Undergraduate Computer Science Projects in UK: What is the point?

Vassiliki Bouki

University of Westminster, Watford Rd, Northwick Park, Harrow, HA1 3TP, London, UK, boukiv@wim.ac.uk

The final year undergraduate project is a crucial component of a Computer Science (CS) degree in UK. The learning outcomes of the project include critical thinking, creativity, time management and communication skills. Although it is apparent that the project plays a major role in the curriculum of a CS degree, many students (and some supervisors) fail to appreciate its significance. In addition, the last three years we experience a downfall in the quality of projects in some UK Universities. The reason behind it might be related with common misunderstandings about the nature of a CS degree. These are the same misunderstandings that several researchers consider responsible for the drop down in CS student numbers. The aim of this paper is to present the expected learning outcomes of the project, to examine how it differs from the other modules and to discuss concepts such as research, supervision and communication skills in relation to CS projects. Bringing research as early possible into projects could be beneficial for UG students; 'distant supervision' is a practice that could be useful and practical for overloaded lecturers and working students; communication skills should be exercised during the project.

Keywords

Undergraduate Projects, CS Curricula, Real-time Distant Supervision, Research in Undergraduate level

1. Introduction

The final year undergraduate project is a crucial component of a Computer Science (CS) degree in UK. It usually counts as a double module in the final year and it is the one that makes the difference between an ‘Ordinary’ and an ‘Honours’ degree. The project is the epitome not only of the knowledge students acquire during their studies but also of what they are able to do as ‘computer scientists’. This goes further than the mere knowledge of ‘programming’. It includes critical thinking, creativity, time management and communication skills. Although it is apparent that the project plays a major role in the curriculum of a CS degree, many students (and some supervisors) fail to appreciate its significance. Some repeatedly ask ‘why bother with project?’ [1]. In addition, the last three years we experience a downfall in the quality of projects in some UK Universities. The final year project seems not to be received well by the students who struggle to achieve its learning objectives. The reason behind it might be related with common misunderstandings about the nature of a CS degree. Many students tend to equal computer science with programming and this makes some of the learning objectives of the project (e.g. the writing of a long academic report) incomprehensive. These are the same misunderstandings that are considered responsible for the drop down in CS student numbers by several researchers [2, 3].

In this paper we will present the expected learning outcomes of the project. Then we will examine how it differs from the other modules in a CS degree. Our aim is to show how far these outcomes and the nature of the project are from the common way of thinking that defines CS as programming. Finally, we will discuss how concepts such as research, supervision and communication skills are related to projects.

2. What is in a Project?

Recently, it has become apparent that many people, even computer science students or students who want to pursue a degree and a career in computer science, have a wrong impression of what 'computer science' is. Most tend to believe that 'Computer Science = Programming' [2]. In a survey among 836 high school students in 9 different schools in USA the most common respond to the question 'what computer science students learn' was 'programming' when other responses included 'computer repairs', and 'how computers work'. [3]. Some argue that this misunderstanding is mainly responsible for the sharp drop down in the number of students who select computing for their studies. In order to bring students back in computer science, we should convince them that there is more on computer science that just programming. It is a 'whole science'. The final year project is a crucial component of the curriculum that proves that a CS degree is about a science and not about a skill. On the other hand, if students misunderstand the nature of a CS degree, the project module is the one that will suffer more than any other module.

2.1 CS Project Taxonomies and Expected Outcomes

Researchers have proposed taxonomies of CS projects based on several criteria. [4, 5] Ph. Weaver classifies CS projects into two main categories: development projects and research oriented projects. Very rarely, these two categories are exclusive. In most cases, development projects include a research proportion and research projects include an element of development. The main difference is the focus of the project. Development projects focus on methods, techniques and tools that have been learnt during the studies when the research proportion is minimum.

This category includes:

- Database system projects; they emphasise on the construction of a database and include data handling and interface programs.
- Large business system projects; they aim in the construction of a system for the needs of a large organisation. Initial feasibility analysis, data gathering and analysis, development of a prototype are required in this type of projects.
- Multimedia projects; they involve the development of programs that interact with the user through different media, such as graphics/pictures, sound and video.
- Complex programming projects; they focus on the design and the implementation of complex algorithms.
- Technical implementation; they are concerned with the delivery of significant infrastructure components (e.g. complex hardware devices, operating systems or database management systems).

