

Software Carpentry Evaluation Plan - January-February 2012

(draft)

What do we need?

We need an evaluation process for workshop participants and online learners that will:

1. tell us if the material is having a positive effect on their daily work,
2. let us know what is working well and badly, to help us plan the next iterations,
3. be systematic enough to be applied by researchers or workshop organizers with fairly uniform results, and
4. be lightweight enough that data collection and analysis will not demand excessive resources.

These requirements pose several challenges. First, we will probably need at least two separate evaluation processes: one for workshop participants (who also access the online materials) and another for exclusively online learners. For now, we will only focus on workshop participants that will then go on to do several weeks' worth of online learning; we can then revisit this plan and tackle the other cases.

Second, if we collect our data in a systematic and lightweight manner (say, with a survey), we may miss deeper insights that are more important, but if we focus on qualitatively exploring what's actually important, we won't have uniform data to aggregate (or show to funding agencies or potential participants).

What can we do?

The need for a systematic and lightweight process means that *we will eventually need to use structured interviews or surveys*. On the other hand, previous use of surveys shows that we first need to get in-depth data (through unstructured interviews and lab visits) to "tune" those interviews or surveys. We may also need to re-tune periodically, with each cycle of Software Carpentry.

For the period in which we currently have funding (up to June 2012), we should be able to do four things:

- ***In progress*** - Perform an initial version of an evaluation survey (see next section).
- ***In progress*** - At every workshop or boot camp where a separate evaluator cannot be present, use the new survey (bad data better than no data, and we can at least use the responses as more input to correct the survey itself). After a test run, request similar feedback from online participants.
- In parallel, for a couple of workshops (preferably close to the North American West Coast, if Jorge is going to do this), have an evaluator perform pre- and a post-workshop

interviews and lab visits with confirmed participants. These visits will ideally be at least a couple of days long, and will involve observations and interviews. They have two goals. First, to document the computing practices of the participants and of their labs, before and after the workshop. (The “after” should probably be at least a few weeks from the workshop’s end.) Second, to compare the state of computing practice at these labs and the gains from the workshop with the self-reported data from their surveys. This will help us understand the problems with the new version of the survey, to keep iterating it.

- Finally, once we set up a way for online learners to authenticate and collect badges, we can add a checkbox on sign-up for them to indicate if it is OK for us to contact them to help us evaluate the effectiveness of our material. For those who agree, we can initially contact over email, and either arrange for a chat session or a Skype call in which an evaluator can interview the participant with questions like those from the field visits above.

As a result, by the end of this period we should have (1) quantitative and qualitative data from the revised survey on the impact of SWC, (2) qualitative data from visits and in-depth interviews that will be much stronger than testimonials, and (3) a better survey for the next iteration of this project.

Initial evaluation survey

The survey questions are supposed to cover these topics. Keep in mind that the survey should be fairly short.

Topics

- State of your practice (*what do you use?*)
- State of your skill level (*how good are you?*)
- State of your perception of the usefulness of these skills (*how valuable is this?*)
- State of your reach (*what can't you do that you'd like to?*)
- Reasons for inaction
- (*After the workshop/training*) Estimate of time gained or wasted

Survey Questions

Roughly how many hours do you work per week? How many of those do you spend creating, modifying, or testing software?

Indicate your level of use of each of the following tools or techniques [*Options: (a) do not use, (b) occasionally, (c) routinely*]

- Version control
- Shell commands
- ...and so on for all of the topics in our lectures, whether discussed in the workshop or not

[For each topic, ask four questions in the format: “Do you understand the following <topic> command well enough to explain them to someone else? Yes / No” The four questions should be one each of:

- An advanced beginner command/technique seen in the workshop
- An advanced beginner command/technique *not* seen in the workshop
- A competent or proficient command/technique seen in the workshop
- A competent or proficient command/technique *not* seen in the workshop]

For example (taken from the survey done for STScl students):

- Shell commands: can you explain
 - `ls data/*.txt`
 - `find ~ -name '*.py' -print`
- Version control: can you explain
 - `svn update`
 - `svn diff -r 1723`
- Python: can you explain
 - `x = {'east' : 5, 'west' : 11}`
 - `__init__(self, *args)`
- Testing: can you define
 - fixture
 - mock object
- Make: can you define/explain
 - dependency
 - `cp $< $@`
- SQL: can you explain
 - `select * from data where data.left < data.right;`
 - inner join

Indicate how hard would it be for you to get your work done without each of these: [Options: (a) *I don't know what this is*, (b) *unimportant*, (c) *marginal*, (d) *important*].

- Version control
- Shell commands
- etc...

Do you have research goals that you cannot attain because of a lack of computational or programming expertise? If so, please elaborate.

For those topics or skills that might make your work easier, but on which you have not become proficient: why is this the case?

(After) Estimate how many hours have you gained or lost weekly as a result of the workshop. If there were net gains or losses, please elaborate.

(After) Can you think of computer-related things that you're doing differently than you used to since taking the workshop? If so, please elaborate.

(*After*) What scientific results have you produced (or are you producing) that would have been impossible or out of reach before this training?