

Community volunteers in collaborative OER development

Irwin J. DeVries

Open Learning

Thompson Rivers University, Canada

The purpose of this comparative case study is to explore and examine the practices of open course design and development community volunteers undertaken in the Open Education Resource universitas (OERu) network, an international partnership of member post-secondary institutions. With a focus on the design and development of an OER-based university-level course, the study identifies and describes features of an OERu open design and development volunteer community and compares and contrasts it to a similar community in the free and open source software (FOSS) development field.

Keywords: OER, free and open source software, open course design and development, OERu

Introduction

The purpose of this study is to explore the formation and development of a small community of volunteers who undertook the work of designing and developing an open course in the Open Educational Resource universities (OERu) using an open design and development process. The OERu is an expanding network of over 30 post-secondary institutions and organizations worldwide committed to building OER-based courses and programs, and to providing formal recognition for course completion.

Collaborative open course design and development such as that taking place in the OERu is a relatively new phenomenon in higher education. I therefore chose to employ a comparative case study research design (Cresswell, 2013; Stake, 2006) that would enable insights to be gained from a comparison with an open design and development process in a similar field. After an extensive search I located a suitable comparator case in the field of free and open source software (FOSS), where communities of volunteers have for many years collaborated in the open to product open source products. The comparator case study (von Krogh, Paeth & Lakhani, 2003) was similar in many ways in scope, size and structure with the OERu course development project under study. Data were gathered from developer communications, artifacts and developer contribution histories within the OERu's wiki-based development environment, and from semi-structured interviews with developers. A process of thematic coding and analysis led to the emergence of four themes: ethos and motivation for participating in OERu course development; induction and persistence of volunteers; division of labour; and coordination and communication. Each of these themes is now described, followed by a discussion of findings and conclusion.

Motivation and ethos

What motivates volunteers to engage in the difficult work of open design and development? Developers interviewed were all highly educated and experienced educators, with busy careers outside their volunteer work in the OERu. In both open design and development and free and open source software (FOSS), developers expressed strong motivation to participate. All OERu volunteers interviewed shared freely their strong personal philosophies concerning reducing barriers to education and credentials, and support for the growth of open educational resources and practices. They saw benefits to their and their institutions' participation in open design and development projects, particularly where their institutions viewed such engagements as potential catalysts for innovation and transformation. Those in FOSS also wanted to make a contribution to the public good as well as gain skills and participate in the development of software that might be of use to them personally or organizationally as well (Choi & Pruett, 2015; Baytiyeh & Pfaffman, 2010; von Hippel & von Krogh, 2003).

The ethos among developers in the FOSS culture was quite similar to open design and development in the OERu in both respects described by Oberg (2003): open processes and philosophies. OER were rooted in an ideology of sharing content in a free cultural works environment, and FOSS similarly was fostered in the ethos of the GNU General Public License (GPL) and other “open” licenses, which then served as the basis for Creative Commons. Developers in OERu unanimously expressed deep commitment to the philosophies of openness and sharing. For example,

My passion [*is*] to share knowledge. I believe education is a fundamental right, and OER is a vehicle to realizing that mission of widening access...

This developer wanted to enable “more affordable access to post secondary education” and was attracted to the OERu because of the fact that

...it’s open in all material respects — in terms of its licensing and in terms of its philosophy, in terms of the mission of what the OERu is trying to achieve. All knowledge should be free. It’s part of being, and my philosophy is knowledge is there to be shared.

All participants expressed similar commitments to a philosophy of sharing educational resources and opportunities that they reported affirming at a deep personal level. In the words of another developer,

Well, I am just a big proponent for the philosophy of open. I just think education is meant to be shared ... it makes no sense to me that someone would create something that is useful for students learning and then you put it away, lock it away in your own desktop or, I just can’t compute that. So, I have my own philosophy, all my years, the minute I find something that looks interesting, whether it’s an article, whether it’s a media piece, I immediately take the time to find out who might find it useful. So I totally 100% believe in open. Sharing knowledge, sharing and reaching out ... not just to give but to have that community where you can collaborate, where you can ask of the people for help.

And in the words of another developer,

I was never hiding whatever resources or things I’ve developed...It’s not a treasure that I have to hide and lock in my desk. So I guess it is in a way a personal philosophy.... I didn’t need much of persuasion or conviction to say this is a good thing. I kind of knew it is.

Similarly in free and open source software (FOSS), many volunteer development communities are formed to contribute to the “greater good” (Baytiyeh & Pfaffman (2010, p. 1348). Other rewards such as participation in a community, social engagement, recognition and identity construction are expressed as motivators by FOSS developers (Fang & Neufeld, 2009), elements also highlighted by OERu developers in their interviews. For example, one of the main reasons for one developer’s joining was stated as his personal commitment not only to professional development as a university faculty member; but also,

I have a personal interest in all open initiatives because personally I’m very committed to bringing education to developing countries, bringing education to those who need it.

