

# Learning by Teaching through Collaborative Tutorial Creation: Experience using GitHub and AsciiDoc

Jim Marquardson  
jimarqua@nmu.edu  
College of Business  
Northern Michigan University  
Marquette, MI 49855, USA

Ryan M. Schuetzler  
rschuetzler@unomaha.edu  
College of Information Science & Technology  
University of Nebraska at Omaha  
Omaha, NE, 68182, USA

## Abstract

Learning by teaching is a pedagogical technique that encourages mastery of a topic by having students teach each other. Student presentations, group discussions, and face-to-face teaching are frequently used to have students teach their peers. In this paper, we describe the use of a novel assignment to implement learning by teaching. In a course assignment, learners created a tutorial using the AsciiDoc markup language. The tutorials were uploaded to a Git source control repository on the GitHub platform and combined into a single electronic book. Students were asked to complete their peers' tutorials and provide constructive feedback. The assignment had several goals. First, students would master the topic chosen for the tutorial. Also, students would gain experience creating their own learning plans to master the topic. Next, students would learn about source control and markup languages. Finally, students would publish the resources to make the tutorials publicly available to contribute to the existing corpus of open educational resources. A survey was conducted after the final assignment submission. Results from a reflection survey show that students generally favored the assignment and found it to be a useful learning experience despite some challenges working with the technology stack. Experiences from the instructor's point of view are shared to provide guidance for implementing this type of exercise effectively.

**Keywords:** Innovation in pedagogy, Tutorial creation, Collaborative editing, Source control, Markup languages

## 1. INTRODUCTION

Lifelong learning is imperative in the information systems field given the rapid pace of change. Because students have limited time in academia, it is important that they become self-sufficient

learners. Unfortunately, many people simply do not know how to learn despite years of schooling (Argyris, 1991). Educators must give students opportunities not only to learn, but to internalize the skills necessary to continue their education after formal schooling.

Active learning occurs when students are “doing things and thinking about what they are doing. (Bonwell & Eison, 1991, p. iii)” Learning by teaching is one example of active learning. In learning by teaching, students must research a topic and present information to share with others. Students gain a deeper understanding of the topic of their study through a learn by teaching method than many traditional methods (Stollhans, 2016). A side benefit of using learning by teaching is that students can help create educational resources that can be made available to others. Information systems researchers can be at the forefront of investigating the use of information technology to aid in the learn-by-teaching paradigm.

In recent years, educators have seen the growth of open educational resources (OER) (Masterman, Wild, White, & Manton, 2011). These freely available tools include lecture materials, textbooks, exercises, interactive simulations, and many other resources for improving student access to educational resources.

The current study addresses several research questions. First, is tutorial creation an effective implementation of the learning by teaching approach. Second, are open source tools a good fit for implementing tutorial creation? And third, does learning a new topic with an unfamiliar technology stack impede the learning process?

In this paper, we describe an assignment in which students learn about a topic, then use open source tools to create a tutorial to teach their peers. After review, the tutorials were then published as OER and made publicly available. In future semesters, these same materials can be used as in-class exercises for other courses. All materials are released under a Creative Commons license, allowing others to use them. Through this assignment, students gain a deeper understanding of their assigned topic, and learn a variety of concepts including version control, collaboration, and a markup language.

The pedagogical foundation for the assignment is given in the next section. Following is a detailed overview of the class assignment. Next, quantitative and qualitative student feedback is presented. Observations and recommendations from the instructor follow.

## 2. PEDAGOGICAL FOUNDATION

The traditional classroom lecture has been a target for criticism (Folley, 2009). An unknown author is quoted as saying, “Lecturing is the

transference of the notes of the lecturer to the notes of the student without passing through the brains of either” (Exley & Dennick, 2004, p. 3). Educators have been challenged with developing new techniques to ensure that students learn effectively. Some techniques tweak the traditional lecture by adding active learning elements, such as clickers (Martyn, 2007). In this model, the instructor is still the expert with knowledge that must be transferred to students. We believe that educators should embrace the role of guides rather than sages (King, 1993). Part of being a guide is ensuring that students have the skills to learn independently.