On the other hand, research-oriented projects are systematic, in-depth investigations on a specific topic.

Some examples of projects in this category are:

- Methodology investigation projects; they investigate and evaluate different methodologies.
- Technology investigation projects; they investigate and evaluate the effectiveness of different hardware or software to a given problem.
- Case study; they investigate and evaluate real-life situations. [4]

James, Hawick and James, create a different classification, based on the topic. They identify four main categories of CS projects: building and using software; building hardware; theoretically-based projects and re-writing projects. [5] The first two categories correspond to the development projects as defined by Weaver and the latest two to research projects – although they are not identical. There is a significant overlapping among there categories of projects.

It is obvious that none of the existing taxonomies can be considered exhaustive. In addition, as CS expands and is related to other sciences, the list of possible project types is expanded as well. As a result it becomes a 'non-ending' task to try to define what a CS project is using as criterion the different categories of projects. A more effective way to answer the question might be to define what kind of knowledge and skills we want students to acquire through a CS project. Whatever the type of the project is, it seems that there is an agreement among educators about its learning outcomes.

A student who successfully completes the project is expected to be able to:

- Do comprehensive literature review of issues related to a selected area of CS and critical evaluation of the findings.
- Do effective time management in preparing a plan and following it.
- Write a well-structured and coherent report about the work.
- Justify the methods and processes used in carrying out the project
- Use creative thinking in approaching the project topic.

Apparently, a CS project goes further than designing and writing a computer program. The project aims to teach and give students the chance to exercise research skills; time-management skills; critical thinking and writing skills.

2.2 How does the project differ from Other Modules?

It is true that the final year project is different from what students experience during the first two years of their studies in a UK university. Three main differences make the project unlike the other modules.

First, the project is a non-taught module. The student has to organise it and decide any 'personal', intermediate deadlines. It is based on regular meetings with the supervisor. Most of the meetings (except maybe the initial ones) are expected to be 'student-drive'. This is a big difference to the typical lectures-and-tutorials modules where it is the lecturer who organises the module and designs the activities. As a result the regular meetings with the supervisor play a crucial role for students' progress and success. Being a non-taught, supervised-only module the project puts the student in the centre. The question that arises here is if students are prepared for this. The decline in projects' quality could indicate that students are not really ready. From our experience we could identify the cause of the problem in the educational changes during the nineties. Since the beginning of the nineties, UK universities have experienced a significant increase in students' numbers. This is one reason that makes today's drop in students' numbers looking so bad. The flow of more and more students in the nineties had as a result the looseness of the traditional supervision / tutoring system. In the past the relation between the student and the supervisor was very close and important throughout students' studies – not just in the final year. Good, quality, thorough supervision was the back bone of the British educational system. Today, students enter the final year of their studies without significant experience of one-to-one collaboration with personal tutors or supervisors. Students might have met their personal tutors very few times before they start the third year and most of those meetings are about personal problems than a fruitful academic discussion on an academic topic. The reasons are twofold. On one hand the increase on students' numbers made the supervisors overloaded. As a result, the available time for each student became less and less. On the other hand, and especially with the introduction of fees in UK Universities, most students have part-time or even full-time jobs. It is a common remark among staff at the University of Wesminster, that very few students are 'full-timers' in the traditional way anymore. Most struggle between their studies and their other commitments. They usually try to meet the most 'immediate' requirements of their studies: attendance and deadlines. As a result meetings and collaboration with the personal tutor in the first two year and the project supervisor in the third year, become the least of their priorities.

The second difference is that the project is an independent piece of work. The supervisor provides guidance but the direction of the project is at the discretion of the student. Even the selection of the project – depending on the selection mechanism each University applies – is up to the student. If there is a disagreement between the student and the supervisor, in most cases, students are free to follow the direction they want, taking full responsibility for the final output. And here an odd thing happens. Most students are reluctant to show initiative in the selection and the progress of their projects. They mostly have the attitude of a taught module ('tell me what I have to do / read / learn') rather than adopting an independent, responsible approach. Under the light of the previous remarks this is easily explained. Students are not trained to work independently. They also have jobs where they are allocated tasks and they expect the same to happen in the University. Others, who adopt an independent approach, do it mainly because of lack of time and they understand it as 'I disappear during the whole year, I do not respond to meetings with my supervisor and I submit at the end of the year the final report with no guidance'. In many cases like that, students have a difficulty to accept critique on their work or comments from the markers if they are not in favour of their work. In very few cases, students accept responsibility and are mature enough to make a discussion on their work and accept constructive comments.

