
Coursework commentaries 2016–17

C03320 Preliminary project report

General remarks

The primary purpose of the Preliminary project report (PPR) is to encourage students to begin thinking about, and working on their Projects at an early stage of the year.

The general standard of this year's PPRs was good, with an average mark of approximately 57 per cent. The pass rate was very high at over 96 per cent; a little higher than the previous year.

Individual feedback is provided to each student within a few weeks of the submission deadline. The more information a PPR contains about work done to date, problems encountered and future plans, the easier it is for staff to provide helpful feedback.

Please note that the PPR is expected to conform to a standard structure, as specified in the current edition of the **C03320 Project** subject guide. The subject guide specifies that the report should be around 2,000–4,000 words. The examiners are looking for evidence of a student's ability to write clearly and concisely, and their ability to judge what information should be included and what is irrelevant. Approximately 15–20 pages are usually about right for the PPR. Note that the PPR must be submitted as a **PDF document** (not Word or any other format). This is clearly stated on the online submission page on the virtual learning environment (VLE), and yet every year a small number of students ignore this!

Some of the common weaknesses seen in this year's PPRs were:

- failure to identify an appropriate question to address or aim to achieve
- inadequate literature review (including poor referencing and citation)
- poor Project plan (including lack of thought about testing and evaluation).

Each of these topics is discussed below. Note that these three issues come up every year in the PPRs. If you are reading this report at an early stage in deciding on your Project and take time to ensure that each of these issues is addressed, then you should be in good shape for producing a strong PPR.

Failure to identify an appropriate question to address or aim to achieve

Ideally, the examiners are looking for a Project to address a specific problem by following the structure of an academic research Project: namely, identifying a specific question to be addressed; proposing a means of answering that question (which may entail proposing a solution to an identified problem); performing some sort of experimental data collection relating to the proposed means of answering the question; analysing the collected data; and drawing conclusions from the analysis which relate back to the original research question.

Projects which merely involve the implementation of a piece of software or website, with no academic question driving the development will struggle to achieve the highest grades; although such Projects can

receive good marks if approached in the right way. In order for such Projects to be acceptable, they must demonstrate the application of solid software development practice, including requirements gathering, design, implementation, testing and evaluation. Even a Project which on the face of it is a straightforward software development task can be cast as an academic research Project, if appropriate questions can be addressed: for example, ‘Can novel feature X improve some aspect of a business process?’; ‘Can novel user interface feature Y improve customer satisfaction with the system?’ The more specific a question can be framed and the more specific the means of analysis, the easier it will be to provide a definitive answer to it in the Project.

A small number of PPRs seemed to offer no original contribution from the student. The nature of these Projects generally involved the student ‘finding out about’ a subject, and reporting what had been found. Projects that involve no software implementation are particularly prone to this weakness. The examiners are looking for a Project that shows you putting the techniques and knowledge learned during your studies into practice. For a good Project, the examiners are really looking for an original idea and/or something beyond what has been read in books or other sources. This ‘extra’ contribution might be very small, but will evidence something beyond just reading and reporting. For Projects with no software development, it is particularly important that what is written is derived from reliable sources of academic knowledge (e.g. journal articles and conference proceedings) and where appropriate, from relevant texts on industry best practice. Furthermore, it is essential that Projects of this nature have a strong CIS/CC flavour and have some relation to the content of at least one of the courses you have studied.

Having identified a suitable Project area, some students still failed to set out their aims and objectives clearly. The **aims** of the Project describe the broad overall purpose and desired outcomes of the work; the **objectives** describe the concrete steps you intend to take to achieve your aims. The more specific you are about your aims and objectives at the start, the easier it will be to formulate an appropriate plan of work for conducting the Project.

Inadequate literature review (including poor referencing and citation)

The literature review is an important aspect of your Project, and the PPR should include a summary of the literature you have reviewed to date. The literature review serves to put your Project in the context of what other people are doing in the same area. By having a good knowledge of what other people have done, you are less likely to ‘reinvent the wheel’; you might avoid approaches that other people have tried and failed to make work, and find inspiration for how to do things better. A weakness of some of this year’s PPRs was the use of references to websites rather than academic sources such as journals or conference papers. The problem with websites is that they are not peer-reviewed, and the information they contain is not necessarily reliable. If you are using information obtained from websites, consider how reliable it is and whether to include some discussion about the reliability of your sources.