### Self Sufficient Learning

It can be challenging to encourage students to take ownership of their own learning. For many students, doing simply what they are asked has been so ingrained that they are fearful of submitting work that does not meet an instructor’s specific expectations. Others prefer being given clear step-by-step instructions to complete activities. Hershatter and Epstein (2010, p. 216) note that “elements of ambiguity, or any project or exam that requires [students] to work without guidelines, templates, or examples, results in a great deal of angst, because they have not had much practice producing without explicit instructions, well defined criteria for success, and specific deadlines set by others.” Students struggle with ambiguity, so being given freedom to explore topics in a self-directed way can be uncomfortable.

The transition to more self-directed study can lead to many positive outcomes. For example, students who prepared video materials to present to peers were more responsible, motivated, and participatory (Assinder, 1991). Students must practice finding resources and learning on their own because the field of information systems changes rapidly. If students learn how to learn in the classroom, they will be in a better position to keep up with changes in the industry. Students must learn how to break down complex topics, find resources, develop learning plans, and reflect on their learning achievements.

### Learning by Teaching

The formal approach to learning by teaching was developed in the 1980s (Stollhans, 2016). Though initially created for teaching languages, the principles of learning by teaching apply to other disciplines. With the learning by teaching approach, students must think critically about how they will structure learning for somebody else to understand a topic (Grzega & Schoner, 2008). Thus, students must be actively engaged

in thinking critically about the topic and should therefore achieve proficiency in the topic.

Peer-to-peer teaching takes place in the classroom in many forms. Peer editing, presentations, and group discussions allow students to teach their peers. Tutorial creation could be an effective method for helping students learn by teaching. We define a tutorial as a self-contained lesson that guides a learner through a series of activities to teach specified learning objectives. Tutorials incorporate hands on learning by requiring the learner to perform tasks. Writing a tutorial requires that the tutorial creator understand the topic deeply enough to break down instructions in a clear, logical manner so that a peer could successfully complete the activity.

### **Open Educational Resources**

Open educational resources have grown in acceptance and availability over the past decade. These resources are characterized by legal rights of the "5 Rs" of OER: to retain, reuse, revise, remix, and redistribute (Wiley, n.d.). These open resources improve student access to course materials (Caswell, Henson, Jensen, & Wiley, 2008), and can possibly improve learning outcomes.

OER has several advantages over proprietary learning materials such as traditional textbooks. The first and most obvious benefit is cost (Bliss, Robinson, Hilton, & Wiley, 2013). Students and instructors can access OER resources for free, reducing the cost of education for students. A second benefit is improved accessibility. Because they are freely available, OER can often be embedded directly with other course materials, making it easier for students to find the resources they need.

Another benefit of OER is the potential for student contributions. These contributions can take many forms. Because of the OER principles of revise, remix, and redistribute, anyone can take an open resource and improve upon it. Such contributions for an open textbook could include updating with more current information, correcting errors, or adding new chapters or study materials. Because of the low barrier to entry, students can also create their own OER in areas where they have developed expertise.

OER is most frequently published under a legal license known as Creative Commons (<https://creativecommons.org/licenses/>). The Creative Commons (CC) licenses offers options with varying levels of restrictiveness. All CC

licenses require acknowledgement of the original work. The most open license, CC BY, allows anybody to reuse, remix, build upon, or even sell versions of the original work. By contrast, the most restrictive license, CC BY-NC-ND, allows reuse with attribution, but remixing, revising, or selling the work is not allowed. Other licenses offer different subsets of permissions.

### **Killing Multiple Birds with One Stone**

Because of the breadth of material that instructors want students to master, it is sometimes necessary to combine multiple learning objectives into a single assignment. However, combining multiple learning outcomes in a single assignment could cause confusion as struggles with one learning objective could negatively impact others. For example, a novice system analyst might struggle to learn to create work breakdown structures using Microsoft Project, being neither an expert in project management nor Microsoft Project. The learner would not know if the inability to master work breakdown structures was due to lack of proficiency with the software or a misunderstanding of project management methodology. Alternatively, combining multiple learning objectives may be a way for instructors to cover the ever-increasing body of knowledge required in industry. Perhaps it is only by combining multiple learning objectives that students can truly be prepared to succeed with the breadth and depth of knowledge required in the field. The efficacy of combining learning objectives in a single assignment should be empirically investigated.

