mozilla

Teaching Scientists to Think Like the Web

Mozilla Foundation
Final Report to the Sloan Foundation
Grant number B2012-01
October 2012

Table of Contents

Mozilla Foundation Final Report to the Sloan Foundation: Teaching Scientists to Think Like the Web

1. Introduction

In January 2012, the Sloan Foundation awarded a grant of $124,625 to the Mozilla Foundation to support our work with Software Carpentry around teaching scientists the basic skills they need to use computers and the web more effectively in their research. Eight months later, we have run workshops and tutorials for over 680 scientists (a number that will reach 1000 by the end of October 2012), and two independent assessments have shown positive impact. We have also made progress on institutional engagement, have secured over $92,000 in additional funding for this initial work, and have shown that our approach is scalable: a third of the workshops to date have been peer-led, and more than two-thirds of those currently scheduled will be.

2. Deliverables

Our scheduled deliverables and actual achievements are summarized below.

- **Workshop curriculum and instructional materials:** We have developed a core two-day curriculum[1] that covers the Unix shell, Python, version control, testing, and databases.
- **Two workshops delivered in concert with university faculty, and two workshops delivered through a grassroots, peer-led approach:** By the end of October 2012, we will have delivered 27 workshops. 17 of these will have been led or co-led by the PI, and the rest run independently (Appendix A). By the end of 2012, those numbers will have increased to 39 and 26 respectively. We have also run two rounds of an online study group for instructors and would-be instructors[2]. 18 people participated in the first round (of whom 15 completed), and 25 are taking part in the second round (which will be completed in December).
- **A minimum of 80 students completing 10 self-led learning challenges:** Only a handful of our learners showed interest in doing challenges we defined. (As one said, their research is challenging enough.) Most of the funding earmarked for this was redirected to running more workshops, which is what scientists told us they wanted most.
- **Documented analysis of the relative success of each approach:** Dr. Jorge Aranda (University of Victoria) has been assessing the program by applying both quantitative and qualitative techniques to participants from several different workshops, and Prof. Julie Libarkin (Michigan State University) conducted a detailed assessment of participants in the MSU workshop (which included remote learners from the University of Texas at Austin). Their findings have been positive—for example, over 90% of workshop participants would recommend our workshops to colleagues—and are discussed in Section 4.

  A badge program: Work on the software required for issuing badges through our web site was slightly delayed, but came online in mid-August. At that point, we began issuing badges for learners (Section 6), but quickly realized that they had similar difficulties with badges that they

[1] http://software-carpentry.org/2012/03/what-we-teach-in-two-days/

[2] http://teaching.software-carpentry.org

had with the challenges that we defined (discussed above). Understanding around badges is still in early stages but we did find that there *is* significant interest in badging for instructors (Section 5), as this is seen as directly useful to career development. We have issued 15 “Instructor” badges to date, and expect to issue two dozen more in December when the second cohort of participants in our online study group complete their training. We are excited to have found an early test bed in instructors and we will continue to support them.

- **Feedback from institutions regarding their interest and requirements to engage in future iterations of the project:** Almost all of the feedback we have received has been positive and supportive. We have scheduled, or are in active planning for, workshops in over 15 new venues (Appendix A), and expect to run repeat workshops in 2013 in at least eight of the venues visited during the past five months.
- **A plan for institutional engagement:** We are making progress on two fronts (Section 6): having Software Carpentry count toward NSF-required training, and using it as the basis for a "driver's license" for a new European high-performance computing facility. We have also submitted a paper to the *Proceedings of the National Academies of Science* titled “Best Practices for Scientific Computing”, which we believe will stimulate further interest in the upper echelons of academia.
- **15 hours of "how-to" video enhanced with Popcorn.js metadata:** Our learners showed little interest in more instructional video, so after discussion with our program officer in May 2012, we redirected the resources intended for production of these to running more workshops. We did however make several hours of high-quality video from the workshop at Indiana University, and from studio recordings in Toronto, available to instructors as a teaching aid. These videos are available alongside our other material in our YouTube channel[3].

