitti, A., Succi, G., Vlasenko, J., and Wasserman, A.I. (eds.) Open Source Software: Mobile Open Source Technologies. pp. 123–132. Springer, Berlin, Heidelberg (2014).
51. Marton, F., Säljö, R.: On Qualitative Differences in Learning: I - Outcome and Process. Br. J. Educ. Psychol. 46, 4–11 (1976).
52. Flipped Learning Network: Definition of Flipped Learning, http://www.flippedlearning.org/domain/46.
53. Huxham, M.: Fast and Effective Feedback: Are Model Answers the Answer? Assess. Eval. High. Educ. 32, 601–611 (2007).
54. Hopkins, C.W., Pickard, J., Patrick, A.: Getting Students Certified: A Study of Certification Pass Rates in Information Technology Degree Programs. In: American Society for Engineering Education Gulf-Western Conference. p. 1 (2014).
55. Wyrostek, W.E.: The Top 10 Problems with IT Certification in 2008, http://www.ciscopress.com/articles/article.asp?p=1180991.
56. National Union of Students: The Pound in Your Pocket Summary Report. National Union of Students, London (2012).
57. Bloom, B.S., Krathwohl, D.R., Masia, B.B.: Taxonomy of Educational Objectives: The Classification of Educational Goals. Longman, New York (1984).
58. Jovanovic, R., Bentley, J., Stein, A., Nikakis, C.: Implementing Industry Certification in an IS Curriculum: An Australian Experience. Director. 7 (2006).
59. Musthaler, L.: Cheaters: Inside IT Certification Fraud | PCWorld, http://www.pcworld.com/article/150638/certification_fraud.html.
60. Tate, N.J., Lichtenstein, S., Warren, M.J.: IT Security Certifications: Stakeholder Evaluation and Selection. In: ACIS 2008 Proceedings. pp. 991–1001. ACIS, Christchurch, New Zealand (2008).
61. Saleem, N., Gercek, G., Varma, M.M.: Integrating IT Professional Certifications into Graduate CS/MIS Programs: a Blueprint for Success.

How social are Scratch learners? A comprehensive analysis of the Scratch platform for social interactions

Jesús Moreno-León[1], Gregorio Robles[2], and Marcos Román-González[3]

[1] Programamos.es, Spain, jesus.moreno@programamos.es
[2] Universidad Rey Juan Carlos, Spain, grex@gsyc.urjc.es
[3] Universidad Nacional de Educación a Distancia, Spain, mroman@edu.uned.es

Abstract. A major difference between Scratch and other visual programming languages aimed at learning computational thinking skills is that it offers a web platform where learners can interact by sharing with and contributing to the community. Such a platform allows them to learn many of the social aspects of software development that characterize the FLOSS movement. However, most of the investigations that analyze social aspects of the Scratch community have been based on a small scale. This work presents a comprehensive approach, as we have analyzed data from the platform, from 2007 to 2012, to measure how social the Scratch community is by studying user contributions in terms of number of comments, friends, favorites and galleries. The analysis involves over one million learners with almost two million projects. The results indicate that the vast majority of users barely make use of the social capabilities offered by the Scratch platform.

Key words: Computational thinking; Scratch; social software development; social learning; FLOSS;

1 Introduction

Scratch [12] is a visual programming language specifically designed for young learners that is being massively used worldwide; the stats in the Scratch website[1] indicate that there are more than ten million Scratch users. One of the main differences with other, similar visual programming languages, such as Alice [2], Snap! [7] or Kodu [8], is that Scratch offers a web platform with social features to share, study and remix projects, post comments or work in teams. Thus, Scratch programmers have the opportunity to learn the social aspects of software development that characterize FLOSS (Free/Libre Open Source Software) movements, like sharing and contributing to the community.

However, most of the research papers that highlight the advantages for learners of participating in the Scratch community are based on small scale studies.

[1] http://scratch.mit.edu/statistics

The Scratch *team* states that "Participation and collaboration within online communities can support, inspire, and enable young people to become active creators (and not just consumers) of interactive media." [1], but this statement is based on three concrete cases that illustrate different types of collaboration that allowed more complex and elaborated projects. In a similar vein, the Scratch *team* argues that *scratchers* "not only learn important math and computer science concepts, but they also develop important learning skills: creative thinking, effective communication, critical analysis, systematic experimentation, iterative design, and continual learning" [9], based on five cases in which user collaboration fostered the improvement of projects.

