11

Enhancing student research projects with new technologies

Darach Watson

Niels Bohr Institute
University of Copenhagen

Summary. A critical component in supervising thesis students is helping them maintain focus on the final outcome and continue working consistently for the full project duration. However, it can be difficult for a supervisor to gather concrete data on the progress of a student throughout the project period. Poor outcomes can occur in the middle part of the project as initial excitement flags, complexity grows and the supervisor is typically busy with other tasks. I report here on a test of a coupled set-up of an integrated rich text, code, and data application, a web-based version control system, and cloud-based team collaboration tool, designed to improve consistency of overview of the progress of thesis students. The tools may also increase the rapidity and quality of communication and help maintain consistent supervision and focus for the student over the full project period. I used Jupyter Notebooks, Github, and Slack respectively as a supervision tool with a Master's student over a full year. Quantitative data from these tools are used to analyse how well the monitoring set up worked. The technical set up works well. There is an initial investment cost for the student to learn the various tools and from the supervisor in ensuring the student gets used to using them consistently and correctly. However, the overall benefits appear to be significant in terms of the supervisor's overview of the student's progress, leading to potentially greater consistency of supervision. Secondary effects were also positive, with significant development of transferable skills such as software coding ability and team-working. Modifications were made to improve the use of the set up for the following year and the initial data on a student in the next year are provided.

Introduction

Master's students at research-intensive institutes, or going on to research-related careers will often do a year-long Master's thesis based on original research. Long theses are particularly prone to loss of focus and motivation (Litalien et al., 2015), as well as feature- and content-creep. Without resolution, the thesis can be severely delayed, or, ultimately, never completed (Bair et al., 2005), both of which have serious negative consequences for both the student and the hosting department (Wendler et al., 2010).

The same problems can occur in any thesis work, and a significant issue for any supervisor is keeping track of the progress of a student, in particular a student pursuing research work, where there is no well-defined curriculum. One solution to this problem is to curtail the duration and ambition of the thesis project. However, for the ambitious student or supervisor, such solutions are either sub-optimal or simply not acceptable, or limits the type of projects that can be undertaken.

The other solution is to increase the amount or quality of feedback and engagement (Price et al., 2011) either through group work or directly from the supervisor. Research undertaken in a group or as a team can help mitigate the problems of oversight and motivation and focus loss (Hertel et al., 2000), but such group working may not be available in every situation or type of research. On the other hand, in an ambitious research milieu, significantly increasing the amount of supervision time in a simple way is not typically viable, may not be necessary or desirable where the student is doing well and beginning to behave independently, and is certainly not a scalable solution. It may, on the other hand, be possible to use a technological solution to keep tabs on the progress of a student and flag a drop-off in activity, as well as increase the speed, quality, and quantity of communication with a student, without substantially increasing the workload on the supervisor.

In this paper, I examine the integrated use of tools for recorded, interactive data analysis and coding, web-based sharing and versioning control, and collaboration work, to keep track of the progress of thesis projects, and communicate effectively and quickly with research students in the physical sciences.

Method

I supervised two Master's students in consecutive academic years using the following communication methodology. Meetings were arranged twice every week, where this was possible, lasting 30 to 60 minutes. The students were requested to do coding and data analysis using Python primarily with the *Numpy*, *Scipy*, *Matplotlib* and *Astropy* libraries in Jupyter Notebooks and to keep notes of their work and reading in the same format. They should then use the Git software and the Github website to maintain versions of their notes, code, and analysis that are shared with their supervisor. Finally, their Github repository is integrated with the Slack team communication software, which is used as the primary electronic communication tool. Any queries, concerns, or brief explanations can be requested through Slack, and notifications of updates to the student's work are made automatically to the Slack channel associated with the project. The software above is available for free. Advanced features are available from Github and Slack with a paid account.

Jupyter Notebooks[1]

Jupyter Notebooks is a format for writing code and doing data analysis primarily in the languages Julia, R, and Python. My students were using Python. Jupyter Notebooks works as an interactive coding and data analysis environment running in a web browser that executes blocks of code in individual cells, allowing an interactive environment coupled with a record of the code, so that it can be altered and tested on the fly. It also has the capability to produce and include plots (typically using a library like *Matplotlib*), and to include notes written in Markdown or LATEX, with images. Jupyter Notebooks therefore enables the recording of a really comprehensive and comprehensible record of data analysis, code testing, and research notes. Part of a Jupyter Notebooks is shown in figure 11.1.

Github[2]

Git is a version control system, recording in a repository each version, the changes from the previous version, and a note of what the changes were.

[1] See https://jupyter.org for more details.
[2] See https://github.com

Github is a website that acts as a remote storage for Git repositories. These repositories can be shared and changes made cooperatively in a team as required.

