Git-based learning management system

by

Vahid Fazel-Rezai

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Author . . . .
Department of Electrical Engineering and Computer Science
May 25, 2018

Certified by . . . .
Philipp Schmidt
Research Scientist, Director Media Lab Learning Initiative
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chairman, Department Committee on Graduate Theses

Git-based learning management system

by

Vahid Fazel-Rezai

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2018, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

Instructors use software called learning management systems to administer online courses. Traditionally these digital platforms mimic physical classrooms, but new technology opens up potential for interactions not previously possible. In this thesis, we build a learning management system on top of git to create a learning environment that encourages collaboration and exploration. We discuss how it was designed following established learning principles, the technical implementation, and the results of a pilot online class, with 80 learners lasting 6 weeks, as part of the Refugee Learning Accelerator. The platform architecture used in this study can be replicated and reused for other online courses with similar goals.

Thesis Supervisor: Philipp Schmidt
Title: Research Scientist, Director Media Lab Learning Initiative

Acknowledgments

Acknowledgments to go here...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

New education technology, enabled by the Internet, has allowed learners to connect with their instructors and with each other digitally. Pieces of software known as learning management systems take the experience of a classroom online, giving remote learners more access to education and enriching their learning experiences with a variety of technology.

In this thesis we employ git, a version control technology initially created for software developers to collaborate, as the basis of a learning platform designed to encourage collaboration and experimentation. We first discuss the learning principles that guided our design, followed by an analysis of existing technologies we can use to piece together such a git-based learning platform. We then walk through the technical implementation. Finally, we present the results of a pilot run supporting an online course for a distributed group of learners in the Refugee Learning Accelerator.

As a result of this work, I hope that the barrier of entry to administer an online course using a git-based learning platform can be lowered. The design decisions made in this project produce a refined architecture that can be confidently reused in future courses. This architecture is assembled using a combination of existing third-party software and custom event-driven scripts, both of which are outlined in sections below.

Before expanding on the new platform developed in this thesis, we provide a summary relevant context and previous work.

1.1 Refugee Learning Accelerator

The Refugee Learning Accelerator (RLA) was a program run by the Learning Initiative, a group of the MIT Media Lab. The program aimed to gather and support technologists from the Middle East, specifically to work on education tools for refugee learners.

Interested candidates applied as teams through an open application. Around 80 participants were selected for the first phase of the program, which consisted of a 6-week online course during October and November of 2018. Every two weeks of the course, a new technology was introduced and teams were asked to design, and ideally build a prototype for, an education solution using that technology. Seminars were held throughout to help participants learn about the technology and teams received constructive feedback every week, once at the midpoint and once at the end of each challenge. The topics chosen for the online course were chatbots, human centered design, and AR/VR.

The online course served as a pilot for the learning platform discussed in this thesis. Following the course selected teams went on to participate in a week-long in-person workshop in Amman to nurture their project ideas and finally build out their ideas with funding and partnerships. Following the course, they were also surveyed about their experience as a means of gathering feedback for the platform. [refugeelearning]

Funding for RLA including this study and the development of this platform was generously provided by the open society foundation

We ran RLA with ... participants and instructors

We wrote a survey, questions in appendix, and got it approved by the irb. we sent the survey out by email and reviewed the responses.

1.2 Methodology

look up "design research" papers from the media lab and see their methodology

We decided on the type of course we wanted to design for RLA, and set out on creating all the components. In terms of the online platform, we defined our interactions and ...

We implemented the

Chapter 2

Background

2.1 Traditional learning platforms

A learning management system (LMS) is a tool that helps instructors supervise an online course. Popular commercial LMSs are Blackboard, Moodle, and Canvas.

A traditional LMS platform typically provides features including tracking learner grades, file management for assignments and corresponding submissions, and communication mechanisms for instructor announcements or learner questions. [cscw15] These features are often more convenient duplicates of the same in-person interactions of traditional classrooms. Furthermore, LMS usually donâĂŹt fully encompass the activities of a class, leaving the core of the learning for classroom instruction or learner self-study.

Of note, the issue of learner collaboration, which is often a component of more creative, project-based assignments, is often not properly addressed in traditional LMS platforms. This intersection of collaboration and technology is instead known as computer-supported collaborative work (CSCW), and families of tools exist to support online collaboration. Learners thus would have to take to these external tools to collaborate, depending on the particular assignment. [farooq]

2.2 Foundational theory

Assuming SDT (autonomy, relevance, competence), which explains why humans have an innate desire to do things and have individualized passions, we use the 4P's to achieve learning outcomes (reason —— design —-, work collaboratively.... mitch reference)

We had several goals in mind when deciding how to craft the online course, which can be summarized in four principles: Projects. Peers. Passion. Play. We believe learners do best when they are engaged in th

These principles largely rely on students being drivers of their own learning.