This work is the first step of a series of investigations with a different approach, as we are trying to identify the impact of social participation on software development skills by analyzing, not a few cases, but performing a comprehensive study of the whole community. Specifically, the research question that we try to answer in this paper is following:

RQ: How *social* is the Scratch community in terms of number of comments, friends, favorites and galleries?

To give an answer to this question we have analyzed a dataset that includes the first five years, approximately from 2007 to 2012, of public data from the Scratch online community website.

2 Background

Several authors have previously investigated patterns of participation in the Scratch community from different points of view and with distinct objectives.

The contributions, mechanisms of gratification and patterns of participation of 65 young students (between 9 and 17 years old) learning to program with Scratch are studied in [15], finding no differences based on gender.

100 randomly selected projects, along with their associated comments, were analyzed in [3] to evaluate the skills of the developers in terms of collaboration and code reuse. In spite of the important number of generated comments, researchers could not find a single case in which the feedback originated some kind of online collaboration. In addition, the level of code reuse was around 5%, and only 40% of the remixes implicated modifications in the code, being the majority of changes related to modification or inclusion of images and sounds.

Aiming to assess the evolution of Scratch users, both from technical and social points of view, in a subsequent investigation the same researchers selected data from 250 random users that had programmed in total around 1,000 projects [13]. The findings of this paper are unexpected, as even though a positive progression in terms of social skills is detected, a negative progression in the technical abilities is also observed.

Attempts to analyze bigger amounts of data from the Scratch repository can also be found in the literature. The activity of 5,000 users during three months is analyzed in [5]. In general, no correlation between online participation

and project sophistication was detected. However, authors highlight that a very small and extremely active group of users utilized more complex programming instructions than the rest. With the goal of presenting implications in the design of collaborative tools and communities [6], the activity in the community was analyzed to identify patterns of participation. In [14] the comments generated by 5,000 users in a month was studied, stating that comments about projects show a richer language than other kind of comments.

Finally, an investigation with 2.4 million `Scratch` projects, which analyzes the same dataset that we have studied for our research, shows that users who remix more often utilize a larger set of programming commands and are more likely to use computational thinking concepts from the projects they remix [4].

3 Data

For this investigation we have worked with a collection of datasets that includes the first five years, approximately from 2007 to 2012, of public data from the `Scratch` online community website[2].

The set of datasets are divided into *Core datasets*, which describe the major objects and relationships captured by the `Scratch` website, such as users, projects, galleries or favorites; *Text and Code datasets*, which contain user submitted text; and *Project Analytics Datasets*, such as the *project blocks* table, which holds the blocks used in each project.

To answer the research question of this paper we have analyzed the data included in the *Core datasets*. Specifically, we have worked with the information on 1,056,951 users, 1,928,699 projects, 120,097 galleries, 1,313,200 friends, 1,041,387 favorites and 7,788,414 project comments that are available in the datasets.

4 Results

Although there is more than 1 million active users in the dataset, only 304,793 had published at least one project, so we selected this subset to study their social behavior.

Figure 1 shows the distribution of users in terms of time in the community (measured in days between the first and last published projects) and the number of published projects. As can be seen, both are extremely skewed distributions, with a vast majority of users with a small amount of projects and a short time in the community. However, there exists an extremely small set of users who publishes lots of projects and spends a long time in the `Scratch` community.

As shown in Figure 2, the same phenomenon applies when social activities of users are studied. Most of the users barely make use of the social features

[2] Access to the datasets can be requested at https://llk.media.mit.edu/scratch-data/

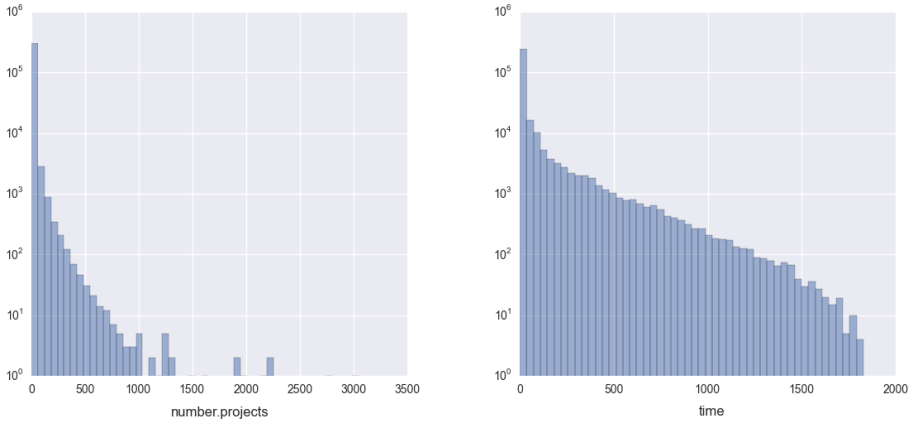

Fig. 1. Distribution of users in terms of number of published projects and time (days) in the community. The vertical axis is in logarithmic scale.

of the `Scratch` platform, while a tiny number of users seem to be remarkably social, as they create galleries, post comments, make friends and mark projects as favorites on a highly active basis.