In a somewhat similar vein, as reported by Dahlander and Wallin (2006), some developers in FOSS also participate as salaried employees “volunteered” by corporations or universities to gain “access and legitimacy” (p. 1256) as well as access to the code. This was also the case with some developers whose time was donated to the OERu by their institution, which saw a strategic advantage in making such a contribution.

Induction and persistence

How are volunteers introduced to their project and its community, and how does their participation persist over time? Responding to an open invitation sent to the open OERu email list, a large number of volunteers initially signed up to contribute their time and expertise to the OERu project. This number declined to a smaller fraction who provided substantial contributions or even comments and feedback in the course over time. For instance, 148 virtual participants signed up to participate in initial planning discussions at an early OERu meeting in 2011 in Otago, New Zealand. In the first few weeks following a little more than 30 actually signed up to continue to volunteer to work on the project, and 24 made contributions to the wiki. In the first stage of the project, approximately one third of this number was devoted to developing two courses to completion, and not all of them were original members of the volunteers who originally signed up. A core of these course developers was designated by their institutions to work on their respective courses.

Similarly, the Freenet study (Krogh, Spaeth & Lakhani, 2003) found that only four developers contributed 53% of the accepted versions of code in that project. In comparison, in the OERu course, three developers contributed an estimated 95% of the content additions and revisions in the course; in both cases a small number of developers was doing a large proportion of work needed to complete course design and development. In the Freenet case study success in the FOSS community of volunteers, typical of FOSS development more widely, was found to be related to growth in size of the community of developers, “people who contribute to the public good of open source software by writing software code for the project” (Krogh, Spaeth & Lakhani, 2003, p. 1217). Joining behaviours of coders was a major part of the focus of the Freenet study, where it was found that there was a large discrepancy between those who announced initial interest in participating compared with those who ended up making meaningful contributions. “Joining behaviour” was defined as the pathways or “scripts” that volunteer coders would follow, from initial lurking on the project email list to making useful code contributions. One initial barrier to full participation was the difficulty of the Java programming language that was used in coding the project. Also in the OERu, there was a need to learn the wiki mark-up language and conventions as documented in shared artifacts in order to work effectively in design and development.

Seemingly obvious indicators of early interest from volunteers in FOSS, such as expressing an interest to contribute, making suggestions for improvements, proposing solutions but with no actual code contributions, asking for a task to work on, engaging in philosophical discussions and such activities did not typically indicate a progression to subsequent code contributions. On the other hand, those who offered contributions of code to fix bugs, engaged in general technical discussions, and offered repeatedly to contribute, along with other such activities tended to go on to become active code contributors. Further, the match between their specialization and the work needed was an important element in joining:

An important element of the feature gift giving was that the cost of creating and giving the gift was relatively low to the newcomers. Our interviews with the developers revealed that those that had contributed feature gifts did so on the basis of prior knowledge and experience they had refined in other circumstances (Krogh, Spaeth & Lakhani, 2003, p. 1234).

In the setting of the OERu it became evident that more developers with a wider array of skills would be necessary to increase the pace and number of courses developed. One developer observed,

It's a pilot project of how open is going to work.... we definitely have to open it up to many, many, many more people. That to me is how open is supposed to work. I should have been able to immediately feel that I could ask a fellow ID a question, or ask a production person a question, you know when I was stuck with all those questions.

There was a later perception by an OERu developer who was initially involved that the primary role given to partner institutions in the OERu overshadowed other developers' individual

interests. For instance,

I was a very enthusiastic WikiEducator, but lost my way when the OER university initiative began as it opened doors for universities, but closed doors for me as an independent educator. I will be lurking if that's acceptable as I don't represent a university.

While there was no overt restriction on participation by the wider body of those who were volunteers in other parts of WikiEducator, there was also not a notable effort on the part of the community to aggressively recruit those who had initially expressed interest as the focus did indeed fall mainly upon the partner institutions to develop their courses. Nevertheless there were also many communications and invitations to the wider community to comment and provide feedback on developments.

In both OERu and FOSS, a high degree of involvement by volunteers is seen as important to the quality and quantity of contributions (Xu, Jones & Shao, 2009). In the Freenet study (Krogh, Spaeth & Lakhani, 2003), because growth of numbers increased with participation, there was interest in the perceived benefits that would draw newcomers to the project. Within the OERu wiki, participation of developers showed a small number (three) who were involved at the very outset in terms of producing actual page edits or comments and remaining similarly involved through the initial OERu planning stage, through the planning and completion stages of the course, indicating a relatively low level of continuity or contributors across the project, constituting only 11% of the initial group of contributors. This finding is not necessarily unexpected, as many initial contributors may understandably have had an interest only in the bigger OERu picture. However, it does reinforce the concern expressed by OERu collaborators that the lack of continuity from end to end made it difficult for later developers to complete the project with a sound understanding of original intentions of early developers.

