Lightning Talk Proposal: Agile Courseware

Proposer

Gene Callahan, NYU Tandon School of Engineering

Abstract

Agile Courseware is not a rival to, or replacement for, educational approaches like Active Learning. The goal of Agile Courseware is to take whatever pedagogical approach works and deliver that material more easily, more rapidly, more error-free, and more collaboratively.

The advantages of Agile Courseware include:
Work is in open, text-based formats: parts can easily be extracted for use elsewhere.
The work exists in open repositories on the Internet enabling sharing and collaboration.
Open content developed for in-person classes can easily be transferred to online classes
It enables continuous learning: Because materials are based on open standards, generally text-based, and stored in public repositories, it is easy for other faculty members to learn from them.
Also, it is simple for other educators to improve on someone else's original work, so that the original author learns from others.
Text-based materials can easily be processed by scripts.
Rapid deployment: Only a text editor and Internet access are needed to deploy new versions. No need to access PowerPoint, fetch files from a laptop, login to a special machine, etc.

Students, acting as TAs, or open-source collaborators, will learn agile development and DevOps principles hands-on. I now have over 20 students contributing to open-source projects I have initiated. This is a great resume builder for all of these students.

A final advantage is that once these tools are mastered, they can be used not just for courseware, but for co-authoring books or papers, and, of course, for software projects.

Description

Adopting Agile Courseware ideas into my pedagogical practice has lifted my productivity to new heights. For instance, I don't need to remember what I've updated from my desktop when I get on my laptop to work on a course. No need to download particular files from Moodle, Classes, etc. to be up-to-date: I just run "git pull" and every new file is on whatever machine I am using!

Automating as much of the courseware creation as possible removes drudgery and errors. Using open-source tools, transparent formats (like HTML and JSON), and version control allows teaching assistants and other professors to easily contribute to the courseware.

These are some of the principles that Agile Courseware should embody:

- Promote collaboration
- Promote sharing
- Promote continual learning
- Iterative development
- Incremental development
- Continuous integration
- Continuous deployment
- Automate to remove drudgery and improve quality
- Breakdown silos between courseware provider, courseware user, between different teacher's courseware, and between different courses
- Mistakes can easily be rolled back
- Courseware can be manipulated by programs
- Platform-independent
- Easy to incorporate other resources
- Uses open-source tools whenever possible

Putting the above principles into practice should not be an all-or-nothing transition: that would be anti-DevOps. Instead, faculty can keep working as they have been, and gradually incorporate more "Agile Courseware."

Existing examples of Agile Courseware:

- Emu86
- Algorithms course: https://gcallah.github.io/algorithms/index.html