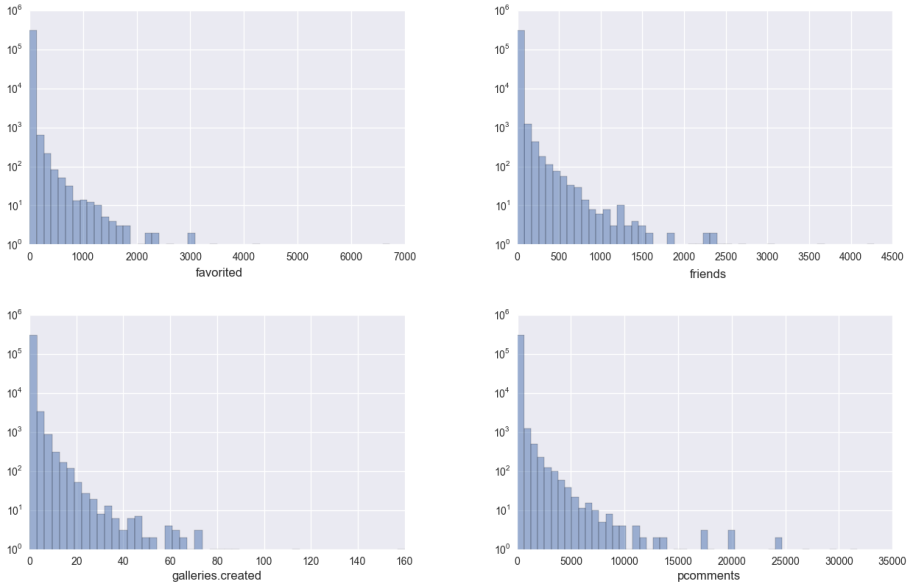

Fig. 2. Distribution of users in terms of number of favorited projects, friends, galleries created and comments posted in project pages. The vertical axis is in logarithmic scale.

In the dataset there are 68,604 users who have created at least 5 projects. Table 1 shows the mean, standard deviation and percentiles for each of the social activities taken into account. If medians are considered, the average user makes one friend, writes five comments, and creates no galleries nor favorites any project. These results indicate that, even for those users who have published several projects and have therefore had time to interact in the community, the immense majority of them barely make use of the social capabilities offered by the `Scratch` website.

Table 1. Social activities of users with at least 5 published projects.

	Galleries	Friends	Favorited	Comments
Mean	0.94	12.72	11.42	100.05
Std	2.55	65.33	69.30	538.75
10%	0	0	0	0
20%	0	0	0	0
30%	0	0	0	0
40%	0	0	0	2
50%	0	1	0	5
60%	0	2	1	10
70%	1	4	3	21
80%	1	8	7	49
90%	3	21	19	161
100%	160	4,281	6,721	31,669

However, there seems to be differences in the projects of users who participate in the community and those who do not make use of social features. Table 2 shows the number of blocks, different types of blocks, costumes, sounds, and user generated strings of projects that are included in collaborative galleries and projects that are not. As can be seen, projects in collaborative galleries are larger, use a broader set of instructions and seems to be more elaborated, as they include more costumes, sounds and texts. We think that future research should further study the differences of learning and development between *social* and *non social* users.

Table 2. Characteristics of projects in collaborative galleries and projects not in them.

	Not in collab gallery	In collab gallery
n	1,469,386	459,313
Blocks	100.84	152.24
Type of blocks	12.44	14.31
Costumes	17.20	25.84
Sounds	3.75	4.86
Ugstrings	36.15	55.01

5 Conclusions and future research

This work is the first step of a series of investigations aiming to identify the impact of social participation on the development of computational thinking skills by analyzing data available in the `Scratch` platform. Specifically, the research question that we tried to answer in this paper is how social is, on average, the `Scratch` community in terms of comments, friends, favorites and galleries.

The results indicate that the vast majority of `Scratch` users barely make use of the social capabilities offered by the website. In fact, if medians of users who have published at least five projects are considered, the average user makes one friend, writes five comments, and creates no galleries nor favorites any project. Future research should compare this level of activity with other social, coding communities.

