Examiners' Reports 2011–12
CO3320, Project
Final Report

We were pleased to see a marked improvement in the general standard of Project Final Reports this year, with an overall pass rate of around 84 per cent, compared to 79 per cent in the previous year. As noted in the Examiner's report for this year's Preliminary Project Reports, the number of students who chose to submit their PPRs online, and thereby benefit from feedback from staff at Goldsmiths, University of London, at an early stage of their project, was far greater this year than last year – this may have contributed to the reduced failure rate. However, ultimately good marks are down to the hard work put in by students.

As with every year, the standards of the submitted Project reports ranged from excellent to poor. Some students tackled very interesting academic questions, and the Examiners could see the potential for publishable material arising from a number of Projects. Projects which scored poorly were often ones which involved only the implementation of a piece of software or website, without attempting to answer any specific academic question.

Ideally, Examiners are looking for a Project to address a specific problem by following the structure of an academic research project: 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.

The *CO3320 Project Study Guide* contains lots of information about how the Project should be structured, and pointers to Further reading, and to what should appear in the Final Report. It is **essential** that all students read this guide **at the start of the academic year**. Although the vast majority of Projects were appropriate in terms of subject, amount and level of work, and quality of report, sadly in a small number of cases the student had completely underestimated the level and amount of effort required. In these cases it was unclear whether the student had even looked at the *Subject Guide*. The Examiners recognise that most students will have never attempted work of this nature before, and that some may have difficulties in gauging exactly what is required; this may be especially true for self-study students who do not have the benefit of a project supervisor. The *Subject Guide* is designed to help here (and it has been revised for 2012–13 to be clearer and more helpful in a number of places). In addition, there is a *Project Library* on the VLE, which contains examples of good Projects from previous years (this was started in 2010–11, and more Projects will be added each year). Students are strongly encouraged to use this resource to get a better feel for what a good Project looks like.

Projects which merely involve the implementation of a piece of software or website, with no academic question driving the development, will not achieve high marks. They may, however, be deemed sufficient for a pass if they demonstrate the application of solid software development practice. Even a Project which is, on the face of it, 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 of the system?). The more specific a question that 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.

For Projects which involve developing software for a group of intended users, be sure to include a process of stakeholder consultation at the start of the Project to establish their requirements and their views on your proposed solutions. There are very few cases where such stakeholder consultation will not be appropriate. Even if no such consultation is attempted, be sure to explain and justify how the requirements have been arrived at.

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 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.

In a number of cases, students proposed software architecture and new business processes to solve a problem or improve upon existing processes at a specific company. This kind of Information Systems Project is perfectly acceptable; however, in some cases the proposals for the amount of hardware and staffing required to implement and maintain the new system appeared to be very unrealistic (usually being much more than would likely be required in practice). Students undertaking such Projects should be sure to provide full explanations and justifications for any resources specified.

Of the Projects which **did** attempt to address a specific academic question, while many were done well, some were not done so well. A common problem with these types of Project is that the student apparently ran out of time; in these reports, the initial chapters on literature review, Project design, etc., would often be done well, but later chapters on analysis of results and discussion would be very superficial. This demonstrates the importance of good time management during the execution and writing up of the Project. It can be tempting to spend too much time on certain sections of the Project – often those sections which the student finds particularly enjoyable – while losing sight of the overall timetable and of the time available in which to complete all of the remaining tasks. It is vital to have a detailed Project timetable, and to regularly check your actual progress against it. If necessary, be prepared to change the timetable, and possibly cut down on some non-essential items, if you find that you are falling behind schedule. It is better to submit a smaller, complete Project report than a more ambitious, but incomplete piece of work.

One or two of the submitted Projects were essentially just literature reviews on a particular topic. A literature review is **not**, in itself, an acceptable Project. If no code is being developed, a Project based on a literature review should be enhanced by the student's own critical evaluation of previous work and ideas for moving the topic forward, future developments, etc.

One or two of the Projects discussed an **idea** for a new piece of software, technology or business process, but with no implementation, testing, or even surveying and feedback from potential stakeholders. The main problem here is that with no evaluation, there is little that can be said about whether the idea would actually work in practice, or whether unforeseen problems might arise. It is therefore extremely hard to gain a pass mark with this type of Project, and students are strongly encouraged to do some kind of implementation and testing in such cases.

A small number of students developed a piece of software but did not submit their code either electronically (for example, on a CD-ROM) or in hard-copy (as an Appendix to the Project report). Instead, they just showed screenshots of the software in the report. If a Project involves software development, it is essential that you submit the code (in either soft- or hard-copy), in order for the Examiners to evaluate it. If no code is submitted, a software Project is unlikely to pass. Even if the code is not stand-alone (for example, if it relies on other third-party libraries), the Examiners should at least be able to see the code even if they can't run it.

A common problem in the literature review section of the reports was to include long descriptions of literature of little or no direct relevance to the Project. The literature review should discuss **relevant** literature and carefully consider how it relates to the Project; it should not just be a summary of every single item you have read.

Many students made good use of surveys for various purposes, for example, to canvas opinions on a topic, or to obtain feedback on the usability of the developed software. This year, we saw a number of students make good use of online survey tools (such as surveymonkey.com or Google Docs Forms) to enable them to reach more people than may have been possible with paper-based surveys. However, in many cases the students failed to explain who the survey participants were or how they were chosen. Be sure to think carefully about participant selection, and to explain this in your report.

Many students made appropriate use of appendices for providing extra information, source code listings, more detailed analysis, etc. However, some students included information in the appendices which was not referred to in the main text, and therefore might easily not have been noticed by the Examiners! Be sure to mention at the appropriate point in the main text where extra information is provided in an Appendix.

The majority of the submitted reports were of an appropriate length. However, some were too short, missing out important details about what was done and how. Others were far too long, including irrelevant details (often in the literature review section, as described above). Projects are judged on the originality and quality of the work, not on the length of the report. Indeed, very long reports risk being marked down if the content is of little relevance to the Project topic.

Although most of the submitted reports were well structured, a small number were not. It is expected that the report will be divided into a number of chapters, and that each chapter will likely be divided into a number of sections. Chapters and sections should be numbered for easy reference, as should figures, tables, etc. Although no specific numbering system is prescribed, it is a good idea to number figures in parallel with chapter numbers (for example, such that the third figure in Chapter 2 would be labelled Figure 2.3). A small number of submitted reports had strange numbering systems, or no numbering at all, which makes reading the report and finding figures, etc., difficult.

The style of writing in the reports was generally good or at least acceptable, with a few exceptions. In the worst cases, the quality of writing was so bad that it became impossible to understand the meaning of some sections of the report.

Most of the stylistic problems mentioned in the previous few paragraphs are very easy to avoid. It is always a good idea to make the report as easy for the Examiners to read as possible, so that they do not have to struggle to understand exactly what you have done, how and why.