Slack[3]

The team collaboration software Slack is widely used and has many integrations with other software systems. Here a channel was set up for the project and all communications related to the project were made on this channel, providing a record of conversations, advice, and feedback, as well as permitting rapid interchange on small questions. The Github plugin for Slack was used, which sends a notification and a link to the relevant channel automatically when a new version was uploaded to the repository. Part of a Slack session is shown in figure 11.2.

Results

The students with whom this method was trialled, B, and D, were instructed in the approach at the first meeting before they selected their supervisor. Both were happy with the approach. Neither had significant coding experience, with very little Python expertise and no experience with the software tools used above.

[3] See https://slack.com

```
          nan  702.87901075  769.77710622  706.10267365  725.30769673
 716.53100115  868.78108939  749.52884503  721.93173962  748.65775911
          nan           nan  724.86214842  464.0528132   798.43240415
 764.44413021  695.55462346  265.28876448  711.63823009 1089.33309065
 833.86514294  738.76507834  720.26517259  760.18054984  804.27784863
 709.44057924           nan           nan  754.73386325  695.47669182
 783.29469516  784.83046394  691.95385231  348.65554208  853.62868951
 729.05843783  538.01579983           nan  611.50891067           nan]
```

```
In [ ]:
```

```
In [17]: plt.figure(figsize=(10,7))
plt.errorbar(comp_array[:,0],comp_array[:,1],yerr=noise_quasar_1[:,1],fmt='o',capsize=3)
plt.errorbar(comp_array[:,0],comp_array[:,2],yerr=noise_quasar_1[:,2],fmt='x',capsize=3)
plt.errorbar(comp_array[:,0],comp_array[:,3],yerr=noise_quasar_1[:,3],fmt='*',capsize=3)
#plt.ylim([3*10**(-15),0.45*10**(-14)])
plt.show()

plt.figure(figsize=(20,15))
plt.errorbar(comp_array[:,0],comp_array[:,1],yerr=noise_quasar_1[:,1],fmt='o',capsize=3,color='b',
label='STA1')
plt.errorbar(comp_array[:,0],comp_array[:,2],yerr=noise_quasar_1[:,2],fmt='x',capsize=3,color='r',
label='STA2')
plt.errorbar(comp_array[:,0],comp_array[:,3],yerr=noise_quasar_1[:,3],fmt='*',capsize=3,color='g',
label='STA3')
plt.errorbar(comp_array[:,0],comp_array[:,1],yerr=noise_quasar_1[:,4],fmt='o',capsize=8,capthick=2
,color='b')
plt.errorbar(comp_array[:,0],comp_array[:,2],yerr=noise_quasar_1[:,5],fmt='x',capsize=8,capthick=2
,color='r')
plt.errorbar(comp_array[:,0],comp_array[:,3],yerr=noise_quasar_1[:,6],fmt='*',capsize=8,capthick=2
,color='g')
plt.legend()
#plt.ylim([3*10**(-15),0.45*10**(-14)])
plt.show()

plt.figure(figsize=(10,7))
plt.scatter(comp_array[:,0],comp_array[:,1])
plt.scatter(comp_array[:,0],comp_array[:,2])
plt.scatter(comp_array[:,0],comp_array[:,3])
plt.ylim([6*10**(-15),1.5*10**(-14)])
plt.show()

np.savetxt('../NOVEMBER/NGC3783_NOISE_Ks.txt',noise_quasar_1)
np.savetxt('../NOVEMBER/NOV-NGC3783-Ks',comp_array)
np.savetxt('../NOVEMBER/NGC3783_SNR_Ks.txt',SNR_part_1)
```

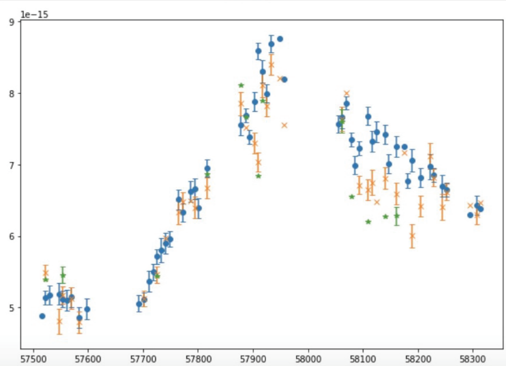

Fig. 11.1. Section of a Jupyter notebook on Github from student B.