Prior to and alongside the development of OERu courses, overall planning for the OERu was documented in the wiki. A small number of contributors made the largest number of contributions, and one contributor in particular documented most of the discussions and emails in the wiki (Figure 1). A spike in contributions took place early in the project and diminished after that time. The patterns of persistence that emerged in the analysis were of particular interest. They showed both the patterns of continuity of contributors throughout various stages of the project, and the relative amounts of work provided by each. In both cases the patterns provide clues to some of the challenges faced by developers involved in the project.

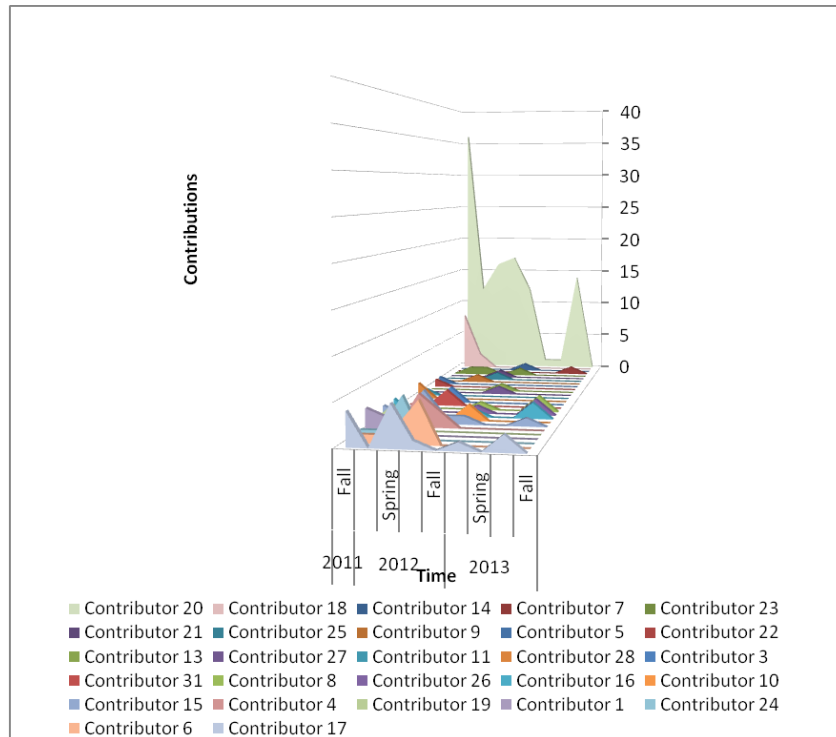


Figure 1: Contributions made in general OERu planning stage

It is helpful at this point to look to another field of collaborative design, architecture. In collaborative design in architecture, developers working together on a design do not typically engage in an ongoing process of negotiation but rather in “...parallel expert actions, each of short duration, bracketed by joint activity of negotiation and evaluation” (Kvan, 2000, p. 412). Similarly, in the OERu course, the most progress in collaboration occurred in occasional conference calls where issues would be settled and tasks negotiated. Developers entering the process later in a project would not have the depth of shared history and understanding as those who had been part of the discussions and negotiations from the very start. They would then need to rely more upon various artifacts in the wiki such as records of previous decisions and notes or revision histories in discussion and history pages. Clearly the process would have benefited from having in place a prescriptive framework for communication roles and strategies among collaborative design teams (e.g., as described by Sonnenwald, 1996), along with effective information retrieval technology.

The existence and maintenance of a robust body of volunteers is identified as vital to the ongoing health of an FOSS project, including the growth of established rules and a group culture that fosters commitment and constructive behaviour patterns (Gallego et al., 2015; Hendry, 2008). A difference noted between induction into the OERu and FOSS was described by a developer:

... in an open source community if you ask a newbie question and you haven't even gone through the previous discussion forums, you will be castigated. So in open source there's this culture of, you go out and read what has been done, and then if you don't know what's happening, then you engage with the community. I've noticed there's a lot more tolerance with education folk.

However, comparing FOSS development with similar practices in the OER, a developer noted:

...the nature of the development [*in FOSS*] is such that you've got objective measures for seniority. You know, if you proved yourself, the code must work and those are the things that it must and this is an objective measure.