It must be taken into account, though, that since 2012 there have been important modifications in the `Scratch` website focused on the enhancements of users' social participation. Hence, while 283,118 comments were generated during October 2012, that figure increases to 2,232,072 in the same month of 2015[3]. Consequently, access to the new dataset with data of the activity in the community in more recent years would allow us to perform new investigations that could yield different conclusions.

Furthermore, we must note that the social participation that has been analyzed in this paper is limited to online activities. Nonetheless, other social actions performed in offline contexts, such as helping a peer in a classroom or working in teams, which also have effects on the development of their coding skills, are out of the scope of this study.

At this moment we are performing an investigation that analyzes the impact of social participation in the learning of programming skills. In order to measure the coding skills we have modified Dr. `Scratch` [10], a static code analyzer for `Scratch` projects that automatically assesses computational thinking skills [11], making adjustments based on the data of the *Project Analytics Datasets*. In consequence, the results of future research will allow us to measure whether, and to what extent, actively participating in a software community accelerates the learning of programming skills.

6 Acknowledgments

The work of the authors has been funded in part by the Region of Madrid under project "eMadrid - Investigación y Desarrollo de tecnologías para el e-learning en la Comunidad de Madrid" (S2013/ICE-2715). We are really thankful to the `Scratch` Research Data *team* for granting us access to the data, especially to Benjamin Mako Hill and Andrés Monroy-Hernández.

[3] See https://scratch.mit.edu/statistics/

References

1. K. Brennan, A. Monroy-Hernández, and M. Resnick. Making projects, making friends: Online community as catalyst for interactive media creation. *New Directions for Youth Development*, 2010(128):75–83, 2010.
2. S. Cooper, W. Dann, and R. Pausch. Alice: a 3-d tool for introductory programming concepts. *Journal of Computing Sciences in Colleges*, 15(5):107–116, 2000.
3. A. Dahotre, Y. Zhang, and C. Scaffidi. A qualitative study of animation programming in the wild. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, page 29. ACM, 2010.
4. S. Dasgupta, W. Hale, A. Monroy-Hernández, and B. M. Hill. Remixing as a pathway to computational thinking. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, CSCW '16, pages 1438–1449, New York, NY, USA, 2016. ACM.
5. D. A. Fields, M. Giang, and Y. Kafai. Programming in the wild: trends in youth computational participation in the online scratch community. In *Proceedings of the 9th workshop in primary and secondary computing education*, pages 2–11. ACM, 2014.
6. D. A. Fields, M. Giang, and Y. B. Kafai. Understanding collaborative practices in the scratch online community: Patterns of participation among youth designers. In *To see the world and a grain of sand: Learning across levels of space, time, and scale: CSCL 2013 Conference Proceedings*, volume 1, pages 200–207. International Society of the Learning Sciences, 2013.
7. B. Harvey, D. D. Garcia, T. Barnes, N. Titterton, O. Miller, D. Armendariz, J. McKinsey, Z. Machardy, E. Lemon, S. Morris, and J. Paley. Snap! (build your own blocks) (abstract only). In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 749–749, New York, NY, USA, 2014. ACM.
8. M. MacLaurin. Kodu: end-user programming and design for games. In *Proceedings of the 4th international conference on foundations of digital games*, page 2. ACM, 2009.
9. A. Monroy-Hernández and M. Resnick. Empowering kids to create and share programmable media. interactions, 15: 50–53. *ACM ID*, 1340974, 2008.
10. J. Moreno-León and G. Robles. Dr. scratch: A web tool to automatically evaluate scratch projects. In *Proceedings of the Workshop in Primary and Secondary Computing Education*, WiPSCE '15, pages 132–133, New York, NY, USA, 2015. ACM.
11. J. Moreno-León, G. Robles, and M. Román-González. Dr. scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia*, 15(46), 2015.
12. M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai. Scratch: Programming for all. *Commun. ACM*, 52(11):60–67, Nov. 2009.
13. C. Scaffidi and C. Chambers. Skill progression demonstrated by users in the scratch animation environment. *International Journal of Human-Computer Interaction*, 28(6):383–398, 2012.
14. N. F. Velasquez, D. A. Fields, D. Olsen, T. Martin, M. C. Shepherd, A. Strommer, and Y. B. Kafai. Novice programmers talking about projects: What automated text analysis reveals about online scratch users' comments. In *System Sciences*

(HICSS), 2014 47th Hawaii International Conference on, pages 1635–1644. IEEE, 2014.

