Teaching Coding in Journalism Schools: Considerations for a Secure Technological Infrastructure

Meredith Broussard
New York University
20 Cooper Square #702
New York, NY 10003
(212) 992-6974
merbroussard@nyu.edu

ABSTRACT
In this paper, I describe a technological infrastructure to support the unique teaching needs of data journalism instructors. I outline two network configurations that preserve student privacy and emulate the editorial workflow process.

Categories and Subject Descriptors
K.3.2 [**Computers and Education**]: Computer and Information Science Education—*computer science education, curriculum, information systems education.*

General Terms
Documentation, Design, Human Factors, Standardization.

Keywords
Computational journalism, data journalism, teaching, pedagogy, CS2013, CS0.

1. INTRODUCTION
In recent years, there has been a small movement toward teaching journalism school students to code. However, what is meant by "coding" varies greatly. Reading a handful of pro-coding articles, the authors refer to teaching HTML/CSS, Python, Ruby, Ruby on Rails, Javascript, Java, web scraping, PHP, mobile app development, web app development, database design, computational thinking, data journalism, analytics, and data visualization. [1]–[6]

This lack of consensus around the meaning of the term "coding" has made it understandably difficult to implement organized or consistent curricula around coding inside journalism schools. Another complicating factor is that most of the journalists and communication scholars who teach in j-schools are self-taught technologists; their formal training tends to come from the social sciences and the professions, rather than being acquired in engineering or computer science programs.

The fact that abundant Internet tutorials allow self-teaching is extraordinary. However, transferring self-taught skills to instructor-guided learning inside the classroom is not a straightforward proposition. An individual working on his own personal laptop has few constraints. The instructor teaching in a university-managed computer lab will find herself negotiating multiple constraints, including a variety of institutional rules around security, privacy, and software versioning[7]. These rules often create unanticipated obstacles. For example: unlike their colleagues in computer science departments, members of journalism departments are often restricted from accessing web servers or university cloud computing resources.

To date, j-school instructors who wish to teach coding have had to invent their own technological infrastructures to support student learning. This is inefficient. As coding skills are integrated across the curriculum, it will be more efficient for schools to implement consistent, replicable solutions under the institutional support umbrella rather than maintaining a variety of ad hoc efforts. This paper focuses on two technological infrastructure solutions I designed for my own data journalism classes. One solution was successfully implemented and the other is in progress. In the long term, designing and communicating best practices around server configurations will allow journalism school instructors to more efficiently teach technology within institutional contexts, thereby improving quality of instruction and protecting student privacy.

2. LITERATURE REVIEW
Coddington [8] offers a useful definition of the differences between data journalism, computer-assisted reporting, and computational journalism. For convenience, I use the term "data journalism" throughout this paper to refer to the practice of teaching a variety of quantitative and computational topics in a journalism school classroom.

In the U.S., the Family Educational Rights and Privacy Act (FERPA) governs the privacy of student educational records. Under FERPA, universities are obligated to keep student educational records private except for directory information. What, exactly, constitutes "educational records" is ambiguous. Definitions have largely been set as a result of litigation. For example, as a result of a 2002 court decision, it is not a FERPA violation to require students to read and comment on each other's work as part of a classroom exercise.

As part of FERPA, universities have policies around data stewardship. A definition from the National Center for Education Statistics:

> Data stewardship is an organizational commitment to ensure that data in education records, including personally identifiable information:

Computation + Journalism 2015, October 2-3, 2015, New York, NY, USA.

» Are accurate, complete, timely, and relevant for the intended purpose;

» Are collected, maintained, used, and disseminated in a way that respects privacy and ensures confidentiality and security;

» Meet the goals of promoting access to the data for evaluating and monitoring educational progress and educational programs; and

» Meet the goals of assuring accuracy to ensure that decisions relating to an individual student's rights and educational opportunities are based on the best possible information." [9]

Rakers [10] suggests that universities are in a unique position because of their responsibility for managing a huge amount of sensitive academic and administrative information for students, employees, and alumni. Faculty and staff awareness of FERPA varies greatly. Ideally, any technical solution for a j-school will function within the IT framework of the university.

In addition to enhancing information security and compliance, a school-wide solution will allow the school to avoid accumulating excess technical debt. Ernst [11] defines technical debt as: "The trading of long-term software quality in favor of short-term expediency." Organizations accumulate technical debt when they build a technological system for one case, then add on parts as needed without reviewing the overall system design or ensuring thorough documentation. This the technological equivalent of building the Burrow, the Weasleys' house in the Harry Potter series:

> It looked as though it had once been a large stone pigpen, but extra rooms had been added here and there until it was several stories high and so crooked it looked as though it were held up by magic (which, Harry reminded himself, it probably was). Four or five chimneys were perched on top of the red roof. A lopsided sign stuck in the ground near the entrance read, THE BURROW. Around the front door lay a jumble of rubber boots and a very rusty cauldron. [12]

After only a few semesters of using an ad hoc system, the instructor and the organization will find themselves saddled with an unwieldy code base that is hard to maintain. Such a code base decays over time [13], and usually needs to be completely refactored at great expense. Moreover, omitting the needs analysis process can result in software that does not fulfill the needs of the intended user base, students [14].