The developer further noted that educational development is more forgiving in comparison and

thus any challenges that might be faced by late-joining developers would not necessarily be immediately evident, given in particular that there was, by consensus, no common pedagogical approach to learning design.

In traditional instructional design, typically all participants in the project are either involved in the project from the very beginning, or if brought in later then are thoroughly debriefed on the project's history and status. Collaboration in planning is essential to the success of collaborative development teams (Hixon, 2008) and ongoing communication throughout the process is equally important, along with orientation for all participants to the processes and tools used in the development project (Chiu, 2002). However, a developer in this OERu case was left feeling disadvantaged from the outset:

... the next person down the road might want to do something with the course but they don't have all the same philosophy and all the same agreements that *[others]* had in the beginning. You know, all those conversations ... on why you were doing what you were doing in the way you were doing it. How do we share that with the rest of the world? So I know the lessons are there in this pilot project but it's there in a messy, messy way. We kind of got it in the way of just documenting the process that you would have to clean up because not everybody wants to read through every messy meeting we had. At the end, a different kind of help guide has to come out for the open public A really well put together manual would be something useful for the future folks after we've learned all our lessons. It should be a little more well organized and concise for the people who come after us.

Interestingly, documentation had been developed in the wiki that could have been used by developers, but they were confused by the complexity of the wiki and its flat file structure. Over time another developer pulled these documents together more tightly in one section.

To address the challenge for "newbies" beginning later in the project, a starting point for them would then be, it was suggested in the planning node, a place where some work had already been conducted. The expectation would be to make contributions and even improve others' content, while remaining consistent with the overall direction of the course design. Within this context, however, it was important to have opportunities for developers to gain an understanding of what design thinking had preceded them beyond what was evident in the designed content artifacts or other forms of distributed intelligence. As noted by one developer, there was a need to be able to provide background and context for others just beginning on the course at a later stage. The main way for doing this, apart from abstracting the design from the in-progress artifacts of content and activities, was to review design debates and decisions occurring through and across the OERu wiki and email discussions, and comments provided by developers on talk pages in the relevant section of the course under development. However, this would take a good understanding of the wiki structure and the layout of the OERu, which is complex to a newcomer and takes time to learn.

Beyond these elements, a critical factor in working within the open design and development that did not appear prominently in the Freenet study or in FOSS literature in general was mentoring. Throughout the OERu project the more experienced developers were available to provide support and assistance to the newer participants in development. This was seen by several developers as vital to its success. In the experience of one developer,

[Originally] I didn't even have my own WikiEducator page. *[A mentor]* kind of talked me through how to set up my page, how to bring the images in. She was an email away. She was very, very willing to help. So that made me feel good. *[It]* was really important because I would have given up and not taken part in the project after week 1. Week 2, if *[mentors]* weren't there to help me in that first steep learning curve, then after ... just an email away. Very important because as I said the whole project was difficult for me. If *[a mentor wasn't]* 11 o'clock also online and answering my questions, I think I would ... not *[be]* doing this.

Another viewed membership as a distinctive element that defined open design and development models, based on two key principles of meritocracy and consideration for others in

such acts as mentorship:

One is the principle of meritocracy, where one's seniority — in inverted commas — or respect within a community is actually developed by the expertise you've demonstrated within a community and have built up over the years. So there is this key element of meritocracy. You know it's sitting in these open communities, which is a differentiator. I think it's part of this sort of reward mechanism that's kudos that takes place in these open communities. So I think that is incredibly important. [Second is] the principle of paying forward. And that helps fuel this ecosystem of mentorship. It's this whole notion of...someone helped me when I was struggling. Once I've acquired the skill it's now my turn to help somebody else.

A further challenge encountered was the effort involved in locating, converting, remixing and formatting the content of the original OER into the wiki. Access to a mentor in the form of a highly experienced WikiEducator developer was seen as a crucial support to the developer. This loomed large in the mind of some developers. Thus for those who had not started from the beginning, and hadn't arrived with prior appropriate specializations or training, there was a significant barrier to joining.

At the same time, by joining at the periphery and learning and being mentored, in the manner of a community of practice (Wenger, 1999), a developer who completed a project found it a substantial learning experience and a good basis from which to move forward with many lessons learned, even as part of a larger philosophy about learning:

... it's been a learning experience and I'm looking at everything really that I do as a learning experience because learning is life and life is learning. I'm not sure who said that but that's definitely my point of view. So it's been a great learning experience and I'm continuing to learn and if I'm passionate about others and education, I've got to be committed to keep learning.

While principles of self organization are largely intended to drive the design and development processes in the OERu, the demands of the environment, the potential challenges with conversion of OERs and the need for various levels and types of expertise appear to suggest the potential advantages of some initial recruitment and negotiation of roles among volunteers and the wider community rather than a more informal processes. In the Freenet study it appeared that while there could be potential within a large enough community for a body of developers to flow in and out of projects, but this would not work well in a startup setting.