Four metrics were also defined in the proposal; our performance relative to them is summarized below.

1. **The percentage of online course participants that offer or plan to offer in-person workshops and study groups on their own, as well as recommend the online courses to their peers:** 96% of the participants in the joint Michigan State–University of Texas workshop said that they would recommend the workshop to others. As noted earlier, we are already in planning to run workshops in 2012–13 in eight of the eighteen venues visited in the past four months, and expect that number will grow substantially. The surest indicator, however, is that our workshops are typically over-subscribed by a factor of two or three. And importantly, all but two of the instructors who are actively teaching now started off as learners, either in workshops or self-directed. More information can be found in the 4th metric below.
2. **Badge display and associated reputation:** As detailed below, we shifted our focus from credentialing learners to credentialing instructors. Most are further along in their careers and find recognition of special accomplishments valuable as they apply for post-doctoral or faculty positions. We expect to be able to report on this shift in the first half of 2013.
3. **Institutional engagement:** Faculty members at Michigan State University and Utah State University are asking their universities to count Software Carpentry as fulfillment of the National Science Foundation's requirements for training in Responsible Conduct of Research (RCR). If this is approved, we expect a significant increase in interest at those institutions,

[3] https://www.youtube.com/user/softwarecarpentry/videos

nearly 100% uptake of badges, and emulation at other institutions. In addition, Software Carpentry is working with the Software Sustainability Institute in the UK to formulate a 'driver's exam' for the new DiRAC high-performance computing facility. The aim is to provide formative assessment for scientists who wish to use DiRAC via an hour-long examination that will include written, recorded, and interactive components. See Section 6 for further discussion.

4. **Former students becoming makers:** As discussed in Section 3, we shifted focus based on feedback from our learners. We initially planned to have learners create new examples and lessons for inclusion in our web-based material. What we found was that people who are interested in contributing would much rather do so in person, e.g., by helping out with workshops or online tutorials. As a result, we shifted our focus to recruiting and training the next wave of instructors. 36 people have helped deliver Software Carpentry workshops, half of them independently, and five of those have run online tutorials reaching dozens of people each. As discussed in Section 5, our success here is allowing us to scale significantly over the next two years.

3. Where We Shifted

In April 2012, after gathering feedback from learners and instructors, we re-prioritized a number of elements of the program. These adjustments were decided and managed in accordance with Mozilla's agile and iterative approach to program implementation. They were approved by the Sloan Foundation in May 2012.

Instructional videos
As mentioned above, we originally planned to develop 15 hours or more of new video content during this work. However, it quickly became clear that our target audience preferred to devote a concentrated block of time to learning, rather than trying to fit a little in each week. This realization prompted us to re-prioritize in early May: most of the funding intended for developing new content was redirected toward running more workshops, a smaller portion toward support for running online tutorials, and the rest toward developing an instructor's guide (discussed below).

Badges
The funding that was designated for "technical production of badge media assets" was redirected toward support for development of a badging plugin for WordPress (a popular open source blogging and content management system which powers many sites, including Software Carpentry's). This will allow us to issue badges directly from our site rather than relying on third parties. This functionality is still in development in relation to Mozilla's broader badge infrastructure, but our aim is ultimately to issue badges to instructors, as described in Section 2.

4. Assessment

The bulk of our assessment work was performed by Dr. Jorge Aranda (University of Victoria) who:

- surveyed participants in nine different workshops in Canada, the United States, and the United Kingdom (a survey of a ninth workshop is under way);

- interviewed participants from five different workshops in Canada and the United States;
- attended one workshop, where he performed observations on workshop presentation and participants' reactions; and
- analyzed seven screencasts recorded by participants as they worked through a programming assignment, in order to evaluate this technique's potential as evaluation instruments.

