Collaborative Open Source Curriculum Development

PI: Kathryn Huff
kdhuff@illinois.edu
University of Illinois Urbana-Champaign

Co-PI: Neal Davis
davis68@illinois.edu
University of Illinois Urbana-Champaign

Paul Wilson
University of Wisconsin - Madison

Steven Skutnik
University of Tennessee - Knoxville

Anthony Scopatz
University of South Carolina

Jeremy Roberts
Kansas State University

Robert Borrelli
University of Idaho at Idaho Falls

April 30, 2017

1 Problem

At this very moment, hundreds of professors in the United States are simultaneously preparing lessons on transposing a matrix. They are doing so largely without receiving feedback from one another or directly building on one another's experience [1]. In this way, professors spend an enormous amount of time duplicating curriculum development efforts already tackled by colleagues. What's worse is that these efforts are rarely, if ever, reviewed by, shared with, or extended upon by peers.

Open source software development provides an excellent example of a possible solution. These communities have mastered distributed expert collaboration, dynamic peer review, and de-duplication of effort. In open source software, developers share code revisions in online repositories, review one another's work in small chunks [2], and contribute back their own improvements to the main project.

So, why aren't professors sharing their lesson materials online, collaborating on canonical lesson sets, diffing and merging similar lessons, and reviewing one another's learning materials? **Our goal is to explore the possibility that curriculum development for university courses can operate like open source software development does.**

Previous attempts at collaborative curriculum development have met with challenges. Most notable among these is Software Carpentry (with which the PI and Co-PI are quite involved) [3]. Software Carpentry teaches computational skills for scientists using a shared curriculum hosted on GitHub. Hundreds of instructors accross the planet use and remix the curriculum, but our experience showed that while these instructors used and improved on the original material, they rarely contributed their changes back to the master material. Thus, versions of the original material tend to diverge rather than converge [3, 4].

We expect that fine-grained modularity of lesson components and a clear dependency graph of prerequisite modules for each lesson may assist in overcoming the challenges encountered by Software Carpentry and others.

2 Proposed Solution

We propose a small-scale proof-of-concept for collaborative, open source, curriculum development to improve the transfer of lessons learned between instructors of the same course (either at a single university or among different campuses). This prototype collaboration will provide a template which could be adopted for collaboration among faculty teaching courses with an inherently larger scale (e.g. CS101).

2.1 Collaboration Roles

Faculty in Nuclear Engineering from five institutions[1] have agreed to be participants in this prototyping effort. We all currently use GitHub to store, revise, and collaborate on research, especially source code. Additionally, this select group already started to host our course curricula online as well, but these are typically single author repositories (e.g. [5]).

The PI and participants will collaborate on a master set of learning modules for an upper-division course in nuclear engineering : The Nuclear Fuel Cycle. In the near term, we will develop fine-grained lessons and assessments which may be mixed and matched to meet the learning objectives. The curriculum will be hosted on GitHub, tested by all of us, and improved continually.

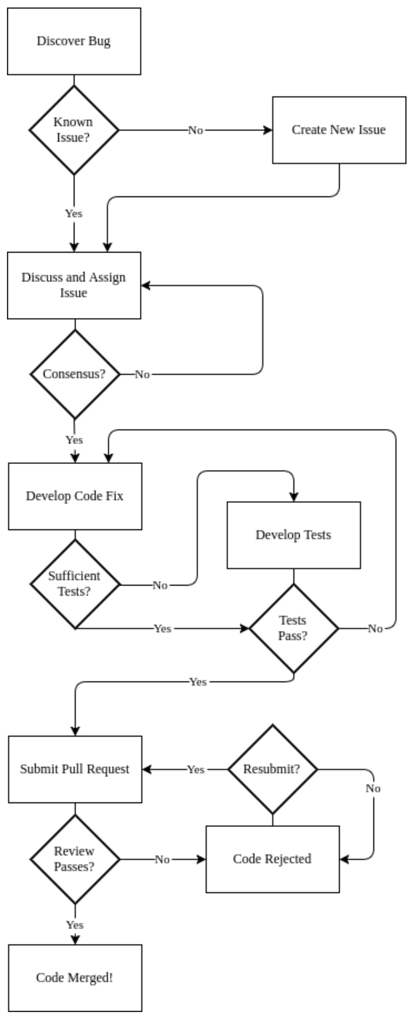
The PI and external collaborators will contribute and develop lesson material including lesson notes, in-class exercises, worked example problems, assessments (homework, quiz, and test questions), project assignment descriptions, media (images, movies), supporting material references, etc. These collaborators will primarily rely on the material they have developed for their own courses, but will need to convert that material into the format of the collective resource (format decided at the first workshop). Additionally, these collaborators will be core maintainers of the repository and will accordingly be responsible for conducting reviews of new material pull requests.