We turn to self-determination theory (SDT) as a model with which to understand learnersâĂŹ motivation. SDT [citation needed]

Self-determination theory: autonomy, competence, relatedness. Hence, we prioritize these in designing features of the platform.

2.3 Pedagogical tensions

overall course vs community.........

4 tensions content vs connection, individual pathway vs shared experience, solitary vs team, projects vs puzzles

for each tension, there are benefits to both sides, it's a matter of balancing the two

pedagogy and platform are closely linked but not causully linked

mastery->feedback, community

Chapter 3

Related Work

We set out to develop a new platform to address

3.1 Overview of git

A specific technology used for online collaboration, particularly in software engineering, is git. Created in 2005 by Linus Torvalds as a tool to support the development of the Linux kernel, git allows users to each have a local copy of a set of files and sync between their copy and a central server in a coordinated fashion. Git is also a version control system, hence allowing users to backtrack changes or split into multiple branches of work from the same starting state. [git]

Git is used as a command line tool, in which users type text commands to perform the desired git actions. In addition, there are graphical interfaces available which serve as a wrapper and aid for users who prefer not to use the command line. Overall, git is considered a powerful collaboration tool, but involves an upfront investment to learn its operations.

3.2 Communities of practice

Use of group chats such as gitter and mattermost to complement projects

3.3 Other uses of git

While git was initially created for and has been largely relied on by the software development community, its use has spread to other knowledge workers.

An example of applying git to a traditional use case is cookbooks: a service called Fork the Cookbook exists in which users can fork, or copy, an existing recipe and tweak it to make it their own. This creates a collaborative and decentralized platform for sharing cooking recipes. [1]

Example applications include recipes, legislation, datasets, general CMS, etc.

3.4 Existing git-based learning systems

Chapter 4

Platform Development

4.1 Design

In this section we lay out the high-level design for the RLA course. The git-based platform is developed as just one of the components of this course, but is always intended to be applicable outside the context of RLA to other future courses.

4.1.1 Balancing tensions

We also aimed to foster a community that would extend into the later phases of the Accelerator.

Finally, we rely on good software development practices and use popular tools where possible.

Communication. Communication lays the groundwork for the type of community formed. We aim to have instructors clearly communicate how to use the platform and be able to guide students throughout the course. We would like to allow natural and structured opportunities to ask questions and give feedback. General course information should be readily accessible to students.

Permissions. We will have to decide how to control authority to perform actions and visibility of content. On one hand, we believe that the platform should be as open as possible. This would allow students the opportunity to pursue their curiosities, to learn from each otherâĂŹs work, and to feel that they have a complete understanding of the platform. On the other hand, we would like to prevent either intentional or accidental actions that may destroy or hinder other participantsâĂŹ work. We also anticipate that sensitive topics would not be expressed without a private way of doing so and thus would like to allow for that.

4.1.2 Suitability of git

In this section we

While git was not designed as a learning platform, we found in it suitable qualities. Here we discuss the benefits of git as a tool for fostering positive educational interactions.

As a version control system, the currency is units of work rather than final deliverables. It is oriented around building larger projects through a series of work. We allow learners to choose their work based on their passions and do not specify a particular end goal. Git gives learners room to play, since branches can be made to explore new ideas and unsuccessful changes can be easily reverted. It is also very much designed for collaborating with peers. After all, its raison detre is to be able to build upon each othersâĂŹ work without conflict. Documenting everyoneâĂŹs work openly and allowing collaboration creates a shared space from which a community could form, just as many have in the open source world. Finally, because git is already widely-used, its kinks have been worked out and help material is readily available online, making it easier for new adopters.

Flexibility in creation tools, format, etc

As discussed in the previous section, git can also be adapted as the underlying mechanism of a content management system. As a versioning system, git enables publishing new content, updates, and revisions in a fashion that can be distributed. Popular git hosting websites such as Github and GitLab have built-in support for Markdown-formatted files and support and static web pages.

4.1.3 Platform architecture

The details of the design were chosen to support the needs of RLAâĂŹs weekly cycle. In each week of the cycle, content is released at the beginning and learners worked towards a particular milestone by the end of the week. Throughout the week learners can collaborate and participate in seminars.