Dr. Aranda's survey explored participants' knowledge of the topics covered by the workshop and their perspective of scientific computing challenges, needs, and abilities. Participants were surveyed twice: days before their workshops (191 responses), and weeks or months after their conclusion (87 responses). The survey found considerable increases in participants' use of shell commands, version control tools, Python, and testing techniques. It also found significant improvements in their comprehension of shell commands, version control instructions, Python and SQL statements, and testing concepts. Perhaps more importantly, participants reported several kinds of improvements in their work habits and abilities after taking the workshop: better proficiency with Software Carpentry tools, greater concern for and awareness of issues of provenance and code quality, better strategies to approach software development (programming efficiency, test-driven development, automated testing, improved code structure), and even new research questions that have become accessible thanks to an increase in participants' software development skills.

Concurrently and independently, Prof. Julie Libarkin (Michigan State University) performed a more detailed assessment of participants in a workshop held there, which was also attended remotely by students from the University of Texas at Austin. Her survey assessed participants' satisfaction with the workshop: 85% reported they learned what they hoped to learn, 81% changed their computational understanding, and 96% said they would recommend the workshop to others.

The interviews allowed us to determine our interviewees' abilities and grasp of scientific computing issues before and after the workshop. There were 69 interviews in total (38 pre-workshop, 12 during a workshop, and 19 post-workshop). Interview data confirm several key assumptions of the Software Carpentry program:

- A large portion of scientific computing researchers is self-taught programmers.
- While they are highly intelligent, they lack a familiarity with basic software development tools and programming techniques that most software engineers take for granted.
- These shortcomings reduce their ability to answer their research questions.
- Attendance to a Software Carpentry workshop enables them to identify their weaknesses, prods them to act on improving their skills, and provides them with the means to do so.

The interviews also helped us identify problems with the current version of the workshops, of which the most significant was finding a pace suitable for the widely varying levels of expertise on different topics by different participants.

The findings from the surveys and interviews described above, along with the workshop observations and screencast analyses, helped us prepare an assessment procedure to be used in future implementations of the workshop. The new assessment procedure will allow us to have a systematic, repeatable, and sustainable evaluation process for Software Carpentry that will continue to inform us on the program's efficacy and to give us feedback on its remaining improvement areas.

5. Instructors

The biggest obstacle to Software Carpentry reaching more people has been the shortage of instructors to lead workshops. We have made excellent progress in correcting this: 19 people other than the PI have led or assisted with workshops since February 2012. *All* of those 19 volunteers have said that they wish to organize and run future workshops, and another 11 of them are currently scheduled to do so.

At the same time, three former learning assistants took on the load of organizing and running online tutorials for workshop alumni. We delivered 25 such tutorials between February and June 2012, several of which covered areas in which the PI does not have expertise, such as statistical computing packages and the newer generation of version control systems. Participants found these tutorials helpful and frustrating in equal measure, largely because of technical difficulties with online video conferencing systems. We are currently experimenting with other ways of implementing distributed tutorials (e.g., using some form of co-piloting with the IPython Notebook), and hope to resume this work early in 2013.

With only one exception, our instructors' backgrounds are in science and engineering, and they have no formal training in computer science. This gives them first-hand understanding of our learners' needs (and credibility with them), but they report that they feel insecure about their understanding of software development. They also report feeling that they don't know enough about education (which is also a common complaint among new university faculty). We are accordingly preparing a prose version of the core Software Carpentry notes; this will be useful to learners, but is primarily intended to be a reference for instructors.

We also worked on crowd-sourcing a short guide to educational research and practice for our instructors. There was much initial enthusiasm from contributors but it has taken more time than originally expected to complete the content. The material we have gathered so far forms the initial basis for the guide, and as this is a publication that is seen as necessary for Mozilla's larger learning efforts, we will continue to develop it into 2013. See Appendix C for a list of the articles that we have gathered so far.

Going forward, we plan to enlarge our badging program for instructors to increase the value to them of contributing to the program. Many universities are pushing departments to move teaching online, which means that having credentials for online teaching makes our instructors more interesting candidates for post-doctoral and tenure-stream positions. We believe that this, rather than badges for learning per se, will have the most significant short-term uptake.