Prof. Neal Davis, Co-PI of this proposal, has contemplated a Collaborative Computational Curriculum for some time [6]. Since he teaches large-scale computational curriculum at Illinois he is an ideal person to potentially scale-up this work in the future. Additionally, he has the nuclear engineering background to understand the context of this prototyping effort. Prof. Davis' role in this effort will be to conduct *action research*, observing our process and teasing out potential avenues for future extensions and applications of the process.

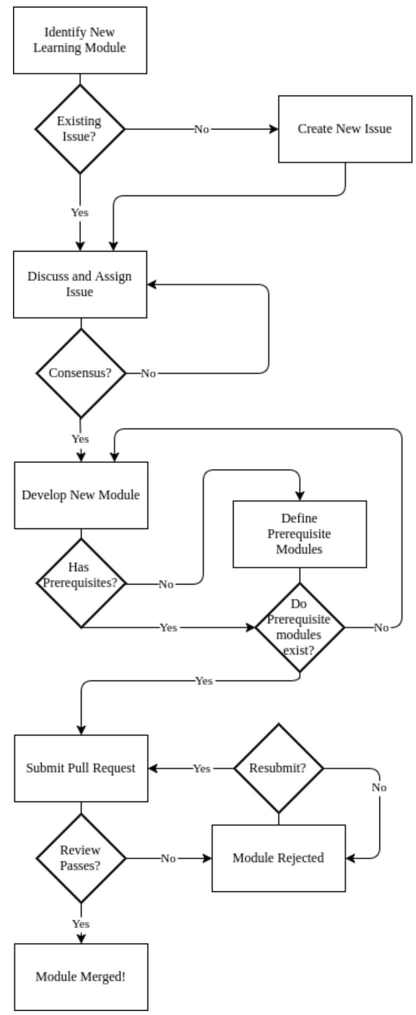
2.2 Open Source Workflow

Development of this proof-of-concept curriculum will be undertaken similarly to an open source software project. In particular, this work will emulate features of open source software development that cultivate distributed collaboration. In open source, for example, new code features are described in *issues* or *tickets* and discussed before implementation. Small, atomic changes are

[1] Paul Wilson at (UW Madison), Steven Skutnik (UT Knoxville), Anthony Scopatz (University of South Carolina), Jeremy Roberts (Kansas State University), and Robert Borrelli (University of Idaho)



(a) This figure, from [7], captures a process, often called Git Flow, through which a new feature or bug fix enters a piece of open source software.



(b) This adaptation imagines the process in the context of learning module development, using the same git version control system and GitHub repository hosting framework.

reviewed before they are merged into the *repository*. These discussions and incremental reviews nurture communities of practice.

In a typical open source project, a main copy of the repository (or main *fork*) holds the official copy of the software package. Individual developers each have their own *forks* where they can work on features and bug fixes. When the developer makes changes that are ready for prime time, they make a "pull request" to the main *fork*. That *pull request* is reviewed by their collaborators, and it is eventually merged into the main *fork* where it can be used by all. Figures 1a and 1b show how this open source software development workflow can be leveraged toward course development.

3 Timeline and Deliverables

3.1 Timeline

The year will be bookended by two workshops, a kickoff workshop and a retrospective workshop. In the intervening time, consistent communication through GitHub's collaboration framework, a Slack channel, an email listhost, a shared Google Drive folder, and monthly Google Hangouts will drive this work.