15. O. Zuckerman, I. Blau, and A. Monroy-Hernández. Children's participation patterns in online communities: An analysis of israeli learners in the scratch online community. *Interdisciplinary Journal of E-Learning and Learning Objects*, 5(1):263–274, 2009.

Open Source Collaboration Tools and Organisational Learning: On Adopting EtherPad in Small Companies

Yacine Atif, Jonas Gamalielsson and Björn Lundell

University of Skövde, Skövde, Sweden
{yacine.atif,jonas.gamalielsson,bjorn.lundell}@his.se

Abstract. Organisations could benefit from the improvements in collaborative learning and increased control gained by the use of open source software. EtherPad enables simultaneous updates on a shared document from remotely located participants. We discuss the adoption life-cycle of this open source tool in the context of a strategic intervention, with a focus on a learning agenda to guide the installation and configuration processes, and to reduce the learning curve involved in the adoption experience.

1 Introduction

Many organisations have requirements for long-term maintenance of software systems and associated digital assets [5]. One type of software systems that is important for small companies revolves around tools to support collaboration between employees who may be geographically distributed over different sites. In some circumstances proprietary cloud solutions for collaboration (e.g. Google Docs and Office 365) may be inappropriate for use in a small company context for a number of reasons. For example, there are risks of being locked-in in various ways (such as file format lock-in, product lock-in, vendor lock-in, and contract lock-in). One means for mitigating such risks is to utilise open standards and open source solutions [4]. In this paper, we focus on how small companies can support organisational learning through the adoption of a specific open source tool (EtherPad) for collaboration between employees.

Over the past decade, many innovative small companies have realised the potential of engaging with open source projects and incorporated mission critical open source software as part of their IT infrastructure. Strategic use of open source software in small companies provides an effective means for maintaining control of software systems and business critical digital assets over long life-cycles [2]. The EtherPad tool (available at http://etherpad.org/) is one example of open source software, which provides means for collaborative creation and maintenance of documentation and notes when individuals are geographically distributed. By use of this tool in a small company context, it is possible to maintain control of sensitive

data on local servers within the company ecosystem. For some usage scenarios, it may be critical that data resides within the organisational context and that data remains within national boundaries.

The overarching goal of the study is to demonstrate how the EtherPad tool may provide a means for organisational learning and collaboration within a small company context. There are two specific objectives. First, we present a strategy for a planned intervention of introducing EtherPad in a small company context. Second, we illuminate how the EtherPad tool may be adopted and effectively implemented and used in a small company context.

2 Intervention Strategy

Open source tools offer a sustained business growth through the brainpower of broader open source communities and implementations of open standards. The potentially significant business benefits generated from immense cost-reduction are prompting small companies to learn about and adopt open source tools and cloud services. EtherPad is an open source tool, which incorporates features to facilitate collaboration through remote personal interactions. A planned intervention facilitates the adoption process through a learning strategy that increases the rate of EtherPad diffusion across coordination meetings. This strategy includes a major learning support element of both the tool and the collaborative knowledge when using it.

The proposed intervention strategy assumes a preliminary diagnosis of a business process addressed by the intervention [7], which consists in facilitating coordination meetings. The intervention needs also to be driven by potential improvements in the business process, which in our case translates into enhancing employee experiences with cognitive collaboration skills through EtherPad [1]. However, in this paper we focus on an action plan for the intervention, by rolling out EtherPad through a systematic installation process and a persuasive experiential learning activity.

3 Intervention Plan

Our approach to open source adoption in small companies is problem-driven to trigger the motivation for learning and embracing open source tools. In this section, we reveal the problem context followed by a suggested solution that integrates the EtherPad open source tool. We then show a design approach and list practical implementation steps to walkthrough the proposed solution. Finally, we present the tool in action along an illustrative business scenario to learn about using the tool in the suggested problem context.

3.1 Problem

Small companies are frequently faced with situations where coordination meetings bring together geographically distributed parties to brainstorm on a future action,

such as promotion, marketing campaign, or planning the development of a new product. Consider a Swedish company headquartered in Stockholm, which manufactures laboratory equipment that are further distributed via agents based in Gothenburg and Malmo, respectively. Distributors supply offshore customers in Budapest and Warsaw. Regular meetings are arranged by headquarter executives with distributor agents and sometimes customers as well to collect feedback and assess the market needs. The proceedings of each meeting are collaboratively elaborated and need to be recorded via an online shared platform that is privately hosted to avoid any leak of information to potential competitors. The final notes are published locally at the company's internal electronic space and made available in a secured environment. Archived versions are stored in plain text formats to be preserved digitally, which maintain vital links of the company intelligence into the past.