We also believe that as the pool of instructors grows, we will be able to revisit the idea of learning challenges. Such challenges are most effective when learners define them for themselves, but most of the challenges that graduate students would define would require highly specialized domain knowledge. A larger and more diverse pool of instructors will allow us to handle these more effectively.

6. Engagement

Academic institutions are not quick-moving, but we have made progress on two fronts. First, faculty allies at Michigan State University and Utah State University have asked their universities to count Software Carpentry as fulfillment of the National Science Foundation's requirements for training in Responsible Conduct of Research (RCR)[4]. Feedback has been positive, and we hope to have approval by the end of 2012.

Second, Software Carpentry is working with the Software Sustainability Institute[5] to formulate a "driver's exam" for the new DiRAC high-performance computing facility. The aim is to provide formative assessment for scientists who wish to use DiRAC via an hour-long examination that will include written, recorded, and interactive components, successful completion of which will result in the examinee being awarded a Software Carpentry badge. We trialled this exam in July and August 2012, and reported to DiRAC's directors at their September 6, 2012 meeting in Glasgow; we expect a final decision in mid-November 2012. While the number of people involved is likely to be only in the hundreds, we feel this will be invaluable experience for us, and a useful precedent for institutional acceptance of badges as meaningful credentials.

7. Additional Support

Since the start of this work in January 2012:

- Mozilla Foundation received $6,000 in support of travel costs for workshop instructors from Enthought Inc.
- Software Carpentry received £12,000 ($18,800) from the Sound Software consortium at Queen Mary, University of London.
- Software Carpentry received $30,000 from the BEACON Center at Michigan State University.
- Software Carpentry received $28,000 from Lawrence Berkeley National Laboratory.

More importantly for the long-term sustainability of this project, we have moved to a “host pays expenses” model for workshops: for all but three of our upcoming workshops, the host site is paying travel and accommodation costs for one or two instructors.

8. Future Work

This pilot program has demonstrated that our approach works, that researchers want more of it, that we can scale, and that we can become self-sufficient. Independent assessment of the pilot phase found that scientists enjoyed the training and that it improved their skills. Workshops are almost always over-subscribed and alumni are beginning to run follow-on tutorials on their own. Given this success, we are eager to increase our ability to reach more participants, expand the training to teach scientists how to use the web in their research, and provide for longer, ongoing support.

As laid out in the follow-up proposal approved by the Sloan Foundation board in October 2012, our

[4] http://www.nsf.gov/bfa/dias/policy/rcr.jsp

[5] http://software.ac.uk

plans for future work over the next 2 years include:

- improving the workshops;
- increasing the number of workshops;
- including more content on the web;
- supporting peer-to-peer mentoring; and
- launching a webmaking science lab.

Not only have we demonstrated that Software Carpentry can scale, but its ‘graduates’ are starting to form a community of practice around science on the web. The new investments by Sloan into the Webmaking Science Lab will set up the ‘basket’ in which that activity can continue to grow.

Please see that proposal for specifics on strategy and tactics.

Appendix A. Workshops

The table below shows the date, location, instructional type (PI-led vs. peer-led) and size of our workshops.

Venue	Dates	Participants
Space Telescope Science Institute	Jan 18-19	9
International Center for Theoretical Physics	Feb 20-Mar 1	46
University of Toronto	Feb 23-24	20
Indiana University	Mar 7-8	27
Monterey Bay Aquarium Research Institute	Mar 26-27	41
NERSC	Mar 28-29	40
University of Chicago	Apr 2-3	35
Utah State University	Apr 14-15	23
University College London	Apr 30-May 1	53
Michigan State University	May 5-7	44
University of Newcastle	May 14-15	53
University of Alberta	May 16-17	24
University of British Columbia	May 22-23	43
Johns Hopkins University	June 18-19	12
INRIA (Paris)	Jun 28-29	36
MIT	Jul 9-10	39
Rutherford Appleton Laboratory	Jul 10-12	25
University of Waterloo	Jul 12-13	23
Dalhousie/Queen Mary's	Jul 16-17	34
University of Toronto (Scarborough)	Jul 19-20	23
DAFx Conference (York)	Sep 13-14	30
University of Oslo	Sep 17-18	22
Columbia University	Sep 28-29	44
Purdue University	Oct 8-9	37