If Fall 2017, any material developed through this process will be incorporated into NPRE 412, the University of Illinois nuclear fuel cycle course taught by the PI, Kathryn Huff.

- **Jul 2017** Kick-Off Workshop
- **Aug 2017 - Dec 2017** NPRE 412 beta testing
- **Jul 2018** Retrospective Workshop

Details of the workshops and interim efforts are given in the next subsections.

3.2 Kick-Off Workshop

A two-day kick-off workshop hosted at the University of Illinois at Urbana-Champaign will allow the participants to sketch the initial framework of the collaboration and coordinate logistics. Through brainstorming and discussion activities, the workshop will deliver a number of products. These products will be published in an online repository and associated website, both hosted by GitHub.

Invited participants in the first workshop will include Illinois PI Huff and Illinois Co-PI Davis, excternal collaborating professors Wilson, Skutnik, Roberts, Scopatz, and Borrelli, and up to 2 Illinois professors who may be likely to extend this work in the future for their own courses.

As indicated in Table 1, products to be developed in the first workshop include learning objectives[8], a concept map[9], a list of fine-grained lesson topics, a directed acyclic graph describing dependencies among lessons (similar to the one in Figure 2. Additionally, concrete logistics for completion will be established, including individual curriculum development assignments, and a repository structure for organizing the lesson material. Additionally, practical decisions will be made such as establishing guidelines for lesson acceptability as well as agreeing on raw formats for lesson content (markdown vs. LaTeX, Jupyter vs MATLAB, etc.), At this workshop, interest and compatibility with frameworks like RELATE [10, 10] and tools like PrarieLearn [11] will be considered.

3.3 Interim Period

Between workshops, the PI and external collaborators will commence the work of developing lessons. These lessons will be submitted by pull request to the main repository and each lesson will include:

- associated learning objectives (identified previously)
- content (e.g. speaking notes, presentation material, derivations, worked examples, active learning exercises, external readings, videos, images)

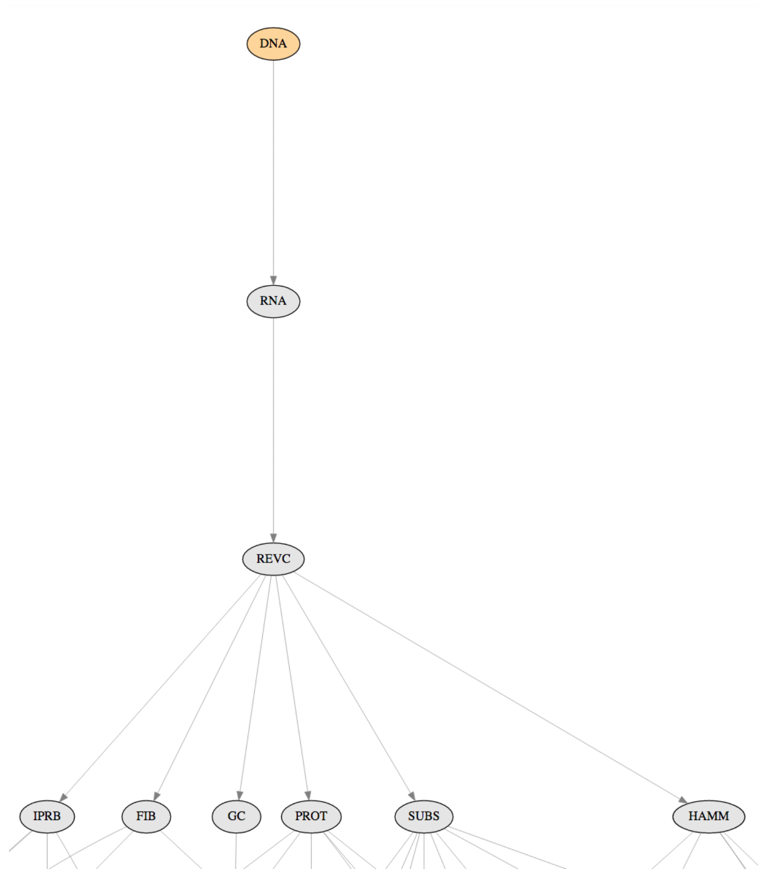


Figure 2: Rosalind.info, an educational bioinformatics resource, organizes exercises based on their dependencies.