Based on an informal review of syllabi for tech-intensive journalism school classes, it seems that many of the skills and learning goals fall in the CS0 and CS1 levels of ACM's Computer Science 2013 curriculum guidelines for undergraduate programs in computer science [15]. Best practices in this document recommend allocating sufficient university resources to support teaching and professional development.

3. DISCUSSION

Learning management systems such as Blackboard, Canvas, or Piazza have been helpful in managing classroom assignments, grading, and online discussions. In the journalism classroom, however, the teaching needs are slightly different. Journalism students will eventually write and research for a wide audience, using a rapid publication cycle much faster than the average academic journal. To simulate this rapid turnaround cycle and to provide experience with common publishing tools, many instructors incorporate free online publishing tools such as Wordpress or Google Docs into the classroom.

Today's reporters use social media to contact potential sources and cover events; to train students for this, many instructors require students to set up social media accounts and post on social media as part of classroom assignments. It is also common to publish student journalism work on YouTube.

Free publishing services and social media allow instructors to set up publishing mechanisms outside the university umbrella. Mann [16] cautions instructors to be cautious when setting up such environments, as it has not yet been determined how FERPA restrictions interact with classroom exercises.

To balance privacy and rapid publishing, I implemented a solution for a data journalism class that may be useful to others teaching elements of "coding" inside j-schools. I offer that solution here, along with another proposed solution.

3.1 Wordpress Solution

Many schools have multi-site Wordpress installations that students and faculty can use to create sites for clubs and events. At the time that I began teaching data journalism, I was teaching at a university that had such an installation, but the university-wide installation did not have the plugins that I needed for my classroom exercises. Because IT policies required that Wordpress plugins go through a rigorous, time-consuming security audit before installation, I could not get the necessary plugins approved and installed in time for the start of the semester. Because of the way this Wordpress installation was configured, I also would not have comprehensive administrative rights over the site. This was an obstacle because one of my goals in the class was to teach students about different roles and administrative features inside a content management system (CMS). Most reporters enter their stories into a CMS, so familiarity with a CMS is a useful pre-professional skill.

At this point, I realized that I could set up a new Internet hosting account for about $50 a year, install the free Wordpress software myself, set up accounts for the 15-30 students in my class, and the situation would be resolved. However, I didn't want to do this because of the aforementioned FERPA considerations. Also, if I went outside the university technology umbrella, I could not take advantage of university single sign-on services, which in this case consisted of LDAP authentication. Students, like all users, tend to forget their passwords and lock themselves out of their accounts frequently. If I used a university service and a student forgot his password at 2 AM, there would be someone on duty at the university to help. If I made my own site, I would be the one who had to reset the password manually at 2 AM.

Eventually, I discovered another multi-site Wordpress installation at the university that did use LDAP authentication. I negotiated to have a site set up on it for each of my classes, and I negotiated slightly more comprehensive administrative rights for my ID only. It helped that one of the undergraduates in my data journalism class had a work-study job in the IT department, where he was the assistant Wordpress administrator. This undergraduate was able to make the case to IT for simple modifications that would have otherwise required a slew of meetings; his involvement also meant he could serve as a peer resource to the other students in the class [17].

We set up the class Wordpress site so that it was accessible on and off campus, but required a password to view any of the pages. Students could have the experience of creating a draft, editing the draft, and publishing it to see their work displayed online through the professional-looking template used by the CMS. The published work was visible through a browser and available to show to me and the other students in the class. However, the fact that it was inside a password-protected site meant that the work was not visible to the outside world.

This feature is very important. I feel strongly that student journalists must not be required to publish their work to the public until it has been edited and fact-checked. When a group of students hand in an assignment, very few of those assignments will be ready for wider publication. This is equally true in the newsroom: we have editors and copy editors in place for a reason. In the classroom, the professor serves as both editor and copy editor. The classroom is an appropriate place to commit both small and egregious errors, correct them, and learn from the experience.

Explaining the various permissions and document states of the class Wordpress installation also provided an opportunity to discuss security concerns with the students[18]. Troubleshooting the site's idiosyncrasies interrupted the class flow on more than one occasion; however, we turned it into a learning experience and by the end of the semester students understood software system interdependencies in a way they did not before.