If your Project involves the development of a software tool or app, your literature review should also include some discussion of similar or related apps (if any) that are already available on the market, and how your proposed product will differ from these.

Some students submitted literature reviews that came across as general tutorials in the subject area of their Project. This is not what the examiners

are looking for. The review should cover material that specifically helps you to achieve the aims of your Project; for example, if your Project is in the area of neural networks for stock market prediction, the review should cover previous work on this topic (i.e. what has been done, what techniques were used, what data was used, what results were obtained, what were the strengths and weaknesses of these previous studies, etc.). It should **not** be a general tutorial on back propagation or other general neural network topics.

Another issue concerning references and citations is that some students included a Reference list at the end of their PPR, but did not indicate in the main text which references were relevant where. This is done by using a **citation**: a short marker in the main text, for example, '(Taylor, 2012)' which denotes an item in the Reference list. Even more importantly, some students copied sentences from other authors' work without the proper use of **quotation marks** and citations. It is perfectly acceptable to copy text from another source (within reason), but only if you clearly indicate using quotation marks and a citation where you have obtained the text. Failure to do this raises the suspicion of plagiarism – presenting someone else's work as your own – whether intentional or otherwise. There are severe consequences for plagiarism so you should know how to use quotations, citations and references appropriately.

Even with proper citation, care must be taken not to overuse quotations from other sources. The literature reviews of some PPRs mainly consisted of a list of quotes, with limited original text from the student. Such reviews are tedious to read, lack narrative structure or flow and are generally of very little specific value. A good literature review involves explaining the relevance to the Project of what has been done before, and how this will influence the way the current Project will be undertaken. The literature review can be useful to help justify the choices you make in your Project, including choice of research question, experimental design and analysis techniques. It therefore requires significant input and insight from you, not just a list of quotes from other authors. Often analysis of points where authors disagree may provide fruitful areas for discussion.

Poor Project plan (including lack of thought about testing and evaluation)

It is important that the Project plan is realistic and achievable within the time available. Some students presented Project plans that were far too ambitious. It is better to submit a smaller but complete Project, than a more ambitious but incomplete one. Plans for extending the work can always be discussed at the end of the Final project report if desired. Drawing up a **realistic Project plan is really important**. This year (2016–17) several students submitted Final project reports that started off very well, but included only very scant (or even completely missing) later chapters (e.g. Results, Discussion and Conclusions). In order to pass the Project, the final report must describe how **all** stages of the Project were conducted at a satisfactory level.

In contrast, a few students submitted very brief PPRs with very light Project plans. Remember that the Final project report is supposed to represent the culmination of at least 300 hours of intense, focused study. Furthermore, the less information there is in the PPR about exactly how you intend to do things, the less detailed feedback the examiners will be able to give you about your planned approach.

For Projects where software development is a major part of the work, the examiners will be looking for evidence that you have followed sound software development methods, including requirements gathering, design (use cases, wireframes, etc.), implementation, testing/bug fixing and evaluation. Students pursuing such Projects should think carefully about exactly what needs to be done for each stage, how it will be done and how long it will take. In your PPR you should discuss the current state of your development plans; a surprising number of PPRs for Projects involving software development do not even mention what language/s, tools or libraries the student intends to use to develop the system. Students will often be faced with the question of whether to use an off-the-shelf tool or library to achieve a certain goal, or whether to implement the system from scratch. Both approach is acceptable in principle and you should think about what would be most appropriate. You will need to balance the cost of learning to use an existing tool against the potential time saved once you have learned it, but on the other hand, you will almost certainly learn more about the details of a technique if you implement the code yourself. If you are faced with such a decision, you should document the alternatives and justify your final decision in your PPR and/or your Final project report.

In addition to identifying a specific question to be addressed, it is also very important to be clear right at the beginning of your Project exactly how you are going to **evaluate** the success of your work. A common failing in the PPRs was the lack of a clear plan for evaluation.

Think about what question/s you want to answer; then think carefully about the criteria for success by addressing the following more specific issues.

- How will you test the system?
- What results data will you collect?
- How will you analyse the results?
- How will you judge the significance of the results (e.g. what will you compare them against)?

For Projects that involve neural networks or other machine learning techniques, think carefully about what data the system will learn, whether you have access to a suitable source of data, whether the data is reliable (perhaps you need to check it for consistency first, or preprocess it in some way), and whether you will need to partition your data in some way (e.g. into training and test sets, and perhaps validation sets too).