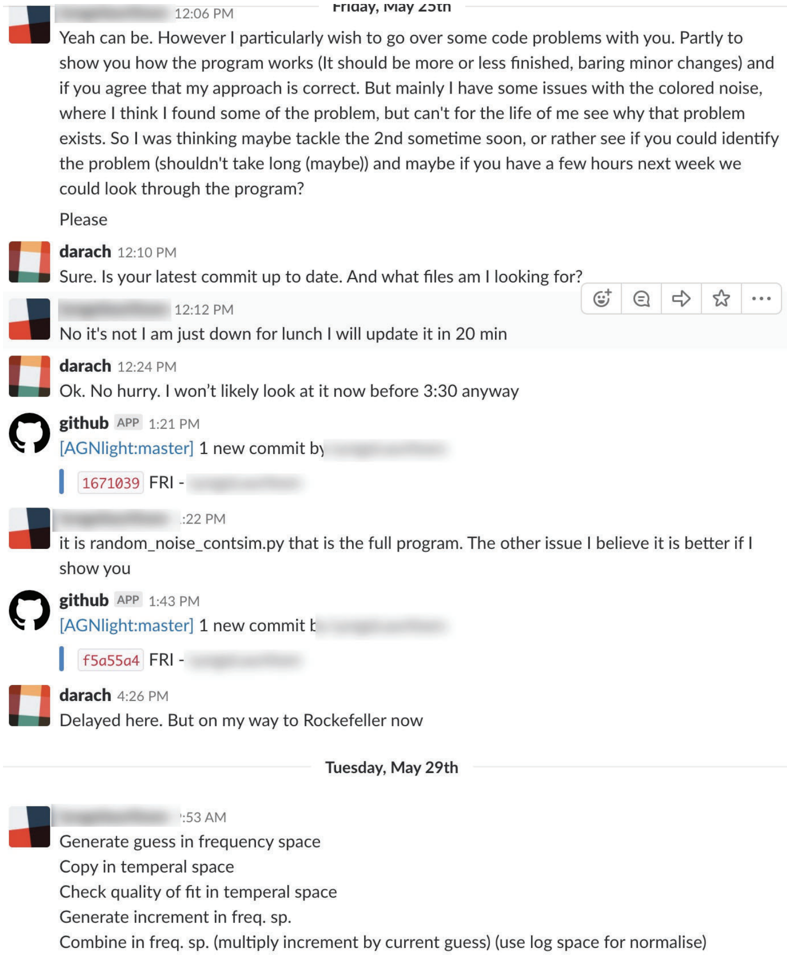

Fig. 11.2. Slack session with student B.

Table 11.1. Students' Github commit days, commits with useful comments, and Slack messages.

	Student B			Student D		
Project Month	No. Github commit dates	(commented commits)	No. Slack messages	No. Github commit dates	(commented commits)	No. Slack messages
1	0	(0)	7	0	(0)	1
2	0	(0)	8	6	(6)	3
3	9	(0)	22	3	(1)	0
4	7	(0)	4	–	–	–
5	2	(0)	3	–	–	–
6	1	(0)	1	–	–	–
7	2	(0)	14	–	–	–
8	1	(0)	3	–	–	–
9	2	(0)	12	–	–	–
10	1	(0)	11	–	–	–
11	1	(0)	5	–	–	–
12	7	(0)	25	–	–	–
13	1	(0)	47	–	–	–

Observations

Activity can be monitored using the number of Github commits (a commit saves a snapshot of the current document(s) or code to the repository), avoiding some of the problems of subjective impressions. As seen from the data in table 1, B got properly started with Github in two months after starting the thesis project. Commits were made frequently in the early months, with fewer in the middle period, and then many more in the last month. The numbers of Slack messages from B followed a similar pattern.

This strongly suggests that after a strong start there was a lull in activity or recording of activity and communication after the first semester, with a similar number of communications and 80% more Github commits in the last four months of 2017 compared to the first seven months of 2018. This was followed by a strong surge in activity in both Github commits and Slack messages in the last six weeks of the project.

My own observations agree with the quantitative data, that student B's progress slowed considerably after the first four months and only really picked up again at the end of the year. The reason for this appeared to be that B became lost in the fine detail of analysing the data. While at the time each incremental step seemed necessary and plausible, looking at the data, it seems that there is a case to be made for a more vigorous supervision intervention.

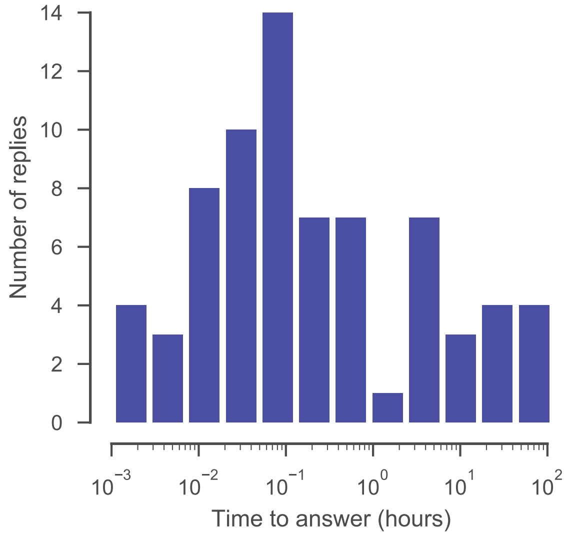

Fig. 11.3. Time to rst reply to a student query on Slack.

