

Applying Software Development Paradigms to Open Educational Resources

Seth Gurell

Center for Open and Sustainable Learning, Utah State University

Summary

This article is a case study in the development of the OER Handbook¹ on the website WikiEducator. WikiEducator is a website run by the Commonwealth of Learning dedicated to the creation of OER for the classroom. Open educational resources² are defined as content that is licensed that allows for modification and distribution without the copyright holder's permission. The handbook is meant to guide educators (an admittedly broad category) in finding, creating, adapting and sharing open educational resources. Parallels are drawn between the development of the OER Handbook and typical open source software development, especially in the development of a polished product. The idea of open texts having a release cycle similar to open source software is explored. Comparisons between how open source software handles errors and feedback from users are analysed with similar functionality in the OER Handbook project.

Difficulties in wiki development, such as renaming of pages, organizing pages, formatting text and image manipulation are outlined, though it is noted that many wiki projects are successful. The suggestion to use Subversion, well-known tool for managing software development, as a model for a new tool to develop OER is also addressed. This paper discusses the similarities between the software development process and tools with OER creation and provides best practices from both perspectives.

Keywords: OER, open source, wiki, educational resources, open texts

1 Background

Before launching into a discussion of a OER Handbook project it is appropriate to give background on WikiEducator³. Though WikiEducator uses the same MediaWiki⁴ software as Wikipedia, Wikibooks, etc., each wiki carries its own history and culture that influences the dynamics of content creation.

1.1 WikiEducator

WikiEducator has its origins within the Commonwealth of Learning (COL)⁵, which is an intergovernmental organization dedicated to e-Learning. At the time Wikimedia, Inc., creators of Wikipedia, Wikibooks and WikiNews, was struggling with the placement educational materials beyond encyclopedia entries. Wayne Mackintosh, a member of COL, didn't find a website that completely matched his vision, though there were many good OER sites available. So he created WikiEducator as a means of creating, distributing and promoting OER. The website went online May 2006 (Mackintosh, 2007).

1 http://www.wikieducator.org/OER_Handbook/educator

2 http://en.wikipedia.org/wiki/Open_educational_resources

3 <http://www.wikieducator.org>

4 <http://www.mediawiki.org/wiki/MediaWiki>

5 <http://www.col.org/colweb/site>

Since its inception, WikiEducator has boasted high number of active contributors in comparison to other larger wiki projects (Mackintosh, 2008). The result is an amicable community that is supportive of newcomers questions, technical or otherwise.

1.2 OER Handbook

The OER Handbook project was initiated by the Center for Open and Sustainable Learning (COSL)⁶ with funding by the William and Flora Hewlett Foundation⁷. The purpose of the project, in the author's words was "to provide a beginner's guide to creating OER material." In the handbook, OER is defined as "educational resources (lesson plans, quizzes, syllabi, instructional modules, simulations, etc.) that are freely available for use, reuse, adaptation, and sharing" (Gurell, 2008). In this paper, OER specifically refers to largely text projects, such as textbooks, though the author recognizes that images often accompany these projects. The audience is educators in general who have a basic understanding. The goal is provide a general overview of OER development and distribution, though it is only an introduction. Depending on the OER being produced, educators will need to learn new software and consult with their institution's legal department.

The project was announced on Wiley's (2008) blog in January. WikiEducator was chosen as the host site, because of its activity, licensing and the willingness of the Commonwealth of Learning to assist in the project.

The handbook is organized around the OER life cycle developed by Wiley (2008) and Tucker (2008). The idea behind the OER life cycle is that the development, use and distribution of OER follows a general pattern. The author and Wiley chose the OER life cycle as the foundation for the OER Handbook, because it provides a simple, cyclical method of OER production. However, it is important to note that there is no definitive method of creating OER; the life cycle is just one proposed strategy.