3.2 Proposed Cloud Solution

At the university where I first taught data journalism, it was particularly important to create a classroom support architecture that worked with and in university computer labs because students' access to technology varied widely. Many of the students did not have off-campus access to computers. Few of the students had their own laptops; many lived at home with their families and shared an older-model desktop with other family members. Some students had only tablets and smartphones for personal use and did all of their homework in university labs. I had a few students who were on the far side of the digital divide, and others who were so unaccustomed to working on Macs that it took them until mid-semester to feel comfortable on the classroom computers. This variation in skill level is quite common in journalism classrooms, as is a significant amount of math phobia. I suspect that introductory programming classes in computer science departments do not face these specific issues, in part because advanced math is a pre-requisite for many computer science classes. I briefly considered requiring my students to purchase a specific laptop to use in data journalism class, but I rejected this option because I knew that requiring a $1000+ purchase would mean excluding students who couldn't afford this expense. Some journalism schools do require students to buy a specific type of laptop, which streamlines some of the technical challenges and allows students to purchase their computer with financial aid funds [19]; knowing the intricacies of financial aid purchasing rules may be useful for others who seek to implement BYO laptop programs in economically diverse university classrooms.

Taking into account all the aforementioned factors, I devised architecture to support the teaching of a variety of lessons. Based on conversations with peers teaching data journalism, the outer limit of journalism teaching right now includes teaching students to create Twitter bots, create data visualizations in D3, perform web scraping, develop simple Python scripts, and create small database-driven web applications, also called news apps. The latter sometimes takes the form of teaching students to develop Django apps.

News apps that use Django or other open frameworks are currently popular among professional data journalists [20]. One good example is ProPublica's "Debt by Degrees: Which Colleges Help Poor Students Most?"[21] This interactive, database-driven news app invites readers into the story by allowing them to search the specific colleges that are most relevant to their own lives. This technique, of seamlessly weaving together code and story, is one of the great strengths of news apps. However, such database-driven news apps are typically developed *outside* of a news organization's CMS [22] and are "skinned," or given a façade, to make the app appear visually identical to the rest of the site. For an advanced web developer, this is technically not very difficult. The CMS is on one server; the app is on another server, and some HTML code links the servers to display a web page inside a browser. For students who are developing computational literacy, however, this is an advanced concept, and implementing such a configuration is an extremely complex task.

Let's take a simple Django app as an example, For the students, a key concept is learning that a web app is made up of different parts: the database, the application layer, and the presentation layer (again, in the newsroom, the presentation layer is often integrated into a CMS or skinned to look like it is part of a CMS).

Figure 1: Django app architecture

A web app needs to run on a web server. So, what is needed to teach is a simple cloud-based web server that students can put code on and see it run.

I specify cloud-based because in the labs where I previously taught, and in many other university computer labs, users were not permitted to run a local web server on a lab computer. A student could run a local web server on a personal laptop, but then we are back at a BYO laptop program, which would not work because of the aforementioned economic constraints of the students in the room. I could have required each student to purchase a personal account on a web hosting service such as Bluehost or Linode, but then LDAP authentication would not be possible and students would be responsible for their own web server administration, which would likely lead to cognitive overload.

Instead, the ideal solution is to have a class-specific web server created and maintained by a university resource who is skilled at server administration and consults with the instructor. This division of labor is essential because of a fundamental difference between computer science departments and journalism departments. In computer science departments, *most* instructors and technical staff are familiar with the command line and can do basic programming and server administration. In journalism departments, *most* instructors and technical staff are not and cannot. It is simply more efficient for data journalism professors

to collaborate with IT around server administration rather than having the professors attempt to do everything themselves.

Just as we teach students about writing and revising drafts, it is also useful to teach students about the process of writing and revising code, and testing it before it goes into production. Currently, only the largest news organizations have data journalism teams [23]. Many students who graduate with data journalism experience will find themselves to be the only coder in the newsroom, which means that they will be responsible for developing code, launching it, and maintaining it in production [24]. The classic server configuration for a staged rollout is thus helpful. Students can develop their code on the development server, test it on the test server, and put final code on the staging server. The student can use Piazza, Blackboard, etc. to submit a link to their final app on the staging server. The instructor can look at the final product and grade it based on what is displayed, instead of having to download code and get it working on a local machine. This also introduces students to concepts of good practice around security for production environments.