The need for true collaboration and snapshots recording of meeting proceedings prompt the adoption of EtherPad. The learning curve needs however to be smoothed out through a systematic instructional roadmap to set stakeholders to speed, and reduce operational costs. In doing so, stakeholders also discover the benefits of collaborative learning and the merits of collective decision-making.

3.2 Solution

The proposed scheme suggests combining an online teleconference application with a collaborative notes-taking open source tool, namely EtherPad. A preliminary meeting agenda document is set out by the management executives to discuss the functions involved in fulfilling customer requests in terms of product development, marketing, operations, distribution, finance and customer service. In this business scenario, roles are attributed to the meeting participants to intervene freely and contribute to the meeting with real-time contents and within designated sections of the shared agenda document. As information flows across the shared document driving market reflections from distributors and selected customers, management decisions are drawn from a dashboard-like platform that is correlated to the supply chain profitability.

3.3 Design

The advent of cloud computing and open source platforms spawned a suite of low-cost evolving software and platforms. As an alternative to using the public cloud however, a private configuration may be prevailed. Our intervention strategy in this paper promotes a platform for content sharing and development to facilitate a private virtual collaborative workspace among participants in coordination meetings. This approach expects a host server application to be installed in a computing node from which EtherPad services are made available to selected consumers. This is how a meeting is initiated in our problem context, where all meeting data are actually recorded in a local host server. Fig. 1 shows a high-level architecture of this configuration.

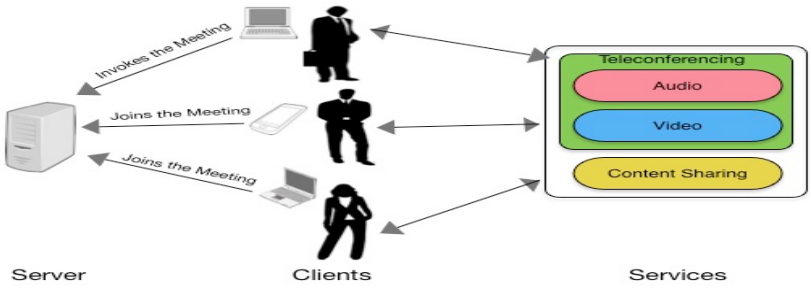

Fig. 1. Collaborative workspace

4 Implementation

To implement the design solution for the collaborative meeting described earlier, EtherPad is used as content sharing platform. EtherPad is an open source tool, which provides Web based pads for real-time text-based group discussions. EtherPad comes out of the box with a pre-configured database and SSL support. The pre-installed open source database management system (DirtyDB) can handle up to one million records. Beyond this threshold, and for further production use, the MySQL open-source DBMS is recommended. The migration from DirtyDB and the configuration of MySQL are facilitated by the highly configurable options provided by EtherPad. In addition, EtherPad is installed from the latest version of the upstream GitHub repository to ensure consistency with released updates and patches.

As shown in Fig. 2, EtherPad interface embeds a chat window that is visible to all meeting members. The toolbar offers a range of functionalities including formatting and scrolling back across time to visualise past snapshots of the meeting notes. There is no particular authentication process to participate in the meeting. Instead, the meeting link to the shared pad is distributed among relevant parties only. The implementation process is mainly handled at the server-side that hosts the EtherPad service. This one-time process makes the open source tool available for private use among all meeting attendees. Next, we reveal a systematic step-by-step installation guide to setup EtherPad. Further details about this process can be found in: https://github.com/ether/etherpad-lite/wiki and http://blog.etherpad.org.

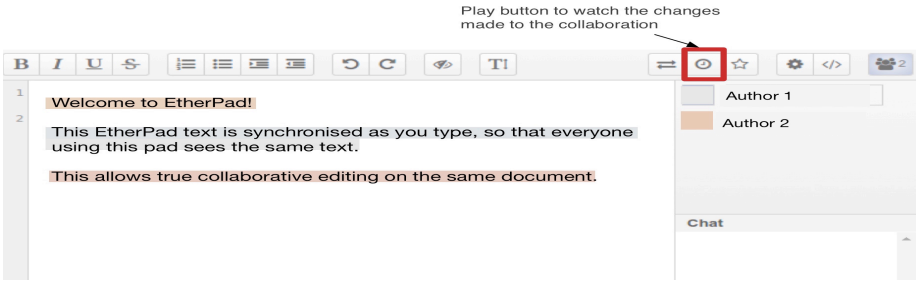

Fig. 2. EtherPad interface

4.1 Server Installation

The installation instructions listed in Fig. 3 assume a dedicated server. However, a private cloud server option is possible using open source virtualisation technologies available through the CentOS project [6]. In either case, the server installation process requires a prior library package and a bridge with the EtherPad original repository to sync the installed version with remotely released updates.