A third difference is that for the completion of the project, the writing of a big project report is required. The project report – usually in the excess of 10,000 words – is a crucial component. It is through the report that students communicate their ideas and findings with the markers. In most other modules students do not have to write long academic reports. It is a big discussion what the role of the report should be and if, at the end of the day, a report is required. In the heart of the argument lies the answer to the question 'what do we expect computer scientist to be'. The argument against the report is that what we really expect from future computer scientists is to have good programming skills and be highly trained and specialised in fields related to computing (software and hardware). The use of language plays a secondary part. Those in favour of the report, argue that computer scientists are still *scientists* and as such they must be able to express their ideas in a clear, coherent way. From our experience, the last few years the quality of the project report drops rapidly. Students are less capable today to write a long, clear, logical report than twenty years ago. The explanation for than might go further than the design of any curriculum. It might also be a crucial component, extremely important for the design of the curriculum and the directions an educational system wants to take. In the technological era, students become less and less used to read and write in a linear way – the one required for an academic report. The Internet and, more specific, the non-linear structure of the Web seems to be responsible for this change. Reading in a non-linear environment involves different cognitive processes than traditional reading [6, 7]. It is not just the lack of experience in writing reports that is to be blamed. It is the limited experience to read and write *in a linear manner* that seems to be a serious barrier for the completion of a good project report. The non-linear approach in reading and writing and the different cognitive processes that are involved, is a new, different way of perceiving knowledge. So, although our students have much more experience (and chances) to read and write non-liner text, we still ask them to be able to express their thoughts in a linear manner. The new data about the liner and non-linear way of perceiving information, provided mainly by the disciplines of Cognitive Psychology and Cognitive Science should be taken seriously into consideration for the design of any academic course.

3. How to Become Computer Scientist through the Project

3.1 Bringing Research into Projects

As we discussed earlier some taxonomies define different categories of computer science projects in relation to the topic they deal with. This topic should be related to a selected area of Computer Science and students have to review the current bibliography on it. The relation between project and research starts long before students start the literature review. The crucial time to bring research into projects is before it starts, during the project selection phase.

In UK, different Universities follow different approaches. In some Universities students are encouraged to find their own topics; in others they are asked to select from lists provided by project supervisors. Project lists operate on a 'first come – first served' basis. Apparently, there are advantages and disadvantages in each approach. Some researchers believe that project lists are the last resort for students who have failed to identify a project. [4]. Our experience is different. It is mainly students who have a genius interest for their projects who prefer to go for the project lists. Students with less interest tend to choose an 'easy' and 'popular' topic. Another argument against project lists is that students may end up with a topic that does not match their interests and as a result it is possible that they'll soon loose their interest [4]. This is definitely a possibility – especially if student's interests are totally strange to the project area and the topic has been imposed to him/her. On the other hand, if we consider it from supervisor's point of view, we could argue that it is more likely that supervisors will offer better and more thoughtful guidance if the topic they are asked to supervise is between their interests and they have research in this field. There are also some obvious advantages when students propose their project topics. The first one is 'creativity'. Students are asked to do some thinking and reflection on their studies, knowledge, abilities and interests. What they select should match their interests and their abilities. This approach agrees with the 'travelling method' [8] in teaching – the project is a journey and the student decides where to go. The supervisor is there just to offer some help – if needed – but not directions. In reality, things are not so smoothly. In many cases students do not really do any reflection. They simply follow the flow. Every year we see some topics that are particularly popular. Students seem to propose them because they believe they are 'easier' and not on the basis of what they want to achieve. In the year of 2006-07, 66 out of 88 computer science students in our University chose to do 'a web site with a database' project. From discussions with external examiners, it seems that this is the most popular topic in other UK Universities as well. Unfortunately, it is popular, because students consider it as easy and straightforward – not because they enjoy the topic. The main disadvantage of choosing a project with these criteria, is that students feel lost when they start the project. They do not really know how to start and from where. Another disadvantage is that it is not rare that supervisors end up supervising a project that is not among their interests and not in the area of their specialization. Apparently both approaches have advantages and disadvantages. At the University of Westminster, we follow a mixed approach. Students have the freedom – they are actually encouraged – to bring their own ideas and discuss them with potential supervisors. A member of staff should agree to supervise the project and give the 'go ahead'. If no member of staff has agreed on the topic or the student has no specific ideas, then they have to choose an idea proposed by a member of staff.