The answer lag time is a key parameter in the responsiveness of a supervisor. We can measure that parameter on Slack by looking at the time between requests by the student and answers from the supervisor. These data are shown for student B in figure 11.3. The data are plotted on a logarithmic scale, showing a fairly even response per decade. More than 90% of responses are within 24 hours. However, the median response time is only 5.8 minutes.

Other outcomes of using the system were that students B and D became moderately proficient with Python programming and version control tools.

Discussion

The commit and Slack chat frequency data show clearly a drop-off in activity in the middle of the project period as anticipated from previous students and consistent with direct observation. This indicates that using this combined technique can be successful in monitoring the progress of students doing work without a well-defined curriculum.

These observations show that using this technological approach without strong control of the process did not entirely correct the problem of a loss of focus and motivation in the middle of the project for student B. Monitoring the number of Github commits and the Slack communication statistics over timescales of weeks or months could allow the supervisor to intervene to correct the motivation and focus of the student.

The lag time analysis indicates that using a communications platform like Slack does offer very fast responses to student queries. This may be due to the direct notifications to mobile phone that Slack offers via the mobile app, the relatively informal method of communicating, encouraging quick, short questions, with a chat-like interface, as well as the easy inclusion of plots and data and integration with Github.

This approach was trialled on students working with the analysis of physical science data and theoretical models. Jupyter Notebooks is particularly well suited to this type of work, and Github is designed especially for computer code versioning. However, there is no obvious reason why this approach would not work well in areas that do not require data analysis tools. Jupyter Notebooks is certainly very capable of fully marked up notes including figures. And Github can maintain versions of files even in binary formats, though it works best with plain text formats. However, the method outlined here: work-recording software integrated with a shared versioning system and a team communications software, is not exclusive to the software used here and could be used with other tools. There seems no obvious barrier to using this technology in areas requiring lab work, interviews, fieldwork, or literature review. However, certainly other tools exist some of which allow for document preparation and version control together. One of the easiest and most robust is Google Docs[4], which automatically handles versioning cleanly and elegantly and might be suitable in less technical situations.

It was expected that using the approach outlined in this paper would also improve familiarity with coding in Python, with structuring code well, with versioning, and with the basic mechanics and sharing mindset required for really effective collaboration. These aims were achieved with the exception of well-structured code.

[4] See https://docs.google.com

Adaptations

While this methodology makes insisting on well-structured code and keeping clear, comprehensive notes straightforward, this was not achieved very successfully with student B. These issues led to a few problems during the research project that this method is designed to solve. My current conclusion is that tighter control of and better feedback to the early versions of the student's commits would solve this problem. This conclusion will be tested in the future.

The following adaptations have so far been made to the programme. First, the use of Github for version control was insisted on early. This took some additional supervision effort, however, the first commit was a month earlier than the previous year. Second, note-taking using Jupyter Notebooks was also insisted on, resulting in early use of Jupyter Notebooks and extensive, detailed notes being posted, which was not the case the previous year and had been at least partly responsible for the rushed end period of that project. Fewer commits have so far been made and fewer lines of code submitted, however, with student D. I believe this is largely due to less manual versioning done by the student, a positive development in coding practice.

Conclusions

In this paper I have laid out a technical framework to monitor the progress of students undertaking research projects, to maintain student focus over the course of the project, and to increase the speed and quality of communication. The framework was successful in keeping track of how the students work developed and increased the rapidity of communication. So far, the method has not been completely successful in maintaining focus over the full project period or in improving the quality of code structuring. With a greater focus on the feedback of early commits, coupled with monthly monitoring the commit and communication statistics, these goals may also be achieved.

Acknowledgements

I would like to thank Anja C. Andersen and Camilla Østerberg Rump for useful feedback on this paper.

References

Bair, C. R., & Haworth, J. G. (2005). Doctoral student attrition and persistence: A meta-synthesis of research. In *Higher education: Handbook of theory and research* (pp. 481-534). Springer, Dordrecht.

Hertel, G., Kerr, N. L., & Messé, L. A. (2000). Motivation gains in performance groups: Paradigmatic and theoretical developments on the Köhler effect. *Journal of personality and social psychology, 79*(4), 580.

Litalien, D., Guay, F., & Morin, A. J. (2015). Motivation for PhD studies: Scale development and validation. *Learning and Individual Differences, 41*, 1-13.

Price, M., Handley, K., & Millar, J. (2011). Feedback: Focusing attention on engagement. *Studies in higher education, 36*(8), 879-896.

Wendler, C., Bridgeman, B., Cline, F., Millett, C., Rock, J., Bell, N., & McAllister, P. (2010). The path forward: The future of graduate education in the United States. *Educational Testing Service*.