FOAR705 Week 4

Brian Ballsun-Stanton | Shawn A Ross | Kathryn Elliot

Faculty of Arts

Friday 23 August 2019

Minute card reflection

Software Carpentry: Shell

Minute cards!

References

Minute card reflection

Most frequent red sticky note theme was: "I'm not super clear on what tasks are necessary each week in some cases in relation to what we are marked on."

In Cloudstor, Assignments, Week4Homework:

"Expectations:

You have two things to work on this week:

Work on: http://swcarpentry.github.io/shell-novice/ and do episodes: Introduction, Navigating Files and Directories, Working with Files. Put results into a new (or new section, or otherwise demarcate them) learning journal.

Finish Elaboration I. This is due in cloudstor only before the Week 5 class (though it will form part of your Elaboration II submission). Commit to git as you are committing all assignments, in whatever organisation you choose."

Focus on the assignments to meet expectations. Do readings if possible.

Elaboration is assessable (combined parts I & II). Worth 15% of the proof of concept marks.

Shell Learning Journal, due as Week 6 participatory task.

Copying the cloudstor Week*X*Homework documents to the week by week lines in iLearn.

Using the step-by-step breakdown of your problems / solutions developed last week, identify technologies (programming languages, software libraries, APIs (Application programming interfaces, the means by which one program talks to another program) and other components of a modern data collection or processing workflow) that could accomplish each step. You may want to specify more than one possible technology.

This document can be written in any format (though we encourage further practice in LaTeX) and should be as long as it needs to be. If you find yourself testing too many steps or technologies which depend on each other, you may wish to adjust your scope to be more specific. The judgement call for "too many" is "Can you see yourself learning enough to use and connect these tools and techniques in the next month?"

Note: You will be testing these things for your next assignment (Elaboration II). Figure out the least work possible to get these things to fail your expectations. Plan alternate routes, assuming that most things will not work the way you expect them to.

This elaboration planning document is due on Cloudstor by the start of class, week 5. You will use it as part of your graded Elaboration II submission, week 6.

Read the beginning of Chapter 7 in , through 'Elaboration and Iterations' (plus any subsequent sections where you think you need more detail).

The goal of the assignment is to:

- get a more detailed understanding of requirements;
- start charting ways to deliver those requirements;
- mitigate risk / de-scope as necessary (fail early and often);
- onfirm your earlier scoping analysis, ensuring that the tool or workflow you are developing does something you need.

The hardest part will probably be finding and choosing technologies. Doing so will likely require consultations with us. Please attend consultation hours and use Slack.

- Social science: build a pipeline to get data from a public source (AURIN, ADA, data.gov.au, etc.) or from your own social surveys using an API and do something with that data (and then maybe resubmit the improved dataset if applicable).
- Ethnography / oral history: adapt an existing metadata capture mobile application to your needs.
- Textual or Media studies: build or adapt a system to manage metadata about the items in your corpus.
- Textual research: develop a text analysis tool using Voyant for your corpus.
- Archival research: build a pipeline around the management of your archival sources (e.g., photographs of them) using Zotero, Tropy, or other open-source tools.
- Cross-disciplinary: develop a full-service, redeployable bibliography management pipeline around one of the open managers (e.g., Zotero) for your thesis; develop a redeployable TeX pipeline / template for your thesis.

Installing the shell:

`https://swcarpentry.github.io/shell-novice/setup.html`

Mac users, go into spotlight, launch terminal.

Windows users download Git For Windows from
`https://gitforwindows.org/`

(other methods out of scope for today, ask during consultation)

At a high level, computers do four things:

- run programs
- store data
- communicate with each other, and
- interact with us

The Graphic User Interface (GUI): Mouse, windows, 'drag and drop'. 'Knowledge-in-the- world'. [Norman, 2013]

Figure 1: Blake Patterson https://flic.kr/p/4D6hmv CC-By

The Command Line Interface (CLI): Empty screen with a $.
'Knowledge-in-the-head'[Norman, 2013]. Type instructions and computer executes quickly.

'The teletype would send that line to the computer, which might or might not respond with some lines of its own, which the teletype would hammer out–producing, over time, a transcript of your exchange with the machine...' [Stephenson, 1999]
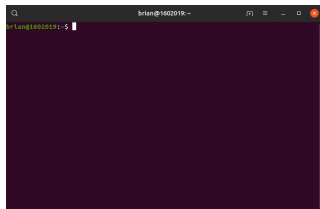

Figure 2: Brian Ballsun-Stanton. CC0

Let us look at the prompt now...

Nelle Nemo, a marine biologist, has just returned from a six-month survey of the North Pacific Gyre, where she has been sampling gelatinous marine life in the Great Pacific Garbage Patch. She has 1520 samples in all and now needs to:

1) Run each sample through an assay machine that will measure the relative abundance of 300 different proteins. The machine's output for a single sample is a file with one line for each protein.

2) Calculate statistics for each of the proteins separately using a program her supervisor wrote called goostats.

3) Write up results. Her supervisor would really like her to do this by the end of the month so that her paper can appear in an upcoming special issue of Aquatic Goo Letters.

Two weeks to run 1520 samples.

Using GUI would take 1520 different file selections. Finishing analysis would take longer than the month available.

12

Episode 2, Navigating Files and Directories:

How can I move around on my computer?

How can I see what files and directories I have?

How can I specify the location of a file or directory on my computer?

Minute cards!

On your green sticky, write one thing we did well today.

On your red sticky, write one thing we could improve upon for next time. Be specific.

References

Norman, D. (2013).
The design of everyday things: Revised and expanded edition.
Basic books.

Stephenson, N. (1999).
In the Beginning...was the Command Line.
William Morrow Paperbacks, first edition edition.

Thank you!

Source code for this presentation is available at:
`https://github.com/MQ-FOAR705/MQ-FOAR705-Week3`