### **Open Questions**

To our knowledge, the effectiveness of using collaborative tutorial creation has not been evaluated in the information systems paradigm. Because information systems students are well positioned to use advanced technical tools, we aim to evaluate the effectiveness of using tutorial creation to teach students not only about the topic of their tutorials, but also about the underlying technology stack used to create the tutorials. It is hoped that instructors can make effective use of class time by including multiple learning objectives in a single assignment while not having ambiguity in any of the learning objectives undermine the learning process of unrelated learning objectives. Finally, student perceptions of publishing their work as OER resources should be evaluated to determine if it could be a motivating factor.

In the following section, we describe some of the tools that can be used to facilitate collaborative tutorial creation.

### 3. FACILITATING TECH TOOLS

This project makes use of several complementary technologies to facilitate eBook creation and collaboration. The primary tools are the AsciiDoc markup language and its associated tools, Git for version control, and GitHub for collaboration.

#### AsciiDoc

AsciiDoc is a document markup language tailored for writing articles, books, web pages, and more (Rackhman, 2018). Because AsciiDoc uses plaintext files, the source files can be tracked and maintained in a source code version control system. Each line of text would be treated as if it were a line of code in a computer program.

Compared to markup languages such as XML and HTML, AsciiDoc requires much less markup and is easier to learn. AsciiDoc provides additional features to make more complex documents possible than with the popular Markdown format without adding significant markup to the document. It is designed so that the source files are human-readable, while still having the necessary machine-readable formatting to be compiled into a PDF or HTML document when desired. Figure 1 demonstrates a simple AsciiDoc document with headings, lists, and embedded images.

```
= Document Title

Introductory paragraph.

== Introduction Subheading

Another paragraph.

* Bulleted list item
* Bulleted list item
+
image::screenshot1.png[]

== Conclusion

. Numbered list item
. Numbered list item

image::screenshot2.png[]
```

Figure 1: Sample AsciiDoc Document

Once a document has been written in AsciiDoc, it can be compiled into other formats for sharing. The most popular tool for compiling AsciiDoc is

AsciiDoctor (Allen, White, & Waldron, 2018). The Ruby programming language and the AsciiDoctor-PDF package can be used to compile AsciiDoc to PDF. First, the Ruby programming language must be installed. The Ruby executable must be accessible in the system path so that it can be invoked from the command line. A tutorial can be compiled from a command prompt. The compiler lists any errors if the document does not compile correctly.

#### Source Code Control

Git is a distributed version control system initially created by Linus Torvalds to manage the Linux kernel's source code. Like most version control systems, Git is primarily used for the management of application source code, but it can be used effectively for any project that needs to track changes within files. With Git, users are not required to formally check out files before making changes. Instead, users clone a repository, make changes locally, commit changes locally, then push changes to the remote repository. Two people can make changes to the same files and Git will integrate those changes seamlessly if the changes do not conflict. If two people edit the same part of a file, a merge conflict will occur, and the user must determine how the conflict should be resolved.

#### GitHub

GitHub is an online Git service that hosts Git repositories and provides other features such as issue tracking. GitHub has become the prominent platform for hosting open source projects (Metz, 2015). Microsoft shut down its CodePlex source control service and Google closed its Google Code source control service with both companies choosing to adopt GitHub for their open source projects (Weinberger, 2017). Microsoft recently bought the GitHub platform (Bass & Newcomer, 2018). For programmers, some have gone as far as to say that a GitHub profile is more important than a resume (Weiss, 2012). Clearly, GitHub is a platform that information systems students should be familiar with.

GitHub created a graphical client for interacting with source code repositories. While the client can be used with any Git repository, it integrates seamlessly with GitHub. Using the graphical client, users can clone a repository, commit their changes locally, and push those changes to the remote repository without having to use Git from the command line.

In the next section, we describe how we used these tools in a collaborative tutorial assignment

and how we measured the assignment effectiveness.