The initial version of the life cycle was created by Wiley and approved by the author in January 2008. The final version was created by input from the author, Wiley and Tucker through discussion both online at WikiEducator and in face-to-face discussion in June 2008. Since the handbook was organized according to the OER lifecycle, it was necessary to finalize a version of the life cycle in time for publishing in August 2008. In a sense, the word "final" is somewhat misleading, as the author expects the OER life cycle go through future iterations as more research is conducted. The following table shows what was originally written on WikiEducator about the life cycle compared to the version included in the life cycle.

Phase of Cycle	
Initial	Final
<p>Get: Searching and finding OER. Getting OER may include using search engines, repositories and finding individual websites. Some potential OER material is not online, including things like class projects. This handbook will show you how find quality OER materials.</p>	<p>Find: start by looking for suitable resources which contribute to meeting the need or satisfying the desire. This may include using general search engines, searching specific repositories and finding individual websites. Some potential components may be available offline, including last year's lecture notes, class projects, handouts for learners and other resources prepared previously.</p>
<p>Localize: Localizing is a complex topic. Essentially localizing means making a resource more useful to a particular situation. For example, translating instruction from one language to another.</p>	<p>Compose: with a collection of resources at your disposal, start piecing them together to form a learning resource for yourself, your fellow educators and/or learners. This is a creative design process of building an educational resource from scratch and/or using components you have found.</p>

6 <http://cosl.usu.edu>

7 <http://www.hewlett.org/Default.htm>

Remix: Remixing is the act of taking two OER materials and merging them to form a new OER. Arguably, remixing is one of the most enjoyable parts of OER.	Adapt: while composing OER, it will nearly always be necessary to adapt components to your local context. This may involve minor corrections and improvements, remixing components, localization and even complete rework for use in diverse contexts.
Use: This section covers the actual use of OER.	Use: the actual use of OER in the classroom,online,during informal learning activities, etc.
	Share: once an OER is finished, make it available for the open education community to re-use and begin the life cycle again.
License: Covers Creative Commons and GFDL licenses. Also explains the differences between Creative Commons licenses. The handbook will not advocate one particular license over another, but instead provide multiple perspectives, because which license you choose is a personal choice.	License: select a license based on how the OER is intended to be used and personal values.
Redistribute: Once the OER is finished it should be distributed and made available for the open education community to begin the lifecycle again.	

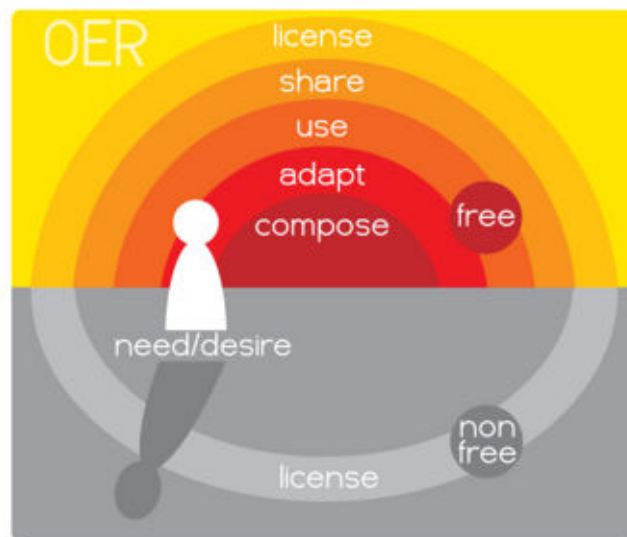


Illustration 1. Illustration of the OER lifecycle from the OER Handbook

1.3 The relationship between FOSS and OER

Free and Open Source Software (FOSS)⁸ and OER have long been intertwined. There have been some who have suggested that OER is best created by FOSS (Tucker, 2007).

But the relationship between FOSS and OER is not unidirectional. Stallman, often considered one of the founders of the free software movement, offered the idea of a free online encyclopedia prior to Wikipedia (Stallman, 2008).

Once open source software appeared, corresponding open documentation was not far behind. That documentation has varied formats over time, but in general, it is tied to the software release. For example, if Software X was a .5 Alpha release, then the documentation would be at

8 <http://en.wikipedia.org/wiki/FOSS>

.5 Alpha as well. Software in its early stages might only have a plain text file, with eventual growth into a multi-page guide, some even with screenshots or screencasts.