- learning assessments (e.g. project descriptions, exam questions)

Interactions will commence primarily via issues, pull requests, and reviews on GitHub during this time. Meanwhile, monthly video conferences will help to spur high-level conversation onward. Co-PI Davis will be invited to engage in these video conference as part of action research toward the final As progress is made, lessons learned will be recorded by all participants in a draft instruction manual.

3.4 Retrospective Workshop

A two-day retrospective workshop ($15K) at Illinois will wrap up the project. Discussion concerning the process will allow reflection as well as a collection of lessons learned. Experiences live testing the developed course material (for example in NPRE412 at Illinois in Fall 2017) will be shared and suggested improvements will be captured as feature requests on GitHub. The lessons learned from the year-long process will be captured in a template GitHub repository and in a collaboration instruction manual in website form. Futu

With these resources, other groups of faculty seeking to collaborate in a similar way can simply fork the template repository and follow the instructions on the website to begin the process of

Day 1

Time	Activity	Lead	Deliverable
8:00	Welcome and Breakfast	Huff & Davis	
9:00	Vision Lightning Talk	Wilson	Idea Collection
9:15	Vision Lightning Talk	Skutnik	
9:30	Vision Lightning Talk	Scopatz	
9:45	Vision Lightning Talk	Roberts	
10:00	Vision Lightning Talk	Borrelli	
10:15	Vision Lightning Talk	Huff	
10:30	Break		
10:45	Concept Mapping Exercise	All	Curriculum Concept Map
12:00	Lunch	All	
13:00	Brainstorm Learning Objectives	All	Learning Objectives List
15:00	Break		
15:15	Refine Lesson Topics	All	Lesson Topics List
16:00	Discussion of Formats for Content	All	
17:00	Break		
19:00	Dinner		

Day 2

Time	Activity	Lead	Deliverable
8:00	Review and Breakfast	Huff & Davis	
9:00	Lesson Dependency Exercise	All	Lesson Dependency Graph
10:30	Break		
10:45	Lesson Dependency Exercise	All	
12:00	Lunch	All	
13:00	Repository Initialization		Template Workspace
14:00	Determination of Milestones	All	Work Plan
15:00	Break		
15:15	Assigment of Responsibilities	All	
16:00	Review and Wrap-up	All	
17:00	Break		
19:00	Dinner		

Table 1: Preliminary schedule for the Kick-off Workshop

developing their domain curriculum.

Invited participants in the retrospective workshop will include:

- Illinois PI Huff and Illinois Co-PI Davis
- external collaborating Professors Wilson, Roberts, Skutnik, and Scopatz
- up to 6 Illinois professors interested in extending this work in the future for their own courses.

4 Potential Impact

This work will have immediate student and faculty outcomes at all of the six universities where faculty are participating (Illinois, UW Madison, UT Knoxville, the University of South Carolina, Kansas State University, and the University of Idaho). It will additionally have long term student and faculty outcomes at Illinois. Beyond this, it may have institutional outcomes at Illinois.

This work will provide an important proof of concept for groups of instructors willing to collaborate on open source curriculum for:

- core courses with many sections in a single university
- niche courses taught by a select group of professors across universities
- fundamental courses in small fields (e.g. nuclear engineering)

Student Outcomes Students enrolled in NPRE412 at Illinois and peer courses at partner institutions (USC, UT-Knoxville, Kansas State University, and UW-Madison) will have an improved experience. Their curriculum will be bolstered by the comprehensive concept maps underlying their coursework[**?**], as well as exercises and assessment tools reviewed and improved by expert professors at peer institutions.

Illinois Faculty Outcomes The effort of developing curriculum in this way may be labor intensive initially. However, if the experience is similar to that of open source software, the maintenance effort will reduce substantially. Minor improvements and updates to the curriculum contributed by colleagues will be effectively 'free' in the long term.