#### 4. METHODOLOGY

Data was collected at a midwestern university. A collaborative tutorial assignment was embedded in an undergraduate cyber security course. In total, 9 students (2 female, 7 male) completed the assignment and a follow-up survey.

Students were asked to create a tutorial that would be combined with their peers' tutorials to create a course eBook. The tutorial topics were required to be in the general domain of network security. With regards to tutorial difficulty, students were told to target an audience that had completed an introductory computer networking course.

##### Assignment Learning Objectives

There were several learning objectives. First, students would develop self-sufficient learning. Students chose their own topics, found resources (such as online tutorials), and established their own learning objectives.

The second learning objective was to have students learn the topic of their tutorial. Students were encouraged to pick topics such as network monitoring, scripting, or other topics that required hands-on use of software. Writing the tutorial would require that students grasped their topic sufficiently that they could teach somebody else about the topic through their tutorial.

The last learning objective was to introduce key technology tools used in information systems. As explained previously, Git and GitHub are popular tools for managing source code. AsciiDoc is a plaintext markup language that would work well in a source code repository. Students were also introduced to the Ruby programming language. Students would need to use the command line to test compile their tutorials.

##### Assignment Flow

Because none of the students in class had prior experience with Git, GitHub, or AsciiDoc, the introduction to the assignment included an overview of these technologies. Principles of version control systems were addressed. Students created GitHub accounts, installed the GitHub client, and cloned the course repository.

The instructor created the initial eBook source code repository. Students were provided with their own directory which contained a template AsciiDoc file pre-filled with suggested headings

and sample markup. The sample markup included the most commonly used formatting items: bulleted lists, numbered lists, section headers, and images. We designed the template to be a sufficient resource that contained examples of all the basic AsciiDoc markup students would need to complete the assignment. The code for the entire eBook was in a single source code repository where the instructor and students all shared read/write access.

Students then installed Ruby and the AsciiDoctor-PDF Ruby package. Once the technology was in place, students could edit drafts of their tutorials, compile, and interact with the shared source code repository.

In a typical editing session, students would pull the latest code from GitHub, make changes on their local machines, commit the changes to the master source code branch, and push the changes to the central GitHub repository. On the last class period of the semester, students were asked to come to class with a complete draft committed to GitHub. Students would spend the last class period following pulling their peers' tutorials and providing feedback and corrections. Students were encouraged to open issues on GitHub to track problems they identified. Though all students had commit access to the entire eBook repository, students were encouraged not to edit their peers' files but instead to only commit changes to their own tutorial. If they desired, they could create pull requests to suggest changes to the original author. GitHub's revision tracking allows the instructor to see who is making changes to which files, and when.

##### Feedback

The tutorials created in the exercise were designed to be subjected to two levels of peer evaluation. First, peers in the class review the tutorial by following the instructions created to learn about the topic presented. From this feedback, the students can identify areas that need clarification or expansion.

The second type of feedback that can be provided to students comes from the open published nature of the tutorials. All tutorials were merged into a publicly available, Creative Commons-licensed GitHub repository and published on the Web. All students gave permission for their work to be published under the Creative Commons – Share Alike license. In the GitHub repository, potential contributors could offer suggestions and feedback, or even contribute changes directly through GitHub's pull requests.

## Survey

After the final tutorial submission, students were sent a link to a survey to have them reflect on the experience. Students were told that their answers to the reflection would not impact their grade on the assignment. To ensure completion, the surveys were not anonymous. A mix of quantitative and qualitative data was gathered to develop deeper insight into the experience (Venkatesh, Brown, & Bala, 2013). The complete list of survey questions can be found in Appendix B.

## 5. RESULTS

Results from the student survey will be given. Following is a reflection from the instructor's perspective.

### Quantitative Survey Analysis

Students responded to quantitative survey questions using 7-point Likert scales with ranges from strongly agree (1) to strongly disagree (7).

