



MACQUARIE
University
SYDNEY • AUSTRALIA

FOAR705 Week 2

Brian Ballsun-Stanton | Shawn A Ross | Kathryn Elliot

Faculty of Arts

Friday 09 August 2019



The Learning Journal

Data Carpentry

Research design 101

Proof of Concept scoping 101

Project management 101

References



MACQUARIE
University
SYDNEY • AUSTRALIA

The Learning Journal

- To serve as a Laboratory Notebook – a record of:
 - Thoughts
 - Intentions
 - Results
- Mechanism for showing your work
- Reminders of common mistakes and solutions
- Demonstration of growth over time

Percentages:

- Exercise Documentation (70)
- Committing your work (10)
- Error Reflection and Solution (20)

Dates:

- Spreadsheets Learning Journal (with all exercises complete) due before class, 23 August
- Shell Scripts learning Journal due (with all exercises) before class, 6 September
- Open Refine learning Journal due (with all exercises) before class, 4 October
- R Learning Journal Due (up to dplyr and tidyr) before class, 8 November

All technical work outside of class, and all Carpentry exercises should be recorded in a document in cloudstor.

For each discrete action taken. (Exercise, part of exercise, command run, code changed, code added, etc...)

- The intention of the action: “What do you intend to be the result of the action you are about to take”
- The specifics of the action taken: Timestamp, commands or actions
- Results: note success or failure, screenshot or copy-paste or summary, error states
- Marginal notes for improvements on what to do or how to think about the idea more effectively

Documentation of errors is critical for success



Ana Stevanovic 12:13 PM

Objective: Recovering older versions of a file

Action:

to recover older files you use the `git checkout` command.

the file name is `data_cruncher.py`

she can use `HEAD` to get the previous commit or she could use the unique ID of the commit she wants

to recover her file she would have to use the `git checkout HEAD data_cruncher.py` command or alternatively she could use `git checkout <unique ID>`

Error: none

Result: option 5: both 2 and 4 is correct



Ana Stevanovic 12:22 PM

Objective: Reverting a commit

Action:

in this example

1. _____ # Look at the git history of the project to find the commit ID
2. Copy the ID (the first few characters of the ID, e.g. 0b1d055).
3. `git revert [commit ID]`
4. Type in the new commit message.
5. Save and close

in step 3 she has not mentioned which text file after the commit ID
after step 5 she needs to commit the new changes

Error: not too sure how to use `git revert` - not mentioned in the notes

Figure 1: Screenshot from a Software Carpentry Learning Journal

- Errors are documented along with steps to recover, just like any other action in the lab notebook.
- All entries have an objective articulated before they are run.
- Commands and code are clearly indicated for future reference.
- Results are clearly documented, and contain a minimal amount of self-reflective feedback to inform the next attempt.

For Carpentries exercises or other experiments (code not directly related to your proof of concept), make a repository (public or private) inside the MQ-FOAR705 organisation on Github. Commit code or outputs (reorganised sheets, etc) to that repository.

An HD:

- Commit messages are clear and have useful descriptions, not just summary lines.
- Files and directories are organised consistently in a fashion which allows for easy command line navigation and sorting (no spaces or other problematic characters).
- All work, not just technical, is committed to an appropriate repository.

Document your errors and how you've found solutions.

An HD:

- Using the error documentation in the laboratory notebook as the basis, a library of common errors, solutions, and ways to find solutions is built for future reference.
- These solutions contain links to good internet resources, and useful rules of thumb for where best to find assistance.



MACQUARIE
University
SYDNEY • AUSTRALIA

Data Carpentry

Shared notes document for the entire unit. Put questions, thoughts or observations from readings or prior class onto page 2. The person who isn't lecturing will try to answer them inline.



Figure 2: Cloudstor: <http://bit.ly/2YW10wm>

Guiding question for this episode:

What are basic principles for using spreadsheets for good data organisation?