For Projects which involve developing software for a group of intended users, you should include in your plan a process of stakeholder consultation at the start of the Project to establish their requirements and views on your proposed solutions. There are very few cases where such stakeholder consultation will not be appropriate. Even if such a consultation is **not** conducted, all software development Projects should include **some** kind of requirements gathering process, and this should be presented in your final report (and in the PPR if possible); that is, the design should be justified (e.g. from a theoretical perspective or from the literature review of previous work, if not from stakeholder consultation), and should not just appear with no explanation of how or why it was designed that way.

For software development Projects, in addition to stakeholder consultation at the design stage, it is also important to include some element of stakeholder evaluation after the system has been developed. For such

Projects, care should be taken at an early stage to decide who will evaluate the end product and how such an evaluation will be carried out. It may be that different sorts of evaluation are appropriate for different groups of stakeholders. Without seeking stakeholder evaluation and analysing the results, it can be hard to evaluate whether the Project has succeeded or failed in its goals.

For Projects that involve questionnaires and user feedback, many of the PPRs showed a lack of thought about exactly what would be required. Be sure to think about questions such as the following.

- Who will you ask? (Is there a single group of stakeholders, or multiple groups? How can you select the most representative sample possible from each group?)
- How many people do you need to include in order to generate reliable results? (Think about what is required for statistical significance, although sometimes practical matters may prevent you from including as many people as you would like.)
- What will you ask?
- How will you analyse the data?
- How long will all of this take?

A common failing with questionnaire design in Projects is that it is unclear how the answers to the questions will help the student in achieving their aims. Careful attention must be given to the design of questionnaires so that they genuinely contribute to the successful completion of the Project. It is all too common to see questionnaires which end up being a waste of time in terms of their contribution to achieving the Project's aims.

In data collection and analysis (as in all other aspects of the Project), the more detailed and specific you can be at an early stage about exactly what you are going to do, how you are going to do it and how long it will take, the higher the chance of you completing a successful Project on schedule.

Even if you have done a good job of drawing up a detailed Project plan at the start, and attempted to foresee potential problems that might arise and thought about how you might deal with them, unexpected delays are likely to occur. It is therefore essential to review your plan regularly and be prepared to adjust it if necessary. It is often useful to rate each feature you are thinking about including as: **must have/nice to have/could have**. You should concentrate on the most important features first, get those fully implemented and tested, and only implement other features if and when you have the time. Some students submit Project plans in their PPRs which do not match the reality of the progress they have described (e.g. the plan might show that system implementation should be well underway in mid-January, and yet it appears they have not started any implementation). Do not ignore slippages from the Project plan; if you are in this situation, you should talk about them in your PPR and present a realistic plan for how you intend to proceed.

For Projects involving the development of mobile or web apps, you should think carefully about what you will submit with your final report in May. The examiners need to see the code you have written, and ideally be able to run it. But they can only give you credit for work submitted before the deadline. Submitting a URL where the examiners can see your live site may seem useful, but the examiners cannot

verify when marking whether what is on the site was working before the submission deadline. Furthermore, the examiners will not have time to set up a web server and other supporting systems to run your submitted code on their own machines. A sensible approach is to include the following in your submission.

1. The source code you developed.
2. A full description in your Project report – with screenshots – of the final system.
3. A video run-through demonstrating all important aspects of the system.
4. (Optional) a URL to a live web server where the examiners can interact with your site, but only to experience what you have already fully documented in points 1–3.

Finally, some students did not allocate time in their Project plan for writing up their Final project report. This is obviously an essential part of the Project and will likely take longer than you expect. It is advisable to write sections of the final report as you go, rather than leaving writing all of it until a few weeks before the submission deadline.

In general, the 2016–17 PPRs spanned a very wide range of standards, from the weak to the truly outstanding. The preceding comments have highlighted some of the common problems. Further advice on how to produce a good PPR can be obtained in the following ways.

- Read the **CO3320 Project** subject guide.
- Look at examples of good Projects from previous years in the Project library section on the VLE (<https://computing.elearning.london.ac.uk/mod/data/view.php?id=6063>).
- Discuss problems and questions with fellow students on the Discussion forum of the CO3320 page on the VLE.