First, attitudes toward the technology tools were assessed. Overall, students enjoyed using AsciiDoc to create the tutorial ( $M=1.56$ ,  $SD=1.01$ ). Likewise, they felt that GitHub is an effective tool for managing collaborative writing ( $M=1.44$ ,  $SD=0.73$ ). Git was reported to be an effective tool for merging versions of a project ( $M=1.56$ ,  $SD=0.73$ ).

Students generally did not feel that they spent a lot of time learning Git, GitHub, and AsciiDoc to a point that the tools distracted from learning the topic of their tutorial ( $M=4.67$ ,  $SD=1.94$ ). The large standard deviation indicates that some students struggled more than others. Most students disagreed with the statement that the tools (Git, GitHub, AsciiDoc) were difficult to work with which degraded the quality of their finished tutorial ( $M=5.44$ ,  $SD=1.13$ ).

Knowing that their work would be made public was a motivating factor for producing high quality tutorials, but not to a large extent ( $M=3.22$ ,  $SD=1.79$ ). Student peer review was motivating to a similar degree ( $M=3.22$ ,  $SD=1.39$ ).

Students were asked how the tools helped or hindered them in creating their tutorial using a 7-point Likert scale from helped a lot (1) to hindered a lot (7), with an option of "not applicable" if they did not use the tool. The most helpful tools were GitHub ( $M=1.33$ ,  $SD=0.5$ ), Git ( $M=1.67$ ,  $SD=0.71$ ), and GitHub pull requests ( $M=2.00$ ,  $SD=1.32$ ). Less helpful tools or features were AsciiDoctor ( $M=2.67$ ,  $SD=3.04$ ) and Ruby

( $M=4.33$ ,  $SD=3.04$ ). GitHub issues were rated the least helpful ( $M=5.38$ ,  $SD=2.56$ ), but it should be noted that no student used the issues feature for this assignment.

Students were asked about their preference between different types of learning activities compared to the collaborative tutorial. A forced choice was used with options to strongly prefer, slightly prefer, or indicate no preference. A value of 2.5 indicates a strong preference for the collaborative tutorial assignment. A value of -2.5 indicates a strong preference for the alternative learning activity. A stronger preference toward the collaborative tutorial assignment was observed when compared against PowerPoint ( $M=0.61$ ,  $SD=1.17$ ) and reading articles ( $M=0.94$ ,  $SD=1.24$ ). A very small preference toward the collaborative tutorial assignment was observed when compared against following written exercises ( $M=0.06$ ,  $SD=1.24$ ) and watching videos ( $M=0.06$ ,  $SD=1.33$ ). Class discussions were preferred over the collaborative tutorial ( $M=-0.83$ ,  $SD=0.71$ ).

Figure 2 shows a comparison of the collaborative tutorial assignment versus other learning activities. The chart shows bars starting from 0 (no preference) to the calculated mean along with the standard deviation.

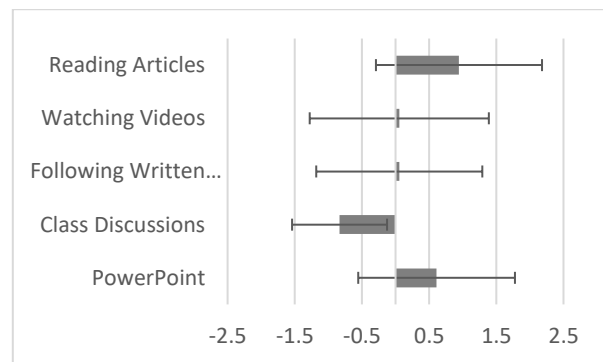


Figure 2: Comparison of Learning Activity Preference

### Qualitative Survey Analysis

First, students were asked what parts of the assignment they enjoyed. Half of the respondents specifically mentioned AsciiDoc. They enjoyed learning the markup language and rendering their tutorials as PDFs. One student mentioned that he will use the same technology stack to write tutorials and documentation going forward. Only one student specifically mentioned Git, which is surprising given how useful most students rated it. One student reported, "I enjoyed getting to pick a topic and having to understand it enough

to explain it to others via a written tutorial. The tutorial writing itself was surprisingly not difficult." Another student remarked, "Being able to have hands on experience made my learning experience better."