We currently have workshops scheduled in the first quarter of 2013 at:

- University of British Columbia
- University of California (Berkeley)
- California Institute of Technology
- Lawrence Berkeley National Laboratory
- University of Oxford
- McMaster University
- Washington University in St. Louis
- Scripps Institute
- University of North Carolina
- University of Hawaii
- University of Edinburgh
- University of Texas (Austin)
- University of Chicago
- University of Waterloo
- Technical University of Munich
- Max Planck Institute (Tubingen)
- Virginia Tech
- Macquarie University
- AMOS Conference (Melbourne)
- University of Washington

and we are in the planning stages for workshops in the first half of 2013 at:

- American University in Beirut
- MIT
- Clemson University
- Duke University
- George Mason University
- Georgia Tech
- Indiana University
- Christian Albrechts University (Kiel)
- Monterey Bay Aquarium Research Institute
- McGill University
- North Carolina State University
- Ohio State University
- Pennsylvania State University
- Stanford University
- Tulane University
- University of Arizona
- University of California (Davis)
- University of Virginia
- Uppsala University
- Woods Hole Oceanographic Institute

We expect to add at least a dozen sites to the list for Q1-Q2 2013 by the end of 2012.

Appendix B. Our Team

Carlos Anderson is a Ph.D. candidate in Evolutionary Biology at Michigan State University, where he is studying the genetic mechanisms of speciation using artificial life. He obtained his B.S. in Computer Science and M.S. in Biology at the University of Central Florida.

Jorge Aranda obtained his Ph.D. in Computer Science at the University of Toronto. After doing post-doctoral researc at the University of Victoria, he now works for a software startup.

Dhavide Aruliah is an associate professor at the University of Ontario Institute of Technology in Oshawa, Ontario. His research interests are in scientific computing, specifically in computational inverse problems, numerical linear algebra, and the numerical solution of PDEs.

Rosangela Canino-Koning returned to university after 13 years of slogging in the software industry trenches to pursue a Ph.D. in Computer Science and Evolutionary Biology at Michigan State University. In her copious spare time, she reads, hikes, travels, and hacks on open source software.

Chris Cannam is a software developer with the Sound Software project at Queen Mary, University of London. He has had extensive experience as a commercial software developer and on numerous open source applications, particularly in the music and audio fields.

Neil Chue Hong is Director of the Software Sustainability Institute, and is based at the University of Edinburgh. His research interests are in community engagement and development, software sustainability, and the integration and analysis of data.

Matt Davis is a software developer at the Space Telescope Science Institute where he works on Python and C projects that support Hubble science. He also spends a bit of time spreading Python around the office. He previously worked at NASA’s Goddard Space Flight Center where he wrote Python software to support atmospheric science research.

Jonathan Dursi is an astrophysicist with twenty years' experience in computational science and computational science. He has taught courses in computing from the desktop to supercomputers in Canada, the US, and South Africa. In 2000, as part of the US DoE ASC Flash team, he won a Gordon Bell Award, one of supercomputing’s highest accolades.

Richard "Tommy" Guy is a Ph.D. student in Computer Science at the University of Toronto. While at Wake Forest University, he helped create Verbal Victor, an app to help children with communication difficulties.

Steven Haddock is a Research Scientist at the Monterey Bay Aquarium Research Institute and adjunct Associate Professor at U.C. Santa Cruz, studying bioluminescence and biodiversity of marine zooplankton. He co-authored Practical Computing for Biologists with Casey Dunn.

Mike Hansen is a Ph.D. student in Computer Science and Cognitive Science at Indiana University. His research interests include quantifying the complexity of software using cognitive models of programmers. He has designed and developed software professionally for almost ten years, and enjoys teaching others the skill and art of programming.