Division of Labor

A vital component in the success of the community in the Freenet study (Krogh, Spaeth & Lakhani, 2003) was identified as specialization of volunteers, i.e., deployment of volunteer talent according to their specialization for "efficient use of knowledge" (p. 1218). In other words, coders were best utilized by working in their areas of greatest expertise, with the implication that a wider variety of types of expertise was required to supply the specific skills needed for particular aspects of the project. With high turnover as found in the Freenet community, this would become even more important, in order to maintain a "critical mass" (p. 1226) of expertise in each of the areas required to complete the project. FOSS projects typically leave it mainly to new volunteers to "work their way in" based on the quantity and quality of their code contributions, and volunteers typically contribute according to their areas of specialization. In the OERu developers with their characteristic instructional design skill set spent much time working well outside their areas of specialization, owing to the fact that few others either were available to take on the various aspects of the course development work and detailed technical implementation, or developers were not aware of them. This was seen as a barrier to overcome as a developer became more acquainted with the new role of learning design in an open wiki environment. For example:

I didn't really plan to be the technology know-how person in the project because that was not my forte. I really was thinking I'd just bring my design expertise and my educational expertise.

The need for developers to venture outside their initial areas of specialization was evident. As described by a developer whose contribution to the project was initially intended to be based on expertise and interest in open education and online learning pedagogy, large amounts of time were spent on such labour intensive work as converting and correcting OER content files, fixing links, tracking down resources, reassembling content from a confusing set of original course files, and so forth. This was described by a developer as “factory work,” and as somewhat distracting from the design goals that were at front of mind in approaching the project:

One of the challenges we got in our open design communities, is the extent that our technology people actually engaged in the process. We don't have a high number of coders or people at that level of technical skill engaging this development process which is kind of odd because if we purporting in sort of open distance learning, professional team approaches, it would be nice to see that sort of skill engaging as well.

The lack of sufficient expertise in the technical area was noted by another developer, who felt an inordinate amount of time was spent undertaking repetitive, manual tasks in converting and formatting content when the expertise this individual brought to the project was of a different nature, including design expertise and a particular interest in equity and provision of free learning opportunities to those who are disadvantaged:

One of the challenges we have in our open design communities is the extent that our technology people actually engaged in the process. We don't have a high number of coders or people at that level of technical skill engaging this development process.

Yet also there was another OERu developer who didn't seem to mind applying a mixture of skills to course development:

I did find not it too difficult to get used to the wiki mark-up, in particular; it was quite easy, and to be honest I didn't really follow the tutorials either. But they were useful at the beginning, but I just [applied] the same learning strategy I did when I had to learn HTML... once I got the basic grasp of tags. When I find a good feature I like in the wiki page I just go to the mark-up and copy that, and replace the text or the image with my own.

It could be said then that each team will have its unique makeup of skills and interest in performing a broad or narrow array of tasks based on interest, background, time and expertise. Nevertheless, a broader set of skills recruited from the outset will permit more developers to work from their respective strengths and thus avoid unnecessary frustration and discouragement.

Coordination and communication

Another important factor to be addressed is how coordination and communication occur in the OERu and FOSS environments. In the initial months of the OERu project, the ambitious cross-OERu project management process that was started could not be sustained by developers, as the main developer heading it up moved on to another institution and no others expressed an inclination to continue this role. It did not appear that a comprehensive project management process was feasible for the OERu project, owing to the breadth and complexity of the various course development projects, and the time developers would need to contribute to their own projects let alone step up to take on larger responsibilities. Further, it appeared that quasi-regular synchronous virtual meetings among developers were particularly valuable in discussing challenges, reviewing progress, planning next steps and dividing work. These meetings and the subsequent notes kept by one or multiple participants placed in an appropriate page in the wiki were of ongoing value to developers.

In the Freenet study (Krogh, Spaeth & Lakhani, 2003), commitments to code versions were approved by a small group of senior administrators, with increased trust placed in coders who

established a record of high quality contributions. Similarly in the OERu, a meritocracy of developers was seen as a part of an ecosystem where credibility of contributions built up over time would give them increased stature and responsibility in the community. FOSS projects typically display decentralized decision-making and representation, although there are occasions where a formal leadership role or representative body in a not-for-profit foundation is established “to protect the community’s interests” (O’Mahoney, 2007, p. 2). The OERu also is governed by a not-for-profit organization, the Open Education Resource Foundation, with an Executive Director who coordinates the efforts of the OERu and provides much impetus and expertise in moving the OERu community forward. Each of the partner institutions involved in developing OERu courses had a great deal of autonomy as to how the courses were developed, subject to working with the guidelines that had been reached across the partnership by means of polls and rough consensus.