Illinois Institutional Outcomes Openness and collaboration both increase visibility. The leadership of Illinois College of Engineering faculty in this effort, if successful, will be exemplary of the forward-thinking nature of the college and the university at large. Furthermore, extensions to this work may be competitive for support from educational grant-making agencies such as the National Science Foundation and even private foundations such as Moore and Sloan.

External Outcomes Of perhaps most relevance in the context of this work is the potential impact externally. This effort will spawn an instruction-centric community of practice among colleagues previously primarily connected by research in their specific technical subdomain. The participating professors at Illinois, UW Madison, UT Knoxville, the University of South Carolina, Kansas State University, and the University of Idaho teach similar courses at their home institutions in part because their research is in similar subfields. In research, it's obvious that many of us might collaborate, and this effort will nurture the same kind of collaboration within our teaching. This is very fitting with the vision behind SIIP – this kind of collaboration accross institutions within a scientific subfield is essential to "teaching like we do research."

5 Budget

The first workshop will be hosted at the Allerton Retreat Center, where a small meeting room will be reserved. Visiting invited workshop participants will be flown to Urbana-Champaign and lodged at the Illini Union Hotel. Table 2 provides details related to the expenses for these workshops.

Event	Item	Cost	Units	$	Notes
Kick-off Workshop	Meeting Room	744	2	1488	Allerton, all day package
	Lodging	150	24	3600	Allerton guest rooms
	Flights	800	6	4800	Estimated
	Taxi	20	12	240	
	Incidentals	100	6	600	
	Continental breakfast	8.5	24	204	Working breakfasts, Allerton
	Dinner	30	24	720	In Champaign-Urbana
Retrospective Workshop	Meeting Room	50	2	100	Illini Union
	Projector	300	4	1200	Illini Union
	Lodging	150	24	3600	Illini Union Hotel
	Flights	800	6	4800	Estimated
	Taxi	20	12	240	
	Incidentals	100	6	600	
	Continental breakfast	8.5	24	204	University Catering
	Coffee Service	32.5	4	130	
	Lunch	15	24	360	University Catering
	Dinner	30	24	720	In Champaign-Urbana
	Total			**23606**	

Table 2: All expenses will support travel for visitors and workshop necessities.

6 Departmental Support

The Department of Nuclear, Plasma, and Radiological Engineering will support this work in kind with release time for Kathryn Huff through her Start-Up funds. Additionally, NPRE will support workshop activities with administrative effort and by providing space. Finally, collaborating external participants will contribute in kind with summer time hours.

References

[1] Elizabeth Green. *Building a better teacher: How teaching works (and how to teach it to everyone)*. WW Norton & Company, 2014.

[2] Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, **Kathryn D. Huff**, Ian M. Mitchell, Mark D. Plumbley, Ben

Waugh, Ethan P. White, and Paul Wilson. Best Practices for Scientific Computing. *PLoS Biol*, 12(1):e1001745, January 2014.

[3] Greg Wilson. Software Carpentry: lessons learned. *F1000Research*, February 2014.

[4] Wilson. Software Carpentry: Lessons Learned, July 2014. https://www.youtube.com/watch?v=1e26rp6qPbA.

[5] **Kathryn D. Huff**. NPRE412, 2017. https://github.com/katyhuff/npre412.

[6] **Neal Davis**. University of Illinois Collaborative Computational Curriculum (UI-CCC), 2016. https://github.com/UI-CCC.

[7] Anthony Scopatz and **Kathryn D. Huff**. *Effective Computation in Physics*. O'Reilly Media, 1st edition edition, May 2015.

[8] B. S. Bloom, D. R. Krathwohl, and B. B. Masia. *Bloom taxonomy of educational objectives*. Allyn and Bacon, Boston, MA. Copyright (c) by Pearson Education.< http://www. coun. uvic. ca/learn/program/hndouts/bloom. html, 1984.

[9] Joseph D. Novak. Concept mapping: A useful tool for science education. *Journal of research in science teaching*, 27(10):937–949, 1990.

[10] Andreas Kloeckner. RELATE 2015.1 documentation, 2017. https://documen.tician.de/relate/.

[11] Matthew West. Prairielearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning. *age*, 26:1, 2015.