In cloudstor shared document:

- How many people have used spreadsheets in their research?
- How many people have accidentally done something that made them frustrated or sad?

livestock_owned_and_numbers
1, (poultry)
3, (oxen , cows)
1, (goats)
4, (oxen , cows)
10, (oxen , cows , goats , poultry)
1, (goats)
1, (oxen)
2, (oxen , goats)
3, (oxen , goats)

Figure 3: Data Carpentry: combined info (CC-BY)

poultry	cows	goats	oxen
1	0	0	0
0	2	0	1
0	0	1	0
0	3	0	1
5	2	2	1
0	0	1	0
0	0	0	1
0	0	1	1
0	0	2	1

Figure 4: Data Carpentry: single info (CC-BY)

We're going to take a messy version of the SAFI data and describe how we would clean it up.

- Download the messy data.
- Open up the data in a spreadsheet program.
- Notice that there are two tabs. Two researchers conducted the interviews, one in Mozambique and the other in Tanzania. They both structured their data tables in a different way. Now, you're the person in charge of this project and you want to be able to start analyzing the data.
- With the person next to you, identify what is wrong with this spreadsheet. Discuss the steps you would need to take to clean up the two tabs, and to put them all together in one spreadsheet.
- Document your group's thoughts in the cloudstor shared document

“Data about data”

Exercise (maybe):

Download a clean version of this dataset and open the file with your spreadsheet program. This data has many more variables that were not included in the messy spreadsheet and is formatted according to tidy data principles.

Discuss this data with a partner and make a list of some of the types of metadata that should be recorded about this dataset. It may be helpful to start by asking yourself, “What is not immediately obvious to me about this data? What questions would I need to know the answers to in order to analyze and interpret this data?”

- Web interface (demonstrate now)
- `desktop.github.com` client
- other clients

Exercise: commit the results (a text file containing your thoughts and the original data) of the cleanup exercise to your own repository on Github now.

Before class next week, in your learning journal, finish reading to “Formatting Problems” and document the two exercises we’ve (hopefully) done today in your Github repository. Also in your learning journal, find an example of each problem in data produced by your discipline. (Bonus points if you can find these problems in published datasets in your discipline).

5 minute break



MACQUARIE
University
SYDNEY • AUSTRALIA

Research design 101

Robust research requires articulation of an explicit research design

- Deductive vs. Inductive vs. Abductive
- Idiographic vs. Nomothetic

Each type is valuable, but you must recognise what you are doing and not conflate them.



MACQUARIE
University
SYDNEY • AUSTRALIA

Proof of Concept scoping 101

To ensure that a product actually solves user problems, software development begins with 'Business Analysis' (BA), where user requirements are enumerated.

But how do we gather requirements and act on them to produce a solution?

It is surprisingly difficult to learn what clients (even yourself!) really need and want



Figure 5: Archaeologists contemplate data standards (FAIMS Stocktaking, 2012)

Approach #2: Ask or observe what people do (facts)

Borrowed from 'Lean startup' methodology. All materials are on Cloudstor or available from <https://www.strategyzer.com/> (free registration required)

The overview document is the 'Value Proposition canvas', but see also the 'Mission Model canvas' for an indication of how the approach can be applied outside tech industry settings.

See the 'Customer Profile' worksheet

- Identify 'jobs' (see 'A day in the life worksheet')
- Identify 'pains' (see 'Customer Pains trigger questions')
- Identify 'gains' (see 'Customer Gains trigger questions')

After you have completed the 'Customer Profile' worksheet, look at the 'Value Map' worksheet

- Identify 'pain relievers' that map to pains (see 'Pain Relievers trigger questions')
- Identify 'gains creators' that map to gains (see 'Gain Creator trigger questions')
- Finally, you can articulate 'Products and Services' - what you are going to build



MACQUARIE
University
SYDNEY • AUSTRALIA

Project management 101

So you have great ideas, what next?

Approach #1: Top-down design ('Waterfall')

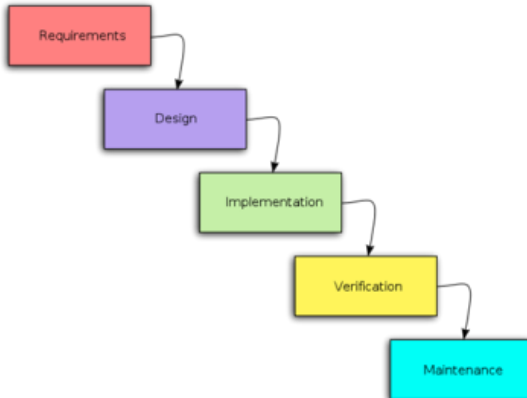


Figure 6: Traditional and linear approach

[Parody, 2018]