Another area for comparison between FOSS and open design and delivery is communication methods. In support of this emphasis, several of those interviewed noted that it would be helpful for the community to review and further organize many valuable but distributed resources across the wiki into a more structured guide to improve sharing of information. Given the nature of developers and the amount of time that they may be involved in a project such as the OERu, this would of course need to be revisited on an ongoing basis, and it would also need to be recognized that no such system would be perfect given the decentralized nature of the community.

The practice of maintaining notes on discussion pages both to communicate asynchronously in situ with other developers and to leave a record for others who joined later in the process was viewed as a valuable asset. Development teams would need to become more alert to the importance of maintaining understandings at the outset that as much communication as possible should either occur within the wiki or, if external, documented in the wiki as well. For instance, virtual synchronous meetings would have notes taken and placed in the wiki in a designated page for maintaining meeting records. Also in this area a set of links to the key pages that track ongoing OERu-wide discussions within the wiki on common elements of concern to all developers would need to be maintained in order for those who join projects midstream can quickly be oriented to the essential elements of the project.

Discussion

The way that volunteer communities function in the OERu and in FOSS settings including the comparator case showed many similarities throughout the study. In terms of motivation, developers in the OERu expressed a very high level of commitment to the underlying principles and ethos of open education and worked beyond usual hours and/or without pay to complete their project, in a manner similar to FOSS developers (Baytiyeh & Pfaffman, 2010; Oberg, 2003). Also, in FOSS, organizations may donate developer time in order to benefit directly or indirectly from the code under development (Dhalander & Wallin, 2006), and in the same way multiple partner institutions sponsored developers to work on the OERu project. Since such arrangements are organizationally encouraged or even required, such work should become part of a regular workload where possible.

Successful FOSS projects have relatively well-developed processes for orienting new developers to the communication tools and practices proven to be successful in such environments (Chiu, 2002). This includes not only email lists, discussion boards, wikis and versioning tools, but also system-wide views and visible design rules or artifacts that promote the sharing of knowledge and intelligence. Similar tools and practices were present in the OERu but communication habits of developers tended to spread information across the wiki and in scattered emails in a manner that made it difficult to retrace where key information could be found. Course development teams will benefit from establishing and maintaining clear guidelines for communication and documentation methods. These protocols were well documented in the wiki, and an orientation for new members would be beneficial, along with continuing reminders from more experienced developers.

Effective maintenance of FOSS over time improves the quality of the project (Koponen & Hotti, 2005) but requires planning and organization. Above all, new developers who join the project

later in its lifecycle need to be able to gain a sense of the project's history and organization quickly with the help, for example, of such factors as systematic naming conventions of files and logs (Stewart, Darcy & Daniel, 2005). Developers in the OERu prototype project similarly found it necessary but also difficult to become oriented to the project in a short period of time, which would suggest the need for practices similar to those in FOSS that maintain a system for the support of new joiners in a course development project (Chiu, 2002). As noted by O'Mahoney (2007), "when code and community do not develop in parallel, the learning curve can be steep, which can affect external developers' ability and motivation to contribute" (2007, p. 142).

Recruiting, properly inducting and maintaining a robust community of volunteers have proven to be critical components in the success of FOSS projects. Because there was a high attrition among the initial OERu developer recruits, there were fewer developers and other volunteers involved in completion of prototype courses by the final stages of the prototype course than desirable, increasing stress on the remaining volunteers. In FOSS some attrition occurs because of skills barriers; e.g., a programming language that is out of the skill range of potentially interested contributors (Krogh, Spaeth & Lakhani, 2003). However, volunteers who aren't meaningfully engaged don't stay around for a long time in both FOSS and in the OERu (Xu, Jones & Shao, 2009). Successful FOSS projects attract sufficient developers with an appropriate array of skills or specializations to cover off the variety of design and technical needs in a course development project (Krishna Raj & Srinivasa, 2012), and over the longer term bring their experience to the project as mentors or administrators (von Krogh, Spaeth & Lakhani, 2003). The evidence gathered from the OERu wiki and communications emphasize this point. Developers reported that having to take on multiple roles, particularly those that would ordinarily be considered technical in nature such as page design, mark-up and production, diverted their efforts toward focusing on their design strengths. Further, they reported a concern that they had overextended the time they had available to work on the course. While a certain degree of familiarity with the wiki environment is necessary for any wiki developer, engaging in more extensive course development was seen as somewhat onerous. Partner institutions of the OERu could consider an increased effort to recruit both internally and elsewhere a rounded team of developers to complete each course.