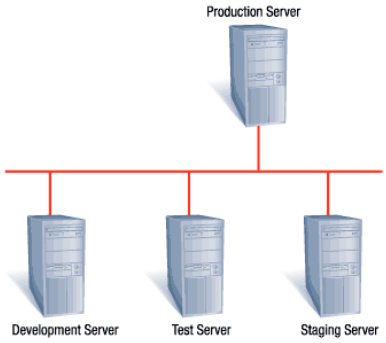

Figure 2: Proposed cloud server configuration

Should the student desire, the app can be moved over into production so it can serve as a portfolio piece when applying for internships and jobs. This step should not be mandatory, however. Students could be given the option to have the production app "expire" and be removed from the system after a student-determined length of time. Alternatively, the university might choose to maintain a class web server network for a fixed amount of time after a class ends (one semester or one academic year, perhaps) and then destroy the apps and other ephemera produced in the class.

4. CONCLUSIONS

In this paper, I have outlined two network configurations to support instructors who are teaching a variety of data journalism topics. Because the field is new, and a core data journalism curriculum has not yet emerged, data journalism instructors employ a variety of ad hoc solutions in their classes. In presenting this paper, I hope to launch a dialogue about security, privacy, longevity, technical debt, and what is needed to support the teaching of data journalism and advanced computational topics inside journalism schools.

5. REFERENCES

[1] P. Bradshaw, "How to Teach a Journalist Programming," *Data Driven Journalism*, 13-Aug-2012.

[2] A. Chimbel, "Why Journalism Students Need a Baseline Understanding of Coding," *MediaShift*, 26-Jan-2015.

[3] O. Khazan, "What Should Reporters Learn in Journalism School?," *The Atlantic*.

[4] O. Khazan, "Should Journalism Schools Require Reporters to 'Learn Code'? No.," *The Atlantic*, 21-Oct-2013.

[5] M. Mulligan, "Want to produce hirable grads, journalism schools? Teach them to code.," *Nieman Journalism Lab*, 05-Sep-2012.

[6] J. Spinner, "The Big Conundrum: Should Journalists Learn Code?," *American Journalism Review*, 24-Sep-2014.

[7] G. Madden and J. B. Tyndall, "Unrestricted Secure Computing," 2014, pp. 47–51.

[8] M. Coddington, "Clarifying Journalism's Quantitative Turn: A typology for evaluating data journalism, computational journalism, and computer-assisted reporting," *Digit. Journal.*, vol. 3, no. 3, pp. 331–348, May 2015.

[9] National Center for Education Statistics, "Data Stewardship: Managing Personally Identifiable Information in Electronic Student Education Records," NCES 2011-602 Brief 2, Nov. 2010.

[10] J. Rakers, "Managing professional and personal sensitive information," 2010, p. 9.

[11] N. A. Ernst, "On the Role of Requirements in Understanding and Managing Technical Debt," Zurich, Switzerland: IEEE, 2012.

[12] J.K. Rowling, *Harry Potter and the Chamber of Secrets*. New York: Scholastic, 2002.

[13] S. G. Eick, T. L. Graves, A. F. Karr, J. S. Marron, and A. Mockus, "Does code decay? Assessing the evidence from change management data," *IEEE Trans. Softw. Eng.*, vol. 27, no. 1, pp. 1–12, Jan. 2001.

[14] T. M. Murphy, J. Teng, and R. Matusky, "Needs Analysis for Instructional Technology Projects," 2014, pp. 23–28.

[15] ACM Computing Curricula Task Force, Ed., *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, Inc, 2013.

[16] K. Mann, "Online Assignments and Student Privacy," *Academic Technology at the College of William and Mary*, 20-Jun-2013.

[17] S. Reges, J. McGrory, and J. Smith, "The effective use of undergraduates to staff large introductory CS courses," 1988, pp. 22–25.

[18] A. Siraj, S. Ghafoor, J. Tower, and A. Haynes, "Empowering faculty to embed security topics into computer science courses," 2014, pp. 99–104.

[19] K. Krueger, "Are Laptop Requirements a Forgone Conclusion or a Burden on Students?," *MediaShift*, 05-Apr-2015.

[20] M. Broussard, "Reading Today's News on Future Computers: The Particular Challenges of Archiving Interactive News Applications," *Newsp. Res. J.*, vol. 36, no. 3, Sep. 2015.

[21] S. Wei and A. Waldman, "Debt by Degrees: Which Colleges Help Poor Students Most?," *ProPublica*, 12-Sep-2015.

[22] S. Klein and T. Fisher, "A Conceptual Model for Interactive Databases in News," *The ProPublica Nerd Blog: Secrets for Data Journalists and Newsroom Developers*, 18-Mar-2014.

[23] K. Fink and C. W. Anderson, "Data Journalism in the United States: Beyond the 'usual suspects,'" *Journal. Stud.*, pp. 1–15, Aug. 2014.

[24] R. Schallom and T. Machado, "Big Ambition, Small Staff, How the F*** Do I Prioritize?," presented at the SRCCON, Minneapolis, MN, Spring-2015.