To achieve these requirements, there are four interactions that the core git-based platform needs to support:

- **Onboarding.** We want learners to be able to register for the course as easily as possible, yet still provide basic information to introduce themselves to the rest of the learner community.

- **Content distribution.** New material posted at the beginning of the week includes a description of the weekâĂŹs challenge and targeted milestones. In addition, relevant resources relevant for the week are gathered and provided for learnersâĂŹ reference. This content may also sometimes include skeleton code or template documents in which learners can report their progress. These pieces of content require the additional ability of being pulled into learnersâĂŹ working directories.

- **Submission.** At the end of the week, learners need a mechanism to put together a package representing the state of their work on which to receive feedback. We also encourage learners to use these submissions as a resource to learn from other learnersâĂŹ approach to the challenge.

- **Feedback.** We believe frequent personalized feedback can be helpful to learners and also cultivates a culture of collaboration. Hence weâĂŹd like our platform to provide an interface where a reviewer can easily peruse a learnerâĂŹs work and leave comments. Note that this process is similar in structure to code review used in software engineering, so we can utilize similar features in the git hosting service.

In addition to managing the course content, we also want to support a few forms of communication that empower learners:

- For seminars, we need a way to broadcast video to learners. Additionally, we would like to engage learners to drive the conversation according to their passions and have face-time with their peers.

- For technical support and other miscellaneous discussion, we would like to have an open, real-time, group chat. In particular, the group aspect will expose the entire community to conversations as appropriate so that everyone feels included and in-the-know.

- For showcasing teams' work, especially to parties outside the course, it would be ideal to have a static website summarizing the challenges and the progress coming out of the course. Anyone could then share the link and in this way feel ownership over their work.

4.2 Third-party Software

4.2.1 Git hosting service

In order to have a group of distributed learners participate in the course using git, we needed to have an server accessible by the Internet running git where the content would be hosted. We looked at two of the most popular git hosting services, Github and GitLab. Both offer a SaaS model where they host git on their servers and simply provide an interface, as well as a self-hosted version where they provide the software but leave running a server to the customer.

We compared options the two in terms of features, styles, and technical performance.

The two options had very similar sets of features. Github has organizations, teams, repositories, and pull requests, while GitLab has groups, subgroups, projects, and merge requests. Both expose a webhook for git commits, allow in-browser editing of files, and can serve hosted files as static websites. Both offer free or discounted private repositories/projects for educational uses. On a few points GitLab had additional features, such as being able to request to join a group and being able to trigger scripts that run before serving static websites.

Many software companies use Github to host either all their code or even just their open source projects, and as a result a Github profile feels very much like a professional portfolio. We wanted to make sure learners are comfortable experimenting and making mistakes, and took into consideration that some may feel less inclined to do so if the work would permanently be associated with their professional portfolio.

Finally, we took into consideration the technical reliability of each option. We found that GitLabâĂŹs SaaS offering, GitLab.com, was often slow to load. For both, a self-hosted instance would include considerable setup and costs to run a server, but would allow customizations such as the domain name. Finally, though both are welcoming homes to open source projects, Gitlab has made the web platform itself open source, whereas GithubâĂŹs is not.

For our course, we ended up opting for GitLab, and chose to mitigate performance issues by using a self-hosted instance of the free tier. However, it would certainly be possible to adapt our approach for either GitLab or Github, in the both SaaS and self-hosted models.

4.2.2 Chat service

To allow the development of a community, we wanted to make communication among learners as well as between learners and instructors as open as possible. GitLab alone, with commit messages and comments on merge requests, did not offer as fluid of a setting as we wanted. We decided to augment GitLab with a team-based chat service where learners could communicate in real time.

We looked at existing cloud messaging services including Slack, Gitter, and Mattermost, all of which have a very similar design. They provide open channels where any user can join and chat, as well as messaging between users and in private groups for more direct communication. We hoped to use these spaces as a forum for discussing ideas, asking each other for help, and even just to meet other learners.

From our options we chose to run a self-hosted instance of Mattermost, largely due to our choice of self-hosted GitLab. Mattermost and GitLab have a built-in integration for authentication and backend configuration. Mattermost is also open source, continuing in our spirit of relying on open source tools.