Students were asked to respond to the question, "What parts of the assignment were most challenging?" Two students reported that picking a topic was the most challenging part. Several students had challenges with different parts of the technology stack. AsciiDoc, GitHub, Ruby, and Git were all mentioned as being challenging to work with. However, no single tool or feature was repeated by more than two students. One student remarked, "The most challenging parts were learning a different format and applying it on the spot. But that was what made it good."

Students were asked what changes they would make to the assignment to make it a better learning experience. Two students would have preferred a more in-depth explanation of using Git and GitHub. One student recommended providing more assistance in picking a topic. Another student would have liked an example of a completed student-created tutorial. One student would have appreciated more peer review.

Students were given an opportunity to optionally provide any additional feedback. Several students recommended that the technology stack and assignment be introduced earlier in the semester to alleviate end of semester time pressures. One student wished there were more time to spend working through other students' tutorials. One student stated, "This assignment was awesome and practical. I probably never would have known about how Git works or what a fork is without this course."

## 6. DISCUSSION AND LESSONS LEARNED

Overall, the assignment was a success based off student evaluations and the instructor's observations.

### Use of Tools

Students responded well to the inclusion of multiple learning objectives in a single assignment. They were able to master several different tools while mastering a new topic of their choosing.

Even though some students in the class were computer science majors, no student had any experience with Git or GitHub. Despite the prevalence of these technologies in industry,

most students do not seem to use them outside of class assignments.

None of the students had used Ruby or AsciiDoctor. While the template provided to the student along with course instruction was intended to be enough guidance, several students reported spending hours debugging AsciiDoctor-PDF compiler errors. Some of the solutions to the compilation problems could be fixed by simple syntax fixes, but for students new to the markup language it took a significant amount of time to troubleshoot. One student copied a Unicode character from a web browser into the text file which caused a compilation error.

The GitHub issues feature and pull requests were briefly introduced. However, class members did not take advantage of them. The only person to open issues was the instructor. Likewise, students did not take advantage of pull requests—only the instructor issued them. Pull requests and branches are more advanced features of version control systems. Students would have needed more experience with Git to feel comfortable using these features. Future assignments could use pull requests to allow students to request that their changes be merged into the main repository, rather than giving direct access.

### Pedagogy

While pedagogy is not the primary focus of a networking security class, basic pedagogical instruction should have been addressed. Students were never informed about instructional best practices to the detriment of the final tutorial quality. In the future, further emphasis should be placed on defining learning objectives, creating activities that directly support the learning objectives, and developing quality reflection and assessment activities.

### Tutorial Quality

The tutorials provided evidence of learning, but it is unlikely that the entire eBook will serve as a resource for other network security classes. Some chapters provided clear learning objectives, focused exercises, and strong reflection questions. Other tutorials were too broad in their learning objectives such that it would have been difficult to create a focused tutorial to reach those objectives. The tutorials can be found on GitHub at [https://github.com/jimmarq/wildcat\\_tutorials\\_volume\\_1](https://github.com/jimmarq/wildcat_tutorials_volume_1).

The instructor rated the quality of the final tutorials on professionalism, completeness, and accuracy. Results were mixed. One student

included several copyrighted images without proper attribution. The instructor asked the student to remove the images. After the student failed to comply, the instructor removed them from the student's submission. In the future, instruction will be provided regarding copyright and licensing for using others' work. Another student clearly struggled with AsciiDoc syntax. The instructor ended up fixing the syntax after the final submissions had been graded. It may have been feasible after the course finished to have the student correct his own work, but at that point there was no extrinsic motivation to compel any further edits.

### Publication Considerations

Some open considerations relate to the open nature of the work. Most course assignments are submitted on a private learning management service and are never exposed to the public unless the students specifically make their work public independently. However, in this assignment the work was being published directly on an open platform for the world to see. In a private environment, copyright mistakes might result in a grade penalty and a warning from the instructor, but consequences could be more severe for public work. Instructors must take greater care to ensure that students are complying with copyright laws and have cited work appropriately. This is especially true when instructors choose to list themselves as editors. It would be advisable to create the assignment first in a private GitHub repository, then make it public only after checking for quality and compliance.