Collaboration and communication are fundamental to the practice of open design and development. Not only content but also design knowledge need to be shareable in a wider open education ecosystem such as the OERu network and among volunteer development teams. However, research in the sharing not only of content but also of learning designs, design patterns (Alexander, Ishikawa & Silverstein, 1977) or learning design "know-how" (Dalziel, 2008) indicates that translating learning designs from one setting to another is a complex matter. As noted earlier, one pathway for further investigation is the use of visible design rules that guide a high-level view of the design process, while making knowledge of deeper levels of detail unnecessary at certain points (Hossain & Zhu, 2009). These may be further shared and discussed in discussion spaces as has been seen in FOSS development (Björgvinsson & Thorbergsson, 2007). Research into distributed intelligence (Perkins, 1992) as well as mediating artifacts (Conole & Culver, 2009) points to ways in which design knowledge can become more visible and thus shared in a communal work setting where collaboration is centred on representations open for discussion within the community. While an "artifact appears to be a self-contained object, it is in fact a nexus of perspectives" (Zitter et al., 2009), a resource most important in a setting such as the OERu where the community is distributed globally. Mediating artifacts are both available for access by all and able to be negotiated and changed. Mediating artifacts include discourses and processes supporting coordination and negotiation or brokering between different domains within a community of practice (Wenger, 1999).

As noted by Dimitriadis et al. (2009), "making design more explicit will facilitate repurposing of the OER" (p. 201). Similarly, Conole et al. (2013) emphasize the importance of social networking spaces where designers can discuss and share ideas on learning designs. Such spaces were in fact available in the planning sections of the OERu wiki. However, because development of learning designs was intended to remain the province of each institution and its developers rather than something shared across the partnership, a robust learning design discussion space did not fully emerge. Rather than become lost in individual exchanges scattered across emails and wiki "talk" pages, a concerted effort to concentrate this discussion could have the potential to create a shared body of knowledge on effective learning designs for

the OERu project or similar open design and development contexts. In the OERu the course prototypes developed for stimulating discussions and negotiations toward consensus exemplified the concept of nexus of perspectives. They perform this function by serving first to generate, and then to record, discussions and decisions in brief summaries, similar to what Scacchi (2007) identifies in open source software projects as “lean descriptions” or “documentary artifacts” (p. 473). Similarly, brief descriptions of decisions may have a similar function and are seen as critical to sharing an understanding of the learning design and other issues faced by the developers.

Conclusion

As the work of the OERu progresses and the body of developers grows, an increased effort toward sharing of learning designs ideas and experiences may help create a strong community with established practices, tools and shared understanding. New and creative design approaches must grow from the developer body working across the OERu to face the many challenges and opportunities documented in this study. A balance of dynamic design decision-making and intentional collaboration among developers in learning design and related skill areas will help to support such innovation. Along with this work, the community would be wise to observe and learn from the methods used in the many successful free and open source software projects that have emerged over the past decades.

References

- Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A Pattern Language: Towns, Buildings, Construction*. Oxford, UK: Oxford University Press.
- Baytiyeh, H., & Pfaffman, J. (2010). Open source software: A community of altruists. *Computers in Human Behavior*, 26, 1345–1354. Elsevier Ltd. doi:10.1016/j.chb.2010.04.008.
- Chiu, M. (2002). An organizational view of design communication in design collaboration. *Design Studies*, 23, 187–210.
- Choi, N., & Pruett, J. (2015). The characteristics and motivations of library open source software developers: An empirical study. *Library & Information Science Research*, 37(2), 109–117.
- Conole, G., & Culver, J. (2009). Cloudworks: Social networking for learning design. *Australasian Journal of Educational Technology*, 25(5), 763–782.
- Creswell, J. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches*, 4th ed. Los Angeles: Sage.
- Dahlander, L., & Wallin, M. (2006). A man on the inside: Unlocking communities as complementary assets. *Research Policy*, 35(8), 1243–1259. doi:10.1016/j.respol.2006.09.011.
- Dalziel, J. (2008). Learning design: sharing pedagogical know-how. In T. Liyoshi & V. Kumar (Eds.), *Opening up education: The collective advancement of education through open technology, open content, and open knowledge* (pp. 375–388). Cambridge, MA: MIT Press.
- Dijkstra, S. (2001). The design space for solving instructional-design problems. *Instructional Science*, 29, 275–290. <https://doi.org/10.1023/A:1011939724818>
- Dimitriadis, Y., McAndrew, P., Conole, G., & Makriyannis, E. (2009). New design approaches to repurposing open educational resources for collaborative learning using mediating artefacts. In *Same places, different spaces: Proceedings Ascilite* (pp. 200–207). Auckland. Retrieved from <http://www.ascilite.org.au/conferences/auckland09/procs/dimitriadis.pdf>
- Fang, Y., & Neufeld, D. (2009). Understanding Sustained Participation in Open Source Software Projects. *Journal of Management Information Systems*, 25(4), 9–50. doi:10.2753/MIS0742-1222250401.
- Gallego, M., Bueno, S., Racero, F. & Noyes, J. (2015). Open Source Software. *Computers in Human Behavior*, 66(49), 390-399. <https://doi.org/10.1016/j.chb.2015.03.029>
- Hendry, D. (2008). Public participation in proprietary software development through user roles and discourse. *International Journal of Human-Computer Studies*, 66, 545–557. doi:10.1016/j.ijhcs.2007.12.002.
- Hixon, E. (2008). Team-based Online Course Development : A Case Study of Collaboration Models. *Online Journal of Distance Learning Administration*, 11(4), 1–8. Retrieved from <http://www.westga.edu/~distance/ojdla/winter114/hixon114.html>.