As the construction of the OER Handbook will show, this process-oriented model of development can be applied to OER development.

2 Document Development Following Software Development

During initial talks in December between the author and Wiley regarding the development of the OER Handbook, the goal was set to have a print version of the handbook available by the Open Education 2008 Conference⁹ in September. A rough timeline was laid out, complete with milestones. In this respect it was a very traditional single-iteration publishing plan with very little thought was given beyond the September release.

Despite this traditional publishing plan, software analogies still appeared. In April, when asked to describe the state of the handbook, the author described it as being “definitely not a Beta, but probably an Alpha release.” In retrospect, the author chose this analogy because its audience was familiar with software development and that it was far more accurate of the situation. One alternative, a percentage, is inaccurate because it tends to indicate a complete, finalized status is possible. A percentage may be appropriate for software documentation when it meant to correspond with software or a novel. But with educational materials, a clear end-goals are not always apparent, and an analogy to open source development is more appropriate.

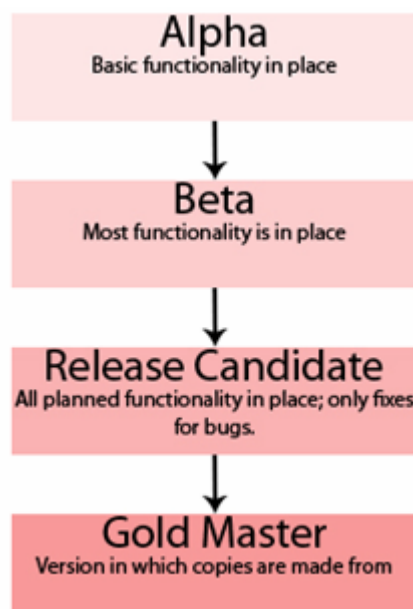
The unconscious move towards an open source software development model was not simply a matter of familiarity or coincidence. An open source software development model provides affordances not granted to by a traditional publishing model. These affordances are necessary to provide the efficiency needed for OER, particularly open textbooks, to become more widespread.

2.1 Advantages of a Software Model

The first advantage is that conveys the understanding that the material presented is constantly under development. This understanding is particularly important when the material is being developed on a wiki in which pages may be changed by any one member of a “flat¹⁰” community (in the case of Wikipedia it can be virtually anyone).

*Illustration 2. Sample software cycle.
Note that there is no single model for software development*

The second advantage is that conveys information about completion without promising an end state. Indeed, having a version 1.0 almost implicitly suggests the possibility of a version 2.0. Authors and contributors can use 'Alpha,' 'Beta,' and perhaps even 'Release Candidate' and 'Gold Master' statuses to give readers an quick understanding of where development stands¹¹.



9 <http://cosl.usu.edu/events/opened2008>

10 <http://www.thomasfriedman.com/bookshelf/the-world-is-flat-3>

11 http://en.wikipedia.org/wiki/Software_release_life_cycle

The third advantage is that it allows for the idea forks or branches from the main document. Allowing for this idea in document development is important because open licenses, such as the GFDL¹² and some Creative Commons licenses¹³, are designed specifically for remixing and reuse. Different versions of the same OER allows for better accommodation of diverse audiences and matches the flexibility found in open source software. Development of educational materials may be aided if reuse is considered forks with the option for merging. Other projects are already exploring these ideas, such as CSCL's FLE3.

That is not to say that the software development analogy isn't without its deficits. Although open source software is gaining adoption, the finer points of the development model are not necessarily as widely known. Therefore, some of the information communicated through borrowing software development terminology is not necessarily understood. Another disadvantage is that the more formalized the OER development process, the higher likelihood it will stifle creativity.

2.2 Issue Tracker

If open texts are developed using open source software development models, then it is only appropriate that some of the support mechanisms used in open source software are used by open texts. One of the most common features of an open source software program is a bug tracker. A bug tracker is web page or web application in which problems with are listed. Typically they include some type of status indication so anyone can see if a bug is in the process of being fixed. There are multiple programs that automate the bug tracking process for developers, many of them open source themselves.