The actual question during the project selection phase is how to match the interests of the staff with the interests of the students. Soon we realised that our students were not well informed about the work members of staff do, their expertise and research interests. If staff informs students about the research they do, students show more interest. An effect beneficial for supervisors is that this approach helps 'continuation'. There are some constant trends in the work members of staff supervise, upon which supervisors can built – topics do not seem to be ‘all over the place’. The main challenge is to make visible the research supervisors do to undergraduate students. At the University of Wesminster, we have created an ‘Interactive Centre’ for projects. Each member of staff has his/her password and they can upload and manage online specific project topics, papers they publish, research interests, teaching responsibilities or any other information they believe it is important students to know in order to do a project with them. It is not just about a list or projects – it is a more coherent presentation of staff’s research interests. The aim is not to push topics to students but to make the staff ‘visible’.

3.2 Supervisor and Supervisee

The relation between the student and the supervisor seems to be a key factor for the success in the project. The role of the supervisor becomes so important that some researchers consider it as more important than the role of the personal tutor [5]. James considers supervision as an ‘important way of tutoring’ and he points that it is feasible even in time of resources pressure when tutoring might not be feasible [5]. Apparently during the regular meetings with the supervisor, both parts get to know each other much better than in the less often meetings with the personal tutor. Frequent meetings with the supervisor are critical for the project success. Meetings can be formal or informal; one-to-one or one-to-a-group; in-person or via e-mail.

For overloaded supervisors as well as working students, distant supervision is possible using tools and principles from distant learning. Most educational software used in UK Universities (such as Blackboard, WebCT, Top Class etc) offer tools for ‘real-time’ communication. The most obvious advantage of using these tools instead of communication through e-mails is that students are able to ask questions and get answers in real time. No long ‘waiting periods’ – that might be from a couple of hours to days – occur and as a result there is no interruption in the learning process. A meeting organised using real-time tools, could have exactly the same learning and pedagogical potential as a face-to-face meeting. Both parts enjoy ‘place independence’ as they can be anywhere they like – should they have internet access. On the other hand the only way to have real-time communication is to impose some time constraints to both parts – supervisor and student should have agreed in advance to login at a certain time.

In a pilot study we run at the University of Wesminster from March to May 2004, a supervisor offered distant, real-time project supervision to a group of ten, 3rdyear project students, registered for the degree ‘BSc in Computer Science’. The ‘Lightweight Chat’ tool, offered by Blackboard educational software, was used. Blackboard offers two tools for real-time communication: ‘Virtual Classroom’ and ‘Lightweight Chat’. The main difference among them is that ‘Virtual Classroom’ allows the lecturer and the students to access material already uploaded on Blackboard and present it in the whiteboard, when the ‘Lightweight Chat’ is more a ‘chat room’. All students who took part in the pilot were familiar with Blackboard in accessing learning material but they had no experience of Blackboard’s ‘Lightweight Chat’ or ‘Virtual Classroom’ tools. They were offered three one-hour long slots each week. Students and their supervisor were supposed to use these slots for discussions about the project. There was no specific agenda. Students could discuss any issue about the project and ask any question. Usually the communication started with a prompt from the supervisor such as: ‘Hi, thanks for joining this session. Please tell me about your progress in the project this week. Are there any problems?’ [9]

The results of the pilot study showed that real-time communication is a great tool for project supervision. It has the potential of ‘face-to-face’ meetings, it gives ‘place independence’ – as all the distant communication tools do – and it does not involve most of the disadvantages of the real-time communication tools. It has been confirmed in several studies that the main problem with the real-time communication tools in teaching is the size of the group [10, 11].

Apparently, if too many people are logged in at the same time, the time required for communication increases and the quality of the communication suffers. A researcher writes 'it seems that if more than 15 people are actively chatting at one time quality suffers.' [11]. In project supervision the group size is not a problem as in most cases two persons only participate in each session: the supervisor and the student. Even if supervisors want, from time to time, to organise real-time meetings with all their project students, the number of student involved is seldom bigger than 15. The other disadvantage with real time tools is that delays usually occur in the communication, due to technical as well as 'human' limitations. No matter how much experience participants have with such tools, they need some time (from seconds to minutes) to read what the other part writes and to reply. This gives the feeling that the communication is not 'instantaneous' [9]. As it concerns the technical limitations, a quick connection is required for any kind of real-time communication. About the 'human limitations', they cannot be avoided but they are improved because of the small number of participants. In addition, it was easy for the supervisor to organize meetings where all the ten supervisee participated. This proved to be a good technique, because it gives to students a sense of community and belonging.