Instructors implementing this assignment must also consider what happens to the course eBook once it has been finalized. If a typo is found at some point in the future, students should know if they are responsible for fixing it. Ideally, individual editors would monitor issues raised on their own work. If not, the instructor may be taking on an ever-increasing burden if implementing this assignment frequently or with many students.

Going forward, one compelling extension of this assignment would be to have future classes edit or expand created in past semesters. This would  
Allen, D., White, S., & Waldron, R. (2018).  
*AsciiDoctor*. Retrieved from  
<https://asciidoctor.org/>

Argyris, C. (1991). Teaching smart people how to learn. *Harvard Business Review*, 69(3).

Assinder, W. (1991). Peer teaching, peer learning: one model. *ELT Journal*, 45(3),

help students engage with their peers, become more familiar with source control, and encourage them to continue contributing to OER.

## 7. CONCLUSIONS

Student-created tutorials can complement other learning activities to help students master new concepts. Responses to survey questions indicate that students enjoyed creating tutorials. A technology stack using AsciiDoctor, Git, and GitHub was used to facilitate collaboration and tutorial submission. Though most students experienced at least one problem getting the toolchain to work correctly, all students were able to successfully use the tools. While learning the topic of their tutorials, students learned how to use the AsciiDoc markup language and Git version control. This assignment shows one way to cover multiple learning objectives in a single assignment.

We learned several lessons while implementing the assignment. In the future, we would make the following changes. First, each piece of the technology stack must be explained sufficiently because few students will have prior experience with the tools. Second, intermediate due dates should be created to ensure that students are comfortable with the technology, pick an appropriate topic, and have time for peer review. In addition to technology, students should be instructed on pedagogical best practices, such as creating effective learning objectives and designing activities to support those learning objectives. Students should also be reminded about citing sources and copyright.

The students who completed this OER tutorial creation assignment learned to use a variety of tools. While cementing their understanding of a networking-related topic, students used version control, markup, and programming collaboration tools to create a resource available to future semesters. These materials can now be used to provide hands-on practice in future classes.

## 8. REFERENCES

- 218–229.  
<https://doi.org/10.1093/elt/45.3.218>
- Bass, D., & Newcomer, E. (2018, June 4). Microsoft Buys GitHub for \$7.5 Billion, Going Back to Its Roots. Retrieved June 11, 2018, from  
<https://www.bloomberg.com/news/articles/2>



- 018-06-04/microsoft-agrees-to-buy-coding-site-github-for-7-5-billion
- Bliss, T., Robinson, T. J., Hilton, J., & Wiley, D. A. (2013). An OER COUP: College Teacher and Student Perceptions of Open Educational Resources. *Journal of Interactive Media in Education*, 1–25.
- Bonwell, C. C., & Eison, J. A. (1991). *Active learning: creating excitement in the classroom*. Washington, DC: School of Education and Human Development, George Washington University.
- Caswell, T., Henson, S., Jensen, M., & Wiley, D. (2008). Open Content and Open Educational Resources: Enabling universal education. *International Review of Research in Open & Distance Learning*, 9(1), 1–11.
- Exley, K., & Dennick, R. (2004). *Giving a lecture from presenting to teaching: Key Guides for Effective Teaching in Higher Education*. London: RoutledgeFalmer: Taylor and Francis Group.
- Folley, D. (2009). The lecture is dead long live the e-lecture. In *8th European Conference on E-Learning, University of Bari, Italy* (p. 204).
- Grzega, J., & Schoner, M. (2008). The didactic model LdL (Lernen durch Lehren) as a way of preparing students for communication in a knowledge society. *Journal of Education for Teaching*, 34(3), 167–175. <https://doi.org/10.1080/02607470802212157>
- Hershatter, A., & Epstein, M. (2010). Millennials and the World of Work: An Organization and Management Perspective. *Journal of Business and Psychology*, 25(2), 211–223. <https://doi.org/10.1007/s10869-010-9160-y>
- King, A. (1993). From sage on the stage to guide on the side. *College Teaching*, 41(1), 30–35.
- Martyn, M. (2007). Clickers in the classroom: An active learning approach. *Educause Quarterly*, 30(2), 71.
- Masterman, L., Wild, J., White, D., & Manton, M. (2011). The impact of OER on teaching and learning in UK universities: Implications for Learning Design. In L. Cameron & J. Dalziel (Eds.), *Proceedings of the 6th International LAMS and Learning Design Conference* (pp. 135–144). Sydney: LAMS Foundation.
- Metz, C. (2015, March 12). How Github conquered Google, Microsoft, and everyone else. Retrieved May 22, 2018, from <https://www.wired.com/2015/03/github-conquered-google-microsoft-everyone-else/>
- Rackhman, S. (2018). *AsciiDoc*. Retrieved from <http://asciidoc.org>
- Stollhans, S. (2016). Learning by teaching: developing transferable skills. In E. Corradini, K. Borthwick, & A. Gallagher-Brett (Eds.), *Employability for languages: a handbook* (pp. 161–164). Research-publishing.net. <https://doi.org/10.14705/rpnet.2016.cbq2016.478>
- Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *Mis Quarterly*, 37(1), 21–54.
- Weinberger, M. (2017, March 31). \$2 billion startup GitHub has officially won over Microsoft. Retrieved May 22, 2018, from <http://www.businessinsider.com/microsoft-shuts-down-codeplex-to-use-github-2017-3>
- Weiss, B. (2012, June 17). GitHub Is Your Resume Now. Retrieved May 22, 2018, from <https://anti-pattern.com/github-is-your-resume-now>
- Wiley, D. (n.d.). Defining the “Open” in Open Content and Open Educational Resources. Retrieved May 31, 2018, from <http://opencontent.org/definition/>