For the OER Handbook it was necessary to create an issue tracker. Mediawiki software always has a "Talk" page accompanying a content page to allow for discussion, debate and coordination of a page. But as the OER Handbook was being developed issues arose that applied to multiple pages, necessitating a place in which they could be discussed on a single page. An 'issue tracker' was created. It was a single page on the Table of Contents in which anyone could submit issues that span multiple pages within the document. A single issue consisted of the following parts:

- **Issue:** a brief 3-10 word description of the problem.
- **Submitter:** who is placing the issue in the tracker
- **Date submitted:** when the issue was first put in the tracker (though relevant discussion may have already occurred).
- **Date Completed:** when the issue was resolved.
- **Comments:** brief elaboration on the problem as well as any comments. Writers starting using the automated signature feature next to their comments to clarify who is making them and provide a time-stamp.

12 <http://en.wikipedia.org/wiki/GFDL>

13 <http://creativecommons.org>

Merge File Format Sections [edit]

Submitter: Ktucker 14:43, 29 June 2008 (UTC)
Date Completed:
Comments:
 As [suggested here](#) - alternative version ready to paste.
 Changes adopted with some minor tweaks. Issue closed. --Sgurell 16:31, 25 July 2008 (UTC)

User Stories [edit]

Submitter: Ktucker 23:28, 22 June 2008 (UTC)
Date Completed:
Comments:
 There are so many great contributions! Do we include all of them? If so, do we categorise them somehow? (actual stories, tips, ...). Also, should we eliminate "User?" --Sgurell 15:17, 23 June 2008 (UTC)
 I've changed the name of each section to "perspectives." --Sgurell 18:37, 23 June 2008 (UTC)
 Issue closed? --Sgurell 04:04, 26 June 2008 (UTC)
 "Perspectives" is good - move when changes made Ktucker 03:28, 1 July 2008 (UTC)
 Added thoughts to future edition suggestions page. --Sgurell 16:22, 25 July 2008 (UTC)

Issue: Name for All Rights Reserved repository [edit]

Submitter: Ktucker/Sgurell
Date Submitted: 25 Jun.
Date Completed:
Comments:
 Ktucker raised this issue, citing the problem with implying open licenses don't have rights. Suggested "permission" or "restricted access." Sgurell recommends "full copyright" or "traditional copyright." Term agreed upon should be fairly short to accommodate titles and minimal disruption of paragraphs that mention it.
 Restricted Use Repositories Ktucker 01:40, 1 July 2008 (UTC)
 Traditionally Copyrighted repositories chosen for Version 1.0, though this may change in Version 2.0. --Sgurell 15:19, 25 July 2008 (UTC)

Issue: Difference between Composing and Adapting OER [edit]

Submitter: Sgurell
Date Submitted: 16 Jun.
Date Completed:

Illustration 3. Screenshot of Issue Tracker.

The formatting of the issue was not uniform, but similar. Some submitters chose to simply use the time-stamp in their signature rather than a Date submitted field. However, the overall idea was the same. Each section was separated by a horizontal lines. Shortly after the issue tracker was created it was separated into two sections: one for 'open' or outstanding issues, and one for 'closed' or completed section. By using headings for the issue field, a hyperlinked table of contents was automatically generated.

Though the issue tracker was help in project management, it was slightly frustrating to add an issue. Each issue, included formatting, needed to be manually edited into the page. Occasionally, a line break or link would be forgotten. Mackintosh, the head of WikiEduator, purposed automating the process through the use of a form. Unfortunately, there was insufficient time and resources to create the form during the development of the first version of the OER Handbook.

2.3 Future Editions

Besides a bug tracker, another common feature in open source software development is the feature request. A feature request is a suggestion for functionality to be added to a program. In the case of the OER Handbook, a future editions page was added to capture potential improvements for future editions of the handbook.