4.2.3 Server infrastructure

4.3 Technical Implementation

In this section we outline the initial implementation of the platform. We assume a technical background, in particular familiarity with Internet architecture and git. 6.1 Server configuration DigitalOcean droplet https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-gitlab-on-ubuntu-16-04 GitLab [built in for DO] NameCheap domain and networking nginx customization Lets encrypt 6.2 Repo architecture Brief overview of possibilities Step by step description of how it all worked 6.3 Scripts As much as possible, git-related operations of the platform were automated to ease the burden on the instructors. Onboarding membership request poller Onboarding group creation Health check with email notifications Create subtrees Create merge request Merge Analyze progress

Chapter 5

Results of RLA Pilot

For six weeks in October and November of 2017, the approximately 80 learners participated in an online course using a git-based learning platform as the first phase of the RLA program. Following the course, an optional survey was conducted, the questions of which are included in the appendix. In this section we report the experience, results, and feedback we observed in the course and survey responses.

5.1 User Experience

Frame around tensions, for each broad set of interactions go over "what did it look like", "how did it work", and "how well did it work"

Chapter 6

Conclusion

narrative summary of thesis

an honest set of personal takeaways (from which reader can judge whether to replicate work) and feeligns about future work

Appendix A

RLA Online Course Survey

questions go here

Bibliography

[1] About Fork The Cookbook. About Fork The Cookbook. `http://forkthecookbook.com/about`. Accessed: 19-May-2018.

[2] Bates, A. W. (2015). *Teaching in a Digital Age: Guidelines for Designing Teaching and Learning*. Tony Bates Associates, Vancouver.

[3] Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*.

[4] Chacon, S. Getting Started - A Short History of Git. `https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git`. Accessed: 19-May-2018.

[5] Cherry, K. What Is Self-Determination Theory? `https://www.verywellmind.com/what-is-self-determination-theory-2795387`. Accessed: 19-May-2018.

[6] David, L. Self-Determination Theory. `https://www.learning-theories.com/self-determination-theory-deci-and-ryan.html`. Accessed: 19-May-2018.

[7] DevArt: Art Made with Code. DevArt: Art Made with Code. `https://devart.withgoogle.com/`. Accessed: 19-May-2018.

[8] Feliciano, J. (2015). Towards a Collaborative Learning Platform: The Use of GitHub in Computer Science and Software Engineering Courses. Master's thesis, Department of Computer Science, University of Victoria. An optional note.

[9] Github Classroom. Github Classroom. `https://classroom.github.com/`. Accessed: 19-May-2018.

[10] Haaranen, L. and Lehtinen, T. (2015). Teaching git on the side: Version control system as a course platform. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pages 87–92. ACM.

[11] Initiative, L. Refugee Learning Accelerator. `https://refugeelearning.media.mit.edu/`. Accessed: 19-May-2018.

[12] Kelleher, J. (2014). Employing git in the classroom. In *Computer Applications and Information Systems (WCCAIS), 2014 World Congress on*, pages 1–4. IEEE.

[13] Mozilla's Global Sprint. Mozilla's Global Sprint. `https://mozilla.github.io/global-sprint/`. Accessed: 19-May-2018.

[14] Niemiec, C. P. and Ryan, R. M. (2009). Autonomy, competence, and relatedness in the classroom: Applying self-determination theory to educational practice. *School Field*, 7(2):133–144.

[15] P. Schmidt, M. R. and Rusk, N. Learning Creative Learning: How we tinkered with MOOCs. `http://reports.p2pu.org/learning-creative-learning/`. Accessed: 19-May-2018.

[16] Resnick, M. (2014). Give P's a chance: Projects, peers, passion, play. In *Constructionism and creativity: Proceedings of the Third International Constructionism Conference*. Austrian Computer Society.

[17] Resnick, M. (2017). *Cultivating Creativity through Projects, Passion, Peers, and Play*. The MIT Press, Cambridge.

[18] Rheingold, H. Mitch Resnick: The Role of Making, Tinkering, Remixing in Next-Generation Learning. `https://dmlcentral.net/mitch-resnick-the-role-of-making-tinkering-remixing-in-next-generation-learning/`. Accessed: 19-May-2018.

[19] Rocco, D. and Lloyd, W. (2011). Distributed version control in the classroom.

[20] ScP. Schmidt, M. R. and Ito, J. Creative Learning and the Future of Work. *DISRUPTING*, page 147.

[21] Zagalsky, A. Why you should use GitHub: Lessons for the classroom and newsroom. `http://www.storybench.org/use-github-lessons-classroom-newsroom/`. Accessed: 19-May-2018.

[22] Zagalsky, A., Feliciano, J., Storey, M.-A., Zhao, Y., and Wang, W. (2015). The emergence of github as a collaborative platform for education. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 1906–1917. ACM.