Step 1: Install Node.js This server-side JavaScript package allows lightweight real-time data-push over Web sockets used to facilitate two-way connections among distributed pads. The package can be downloaded from nodejs.org, and then installed locally.
Step 2: Create a folder where EtherPad is hosted Since EtherPad is an open source tool, all development resources related to this project will be downloaded as well within this folder, including source codes.
Step 3: Clone the Git Repository The open source tool is downloaded from an open source version-control platform, namely GitHub. This approach bridges the local installation with the original project to ensure seamless updates. The following command creates a clone of the current version of the tool project files: `git clone git://github.com/ether/etherpad-lite.git` A new folder will be created with the name etherpad-lite. We should position the workspace within this folder using the command line before proceeding to the next step.
Step 4: Run EtherPad To start EtherPad service, the following command needs to be typed: `bin/run.sh` EtherPad could then be tested on the local machine by pointing the browser to the following URL: http://127.0.0.1:9001. Once EtherPad appears on the browser window, it indicates a successful installation of EtherPad as a Web service.

Fig. 3. EtherPad installation steps

4.2 Client Invocation

The Distribution Manager positioned in the head office at Stockholm would run an instance of EtherPad service to prepare for the meeting. This client request results in a new pad, which has a unique name to be shared among the other meeting attendees. The link to this pad is conveyed to the other parties involved in the meeting, who are remotely positioned. The URL to a pad is made-up of the EtherPad host address followed by the port number 9001 used by the server to listen to incoming EtherPad traffic. This URL creates a new pad. Attendees who join the meeting enter the same URL and then they are prompted to enter the shared pad name. All parties using this

URL will view on their browser a common pad space used for collaborating on the meeting agenda items using this link: `http://"Host server IP address":9001.`

4.3 Plugins

Although EtherPad could now be made functional, there is an exhaustive list of plugins that can enhance its capabilities. To access this list, we need to access the admin interface using the following URL: `http://"Host server IP address":9001/admin/plugins.` EtherPad will respond by requesting the admin credentials, which can be obtained and updated by uncommenting the "users" section of EtherPad configuration file "settings.json" located in the root of EtherPad folder, as illustrated in Fig. 4.

```
/* "users": {
  "admin": {
    "password": "change1",
    "is_admin": true
  },
  "user": {
    "password": "etherpad",
    "is_admin": false
  }
}, */
```
Before

```
"users": {
  "admin": {
    "password": "etherpad",
    "is_admin": true
  },
  "user": {
    "password": "etherpad",
    "is_admin": false
  }
},
```
After

Fig. 4. Enabling "admin" user in EtherPad

This configuration amendment enables the access to EtherPad plugins interface with a given password (which in this case is "etherpad"). The plugins interface could be used to extend the capabilities of EtherPad. We would particularly install the toolbar plugin, which provides a menu style file toolbar shown in

Fig. **5**. The toolbar plugin reveals the file menu, which enables imports or exports of files to or from EtherPad in text, html or EtherPad format. The extension to proprietary document formats (e.g. DOC) and open document formats (e.g. ODF), can also be obtained with the installation of AbiWord on the local system, followed by a further adjustment in the configuration file (i.e. "settings.json"). This configuration amendment consists in replacing the command line: `"abiword": null,` with `"abiword":"/usr/bin/abiword"`.

The plain text export option could be used to maintain an archival duplicate of data generated by EtherPad, for digital preservation purposes. This ensures a sustainable access to the information in archived documents.

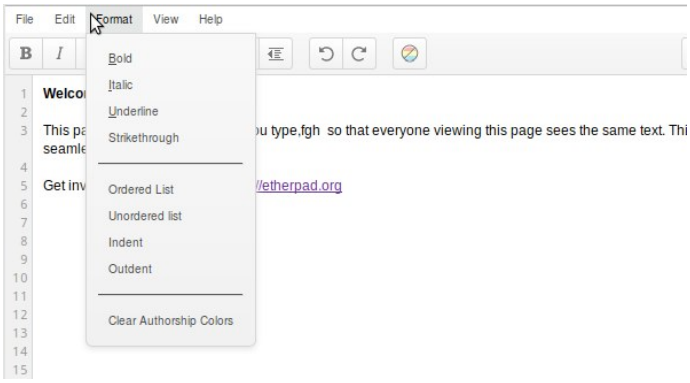

Fig. 5. Menu style toolbar plugin

4.4 Scenario

We elaborate a scenario for organisational implementation of EtherPad in which market changes dynamically prompt a company to review its distribution channels regularly to respond to customer needs. A coordination meeting initiated by the head office of a small company based in Stockholm sets an agenda for this purpose. The feedback collected from neighbouring Gothenburg and Malmo agents reveal a new strategic move to optimise the company distribution channels, as illustrated in Fig. 6. The colours distinguish intervening parties and the critical decision is built collaboratively.