- Hossain, L., & Zhu, D. (2009). Social networks and coordination performance of distributed software development teams. *The Journal of High Technology Management Research*, 20, 52–61. Elsevier Inc. doi:10.1016/j.hitech.2009.02.007.
- Khan, S., & Samuel VanWysberghe. (2008). Cultivating the Under-Mined: Cross-Case Analysis as Knowledge Mobilization. *Qualitative Social Research*, 9(1). Retrieved from <http://www.qualitative-research.net/index.php/fqs/article/view/334/729>.
- Kirschner, P., van Merriënboer, J., Sloep, P., & Carr, C. (2002). How expert designers design. *Performance Improvement Quarterly*, 15(4), 86–104.
- Kvan, T. (2000). Collaborative design: what is it? *Automation in Construction*, 9(4), 409–415. doi:10.1016/S0926-5805(99)00025-4.
- Le Maistre, K., & Weston, C. (1996). The priorities established among data sources when instructional designers revise written materials. *Educational Technology Research and Development*, 44(1), 61–70. <https://doi.org/10.1007/BF02300326>
- Oberg, S. (2003). Bits and Bytes: Serials System Insights: Open Source Software: An Introduction from a Serialist's Perspective. *Serials Review*, 29, 36–39. Retrieved from linkinghub.elsevier.com/retrieve/pii/S0098791303000029.
- O'Mahony, S. (2007). The governance of open source initiatives: What does it mean to be community managed? *Journal of Management & Governance*, 11, 139–150.
- Scacchi, W. (2007). Understanding Requirements for Open Source Software. In K. Lyytinen, P. Loucopoulos, J. Mylopoulos, & B. Robinson (Eds.), *Design Requirements Engineering: A Ten-Year Perspective* (pp. 467–494). Cleveland, OH: Springer.
- Sonnenwald, D. (1996). Communication roles that support collaboration during the design process. *Design Studies*, 17(3), 277–301. [https://doi.org/10.1016/0142-694X\(96\)00002-6](https://doi.org/10.1016/0142-694X(96)00002-6)
- Stake, R. E. (2006). *Multiple case study analysis*. New York: The Guilford Press.
- von Hippel, E., & von Krogh, G. (2003). Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science. *Organization Science*, 14(2), 209–223.
- von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, 32, 1217–1241. doi:10.1016/S0048-7333(03)00050-7.
- Wenger, E. (1999). *Communities of practice: Learning, meaning, and identity*. Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9780511803932>
- Xu, B., Jones, D., & Shao, B. (2009). Volunteers' involvement in online community based software development. *Information & Management*, 46(3), 151–158. doi:10.1016/j.im.2008.12.005.
- Yin, R. (2009). *Case study research* (4th ed.). Thousand Oaks, CA: SAGE Publications, Inc.
- Zitter, I., Kinkhorst, G., Simons, R., & ten Cate, O. (2009). In search of common ground: A task conceptualization to facilitate the design of (e)learning environments with design patterns. *Computers in Human Behavior*, 25(5), 999–1009. Elsevier Ltd. doi:10.1016/j.chb.2009.01.001.

DeVries, I.J. (2015). Community volunteers in collaborative OER development. In T. Reiners, B.R. von Kinsky, D. Gibson, V. Chang, L. Irving, & K. Clarke (Eds.), *Globally connected, digitally enabled*. Proceedings ascilite 2015 in Perth (pp. 77-88). <https://doi.org/10.14742/apubs.2015.987>

Note: All published papers are refereed, having undergone a double-blind peer-review process.



The author(s) assign a Creative Commons by attribution licence enabling others to distribute, remix, tweak, and build upon their work, even commercially, as long as credit is given to the author(s) for the original creation.