With open source projects the feature request guides future development. But feature requests also provide an equally important function: prioritization of issues, problems and concerns. Prior to the addition of the future editions page there was pressure on project participants to solve every disagreement and concern before creating the first print version. Once participants took a more procedural, and less product oriented view of the OER Handbook, issues began to be

divided into what could be done for the first version and what should be done in subsequent versions.

It is important to note that there is a difference between the feature request and future edition suggestions. Feature requests, in general, are very incremental suggestions, such as interoperability with a particular website or service. Future edition requests may involve very drastic changes to an OER. For example, one of the contributors to the OER Handbook proposed several significant changes to the OER life cycle that governed the organization of the handbook. Though these changes were interesting and brought up good points about OER development, implementing these changes in time for the first version of the handbook would have been very difficult. Instead, changes that can be easily implemented are incorporated into the current version of the handbook and rest are put in the future editions page, as well as links to relevant discussion and sources.

3 Best Practices

These practices can be generalized to other OER development. One of the best practices in OER is to have intermediate goals, such as a version 1.0. Setting realistic goals is important, as creating OER in addition to other responsibilities can be very demanding. Occasional milestones also provide opportunities to reflect and determine the future direction of the OER. When creating these goals it is important to include volunteers who contribute to the OER, as they will want to inform any decisions that take place. It is also advantageous to keep the development process as open as possible. As the OER Handbook was being developed, people across the world took note of the project and found that it informed their own endeavors while simultaneously encouraging the author.

4 The Next Generation of Tools

Though elements of the wiki have been extremely beneficial to the development of the OER Handbook, development was not without difficulties. For example, the authors found difficulty with ensuring naming consistency between page titles, URLs and template names. Wikis tend to support a flat page hierarchy, meaning any page is accessible from any other page, and there was some disagreement about how to organize the pages. Linking across deeply nested pages can be confusing, at least to the beginner. Image and table manipulation also has a learning curve. Pages, if incorrectly moved, do not keep their reversion history. RSS¹⁴ and Atom¹⁵ standards, because of security, are difficult to implement. None of these challenges suggest that wikis cannot be used for development. Indeed, the success of WikiEducator, Wikiversity and Wikibooks would indicate that wikis can be used to develop OER successfully.

But given the difficulties of working with wikis, it may be time to evaluate what kinds of tools are needed. Mediawiki software, because it is open source, can be modified and adapted to address some of these concerns, and plugins can be added to provide additional functionality. However, adding plugins to any software is known to reduce stability and coordinating functionality across plugins can be demanding.

Therefore, an entirely new software tool, specifically designed for OER, may be warranted. Such a tool would have the advantage of being built with the best practices and needs of previous OER projects. Given the wide variety of contexts in which OER are designed for, software management would need to be capable in handling a wide variety of organizations and writing styles.

In open source software, one of the most popular programs for managing different versions of software is Subversion¹⁶. Subversion was created by CollabNet, Inc.¹⁷ in 2000. It was meant to

14 <http://en.wikipedia.org/wiki/RSS>

15 [http://en.wikipedia.org/wiki/Atom_\(standard\)](http://en.wikipedia.org/wiki/Atom_(standard))

16 <http://subversion.tigris.org/>

succeed CVS, which was the popular version control program at the time. CVS, though popular, had limitations and problems, necessitating a replacement (Collins-Sussman, Fitzpatrick, & Pilato, 2007). Subversion allows for flexibility in file organization and handles both text and image files. Text files, usually the source code, can be compared and merged. Dozens of people can contribute to a single project in a productive manner, but repositories are generally open access, allowing anyone to view or download.

OER are already in development using Subversion. KnowledgeForge¹⁸ hosts several projects, some of which are designed to create open texts. However, this approach is not perfect. In order for a Subversion-like open text development tool to be successful, it needs to improve on subversion and have a simple user interface. Subversion can be used in two ways: at the line-command (similar to the MS-DOS prompt of computers from the 1980s and early 1990s), or through GUI¹⁹. The GUI is usually part of IDE, or Integrated Development Environment²⁰, such as Eclipse²¹. Unfortunately, even these interfaces can be confusing for beginners. Problems configuring Subversion in the first place can be difficult as well as committing changes to the repository. A WYSIWG (What You See Is What You Get) interface, similar to a word processor, would be preferable to learning a markup language. Initial installation may be unavoidably complex, but changing the repository should be fairly straightforward. Other improvements would have to be made as new practices and preferences develop. Although it does not incorporate all of the features, the web application Beanstalk²² is a step towards this type of tool.