Approach #2: Iterative design with course corrections ('Agile')



Figure 7: Design-test-repeat approach to PM

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more. [Atlassian, 2019a]

Tool #1: Gantt chart

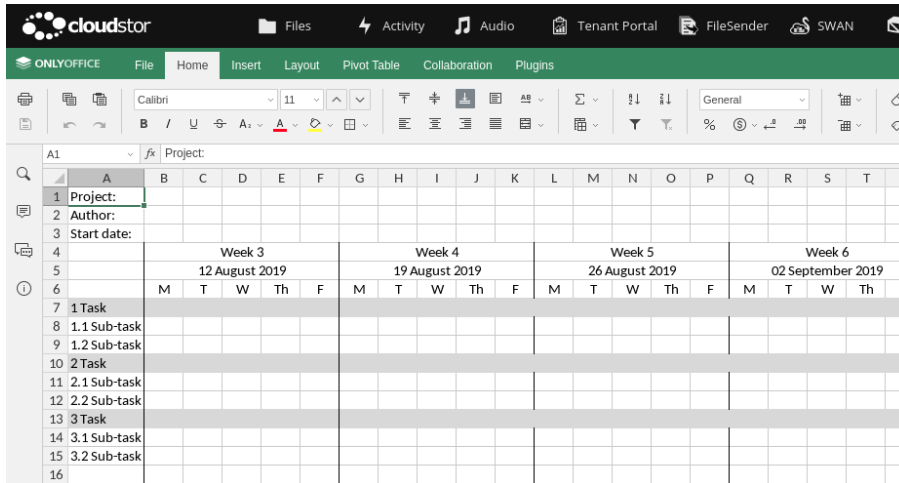


Figure 8: A Gantt Chart template (on Cloudstor)

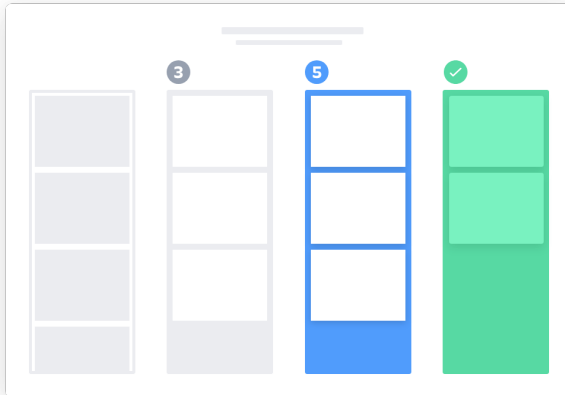


Figure 9: Schematic Kanban board [Atlassian, 2019b]

- Visualise your work
- Limit work in progress

Kanban boards often have columns like: Backlog (wish list), To do, In progress, Done, with the 'To do' and 'In progress' columns having work limits (e.g., 3-5 tasks).

Trello is a popular application for Kanban. Atlassian has Kanban learning materials online [Atlassian, 2019b].



MACQUARIE
University
SYDNEY • AUSTRALIA

References



Atlassian (2019a).

Is the agile manifesto still a thing? | atlassian.

[https:](https://www.atlassian.com/agile/manifesto)

[//www.atlassian.com/agile/manifesto.](https://www.atlassian.com/agile/manifesto)

Accessed: 2019-8-9.



Atlassian (2019b).

Kanban - a brief introduction | atlassian.

[https:](https://www.atlassian.com/agile/kanban)

[//www.atlassian.com/agile/kanban.](https://www.atlassian.com/agile/kanban)



Accessed: 2019-8-9.

Parody, L. (2018).

How to manage modern software projects: Waterfall vs. agile.

[https://medium.com/@lizparody/waterfall-vs-agile-methodology-in-software-development-1e19ef168cf6.](https://medium.com/@lizparody/waterfall-vs-agile-methodology-in-software-development-1e19ef168cf6)

Accessed: 2019-8-9.

Source code for this presentation is available at:

<https://github.com/MQ-FOAR705/MQ-FOAR705-Week2>

This work is licensed under a Creative Commons Attribution 4.0 International License.