Adina Chuang Howe received her Ph.D. in Environmental Engineering. She is currently a postdoctoral research scientist at Michigan State University, where she uses skills learned from Software Carpentry to study microbial communities in the environment.

Katy Huff is a Ph.D. student in nuclear engineering at the University of Wisconsin – Madison, where she helped found The Hacker Within.

Ian M. Mitchell is an associate professor in the Department of Computer Science at the University of British Columbia. His research interests include scientific computing, cyber-physical systems, formal verification, and reproducible research.

Jason Pell is a Ph.D. student in Computer Science and Quantitative Biology at Michigan State University who is primarily interested in tackling large next-generation DNA sequencing datasets. He holds a B.A. in Computer Science from Grand Valley State University.

Jon Pipitone completed his M.Sc. in Computer Science at the University of Toronto in 2010. He has been active since then in a variety of scientific, environmental, and social justice causes.

Mark Plumbley is Director of the Centre for Digital Music (C4DM) at Queen Mary University of London. His research covers analysis of audio and music signals, using techniques such as information theory and sparse representations. Mark leads the EPSRC-funded project SoundSoftware.ac.uk.

Anthony Scopatz has a Ph.D. in Mechanical and Nuclear Engineering from the University of Texas at Austin, and is now a post-doc in the Astrophysics Department's FLASH Center at the University of Chicago.

Jeff Shelton is a Ph.D. candidate in Mechanical Engineering at Purdue University, studying the control aspects of human motion. Following more than two decades in industry, he is interested in aligning educational methods with the evolving societal roles performed by engineers.

Joshua Ryan Smith specializes in electronic devices based on wide-bandgap semiconductor materials and in the past has done work in surface science and nanofabrication. Joshua is a native of North Carolina and received his Ph.D. in physics from North Carolina State University; he learned Python programming in graduate school and has an interest in understanding the design of experiments in terms of the practices of software development.

Nelle Varoquaux returned to university in 2011 after working as a Python software engineer to pursue an applied mathematics degree, specializing in machine learning. She is now interested in using these newly acquired skills to solve biological problems, such as reconstructing the 3D architecture of the genome.

Ethan White is an assistant professor in the Department of Biology at Utah State University. His research focuses on quantitative macroecology, using large ecological databases, advanced statistical methods, and theoretical modeling to understand broad scale ecological patterns.

Greg Wilson started the Software Carpentry project in 1998. He has been a professional software developer, an author, and a university professor. Greg received his Ph.D. in Computer Science from the University of Edinburgh in 1993.

Appendix C. Commissioned Articles Produced for the Instructor's Guide

The History of Education Technology	Audrey Watters
Connectivism	Audrey Watters
Learning Analytics	Audrey Watters
Are American Schools Failing	Audrey Watters
Concept Inventory	Greg Wilson
Learning Styles	Greg Wilson
Stereotype Threat	Greg Wilson
What Does It Mean To Understand Computing?	Greg Wilson
What is Constructivism?	Juha Sorva
Education as Content Delivery	Audrey Watters
Improving Computer Science and Programming Education for Women Improves It for Everyone	Gail Carmichael
What Every Techie Should Know About (20th Century) Education Theory	Audrey Watters
Didactical Models	Laura Hilliger
Old School Educational Theories	Laura Hilliger
Forms of Assessment	Audrey Watters
What's So Hard About Learning to Program?	Juha Sorva
Bloom's Taxonomy	Michael Hansen
Seven Plus Or Minus Two	Michael Hansen
Cognitive Load Theory	Juha Sorva
Three Main Knowledge Types	Laura Hilliger
Polymorphism and Identity	Laura Hilliger
Examples and Self-Explanations	Juha Sorva
Differences Between Novice Performance And Expert Performance	Laura Blankenship
Zone of Proximal Development (Vygotsky)	Laura Blankenship
Direct Instruction vs. PBL and Inquiry-Based Learning	Laura Blankenship
Context and Community Matter (Situated Learning)	Gail Carmichael