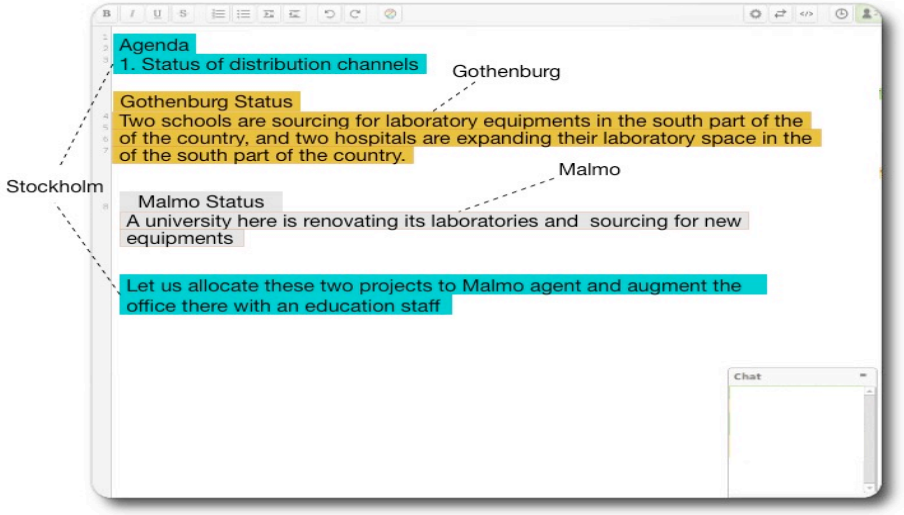

Fig. 6. EtherPad in Action

5 Conclusion

We have presented an intervention approach to drive adoption of an open source tool in a small company context. We also revealed a rollout plan to assist the adoption process by addressing a recurrent organisational need pertaining to collaborative learning. The intervention approach and rollout plan proposed in this study promote technology transfer from open source communities. We find that important implications for practice are that the adoption of an open source tool aid organisational learning, and that organisations gain control of business critical software systems and associated digital assets. Furthermore, we conjecture that an organisational implementation of the open source tool investigated could empower companies to collaborate on the creation of common assets that they can jointly use in product development. Finally, an organisational implementation which involves engagement with an open source project may be strategically benificial for a company as they adopt the work practice of contributing to the open source community. Hence, such a practice will also be beneficial for the broader open source community and promote learning amongst those involved.

References

1. Burns, M., Bodrogini, P.W.: “The Wisdom of Practice”: Web 2.0 as a Cognitive and Community-Building Tool in Indonesia. In Thomas, M (Eds.): Digital Education, Opportunities for Social Collaboration. Palgrave Macmillan’s Digital Education and Learning Book Series, pp. 167–193. Palgrave Macmillan US (2011).
2. Gamalielsson, J., Grahn, A., Lundell, B.: Learning through analysis of coding practices in FLOSS projects. In Robles et al. (Eds.) Proceedings of FLOSSEdu 2012: FLOSS Education - Long-term Sustainability, pp. 13-19 (2012).
3. Gamalielsson, J., Lundell, B.: Sustainability of Open Source software communities beyond a fork: How and why has the LibreOffice project evolved? Journal of Systems and Software 89:128–145 (2014).
4. Haick, B., Klautau, A.: Free software tools for IT management and processes organisation: Case study in scientific research environment. In Proceedings of WMSCI 2013 - 17th World Multi-Conference on Systemics, Cybernetics and Informatics (2013).
5. Lundell, B., Lings, B., Syberfeldt, A.: Practitioner perceptions of Open Source software in the embedded systems area. Journal of Systems and Software 84: 1540–1549 (2011).
6. Membrey, P., Verhoeven, T., Angenendt, R.: Introducing CentOS. In: The Definitive Guide to CentOS. pp. 3–11, Apress, Berkeley, CA (2009).
7. Rider, L.: Coaching as a strategic intervention. Journal of Industrial and Commercial Training 34: 233–236 (2002).