Finally, in addition to a simpler development process, themes should be customizable. Though potentially infinite customization is possible of MediaWiki software, the resulting book should be able to be displayed as a web page, through custom themes, much like a WordPress template²³.

5 Conclusion

The future success of open text development will be based on its ability to derive the best tools and practices of “Web 2.0” technologies. Open source software development has intriguing parallels that offer a perspective on how collaborative projects can be successfully managed and coordinated. Although it is important to note that creativity should still be considered a part of the process. By improving on these processes, and contextualizing them for open education, an important part of achieving critical mass will be reached.

References

- Collins-Sussman, B., Fitzpatrick, B.W. & Pilato, M.C. (2007). What is Subversion? *Version Control with Subversion*. Retrieved June 29, 2008 from <http://svnbook.redbean.com/en/1.4/svn.intro.whatis.html>
- Gurell, S. (2008). Defining OER. *The OER Handbook Version 1.0*. Retrieved August 31, 2008, from http://www.wikieducator.org/OER_Handbook/educator_version_one/Introduction/Defining_OER
- Mackintosh, W. (2008). 4,000+ users; congratulations on another milestone, retrieved June 29, 2008 from http://groups.google.com/group/wikieducator/browse_thread/thread/8138d4e80c57b74e?hl=en
- Mackintosh, W. (2007). WikiEducator: Memoirs, myths, misrepresentations and the magic, retrieved June 29, 2008 from <http://blog.worldcampus.psu.edu/index.php/2007/04/04/wikieducator/>

17 <http://www.collab.net/>

18 <http://www.knowledgeforge.net/>

19 <http://en.wikipedia.org/wiki/GUI>

20 <http://dictionary.reference.com/browse/interactive%20development%20environment>

21 http://wiki.eclipse.org/FAQ_Where_did_Eclipse_come_from%3F

22 <http://www.beanstalkapp.com>

23 <http://wordpress.org/extend/themes/>

Stallman, Richard. (Last updated 2008 June 28). The Free Universal Encyclopedia and Learning Project. Retrieved June 29, 2007 from <http://www.gnu.org/encyclopedia/free-encyclopedia.html>

Tucker, K. (2007). FLOSS, OER and Digital Inclusion, retrieved June 29, 2008 from <http://blog.worldcampus.psu.edu/index.php/2007/05/02/floss-oer-equality-and-digital-inclusion/>

Tucker, K. (2008). Personal communication.

Wiley, D. (2008 January 17). OER Handbook. Iterating Towards Openness. Retrieved August 31, 2008 from <http://opencontent.org/blog/archives/440>

Wiley, D. (2008 January). Personal communication.

Author



Seth Gurell
Graduate Assistant
Center for Open and Sustainable Learning, Utah State University
sethgurell@gmail.com

Copyrights



The texts published in this journal, unless otherwise indicated, are subject to a **Creative Commons Attribution-NonCommercial-NoDerivativeWorks 2.5 licence**. They may be copied, distributed and broadcast provided that the author and the e-journal that publishes them, eLearning Papers, are cited. Commercial use and derivative works are not permitted. The full licence can be consulted on <http://creativecommons.org/licenses/by-nc-nd/2.5/>

Edition and production

Name of the publication: eLearning Papers
ISSN: 1887-1542
Publisher: elearningeuropa.info
Edited by: P.A.U. Education, S.L.
Postal address: C/ Muntaner 262, 3º, 08021 Barcelona, Spain
Telephone: +34 933 670 400
Email: editorial@elearningeuropa.info
Internet: www.elearningpapers.eu