## Appendices and Annexures

### Appendix A: Skeleton AsciiDoc File Provided to Students

= Tutorial Title Name

Author: Your Name Here

== Intro

Describe the goals and purpose of the tutorial here.

== Prerequisites

The "\*" is a bulleted list.

- \* VirtualBox
- \* Kali Virtual Machine

== Instructions

Guide the person through the steps. Starting a line with a period is an automatically numbered list.

- . Step one.
- . Step two.
- . Step three. Use "+" and the backtick character to add additional command.

```
+  
` ``
```

```
sudo su  
` ``
```

- . Step four. Use the following syntax to include screenshots or other images.

```
+
```

```
image::blue-rectangle.png[]
```

- . Step five. Compile with the following command:

```
+  
` ``
```

```
asciidoctor-pdf tutorial.adoc  
` ``
```

== Challenge

Provide some meaningful ways for the learner to apply their knowledge in a new way.

== Reflection

Provide some thought questions that help the learner make sense of how the tutorial fits in the bigger picture.

### Appendix B: Survey Questions

Q1) Rate the degree to which you agree with the following statements (strongly agree to strongly disagree).

- I enjoyed using AsciiDoc to create the tutorial
- GitHub is an effective tool for managing collaborative writing
- Git is an effective tool for merging versions of a project
- I spent a lot of effort learning the tools (Git, Github, AsciiDoc) that detracted from learning the topic of my tutorial

- The tools (Git, Github, AsciiDoc) were difficult to work with which degraded the quality of my finished tutorial.
- Knowing that my work would be made public made me strive to produce high quality work.
- Knowing that my peers would review my work encouraged me to do my best.

Q2) Rate the degree to which the following tools or features helped or hindered you as you created your tutorial (helped a lot to hindered a lot or did not use).

- Git
- GitHub
- Ruby
- AsciiDctor-PDF
- GitHub pull requests
- GitHub issues

Q3) Compare this assignment to other types of learning activities you have completed. For each of the learning activities listed below, rate your preference for learning activities (strongly prefer, slightly prefer, or no preference).

- PowerPoint
- Class discussions
- Following written exercises
- Watching videos
- Reading articles

Q4) What parts of the assignment did you enjoy? (50 characters minimum)

Q5) What parts of the assignment were most challenging? (50 characters minimum)

Q6) How would you change this assignment in the future to make it a better learning experience? (50 characters minimum)

Q7) Do you have any other comments that you would like to provide?