Summarising, the collaboration between the supervisor and the project students is crucial for the success in the project. 'Distant supervision' using 'real-time' communication tools can be the answer for the overloaded staff or working students. Based on a pilot study we carried out at the University of Wesminster, using Blackboard educational software it seems that distant project supervision has the same educational potential as face-to-face supervision.

3.3 Projects and Communication Skills

Communication skills, such as presentation of ideas to more persons than the supervisor and intermediate (oral or written) reports, are good to be encouraged during the project. At the University of Wesminster, students have to submit a progress report – approximately in the middle of the academic year – and present it in the form of a poster. The need of an intermediate formal report comes from real-world experience. In the real world, people responsible for any sizeable project are expected to report on the progress of the project at regular intervals in order to convince their superiors that work is progressing according to the schedule, and – most importantly – to identify any serious problems before they become unmanageable. The poster event is an open, 'public' event, where all members of staff are invited. Students have the chance to explain their ideas to a wide audience and to receive feedback from people other than their supervisor. Students seem to enjoy this kind of event and we consider a good practice if we make students 'go public' even earlier, at the stage of project selection. Having students present their ideas in a wider audience at the beginning of their projects, will make them to think more seriously about their project and try to form more clear ideas earlier. Finally, after the submission of the project report, students face an oral examination. This is actually their chance to stand up for their work, present, explain and defend it. Apart from a good exercise in their communication skills, it is also a useful aid for the markers (supervisor and moderator) to form a more accurate opinion about students' knowledge and understanding.

4. Conclusions

Final year project is a crucial component for most CS degrees in UK because of its learning objectives. The project differs substantially from the other modules. It is a non-taught module and students have to work independently. Some of the objectives might seem incomprehensive or irrelevant to students who even after two years of studies still believe that the only important part in a CS degree is programming. Getting together staff's research and students' interests is crucial. Distant supervision is possible using real-time communication tools and it is beneficial for overloaded staff and working students. Finally, the project should encourage several communication skills, such as writing of an academic report, time management with intermediate deadlines, presentations, the ability to express ideas in a clear comprehensive way written or orally. Combining all these learning objectives, the final year project is the ‘proof’ that ‘computing’ is a science and not just a skill.

References

1 Clark M.S.C. Boyle R.D. A Personal Theory of Teaching Computing Through Final Year Projects. Computer Science Education, 1999, 9:3, 200-214

2 Denning P., McGettrick A. Recentering computer science. Communications of the ACM, volume 48, issue 11, November 2005, pages 15-19

3 Carter L. Why Students with an Apparent Aptitude for Computer Science Don’t Choose to Major in Computer Science. SIGCSE ’06, March 2006, pages 27-30

4 Weaver Ph. Success in Your Project. Pearson Education, 2004

5 James H.A. Hawick K.A. James C.J. Teaching students how to be Computer Scientists through student projects. ACM International Conference Proceeding Series; Vol. 106, pages 259-267, 2005

6 Catledge L. D. James E. P. Browsing Strategies in the World-Wide Web. Georgia Institute of Technology. 1995

7 Protopsaltis A. Bouki V. Reading Strategies in Hypertexts and Factors Influencing Link Selection. In Kommers P. Richards G. (Eds), ED-MEDIA, June 2006

8 Clark M.A.C. and Boyle R. D. A Personal Theory Teaching Computing Through the Final Year Projects. Computer Science Education 1999, Vol. 9, No. 3, pp 200-214

9 Bouki V. Real-Time Communication in e-learning: A Pilot Study Using Blackboard. In WSEAS Transactions on Information Science and Applications, Issue 6, Volume 1, December 2004, pp 1726-1730, ISSN 1790-0832

10 Bromage Adrian, 2002 ‘Atatvistic avatars: Can synchronous learning communities transfer to virtual worlds?’ A paper for the ‘Virtual Learning and Higher Education’ conference,10th-11th September 2002, Mansfield College, Oxford. http://www.inter-disciplinary.net/vlhe1s3.htm (last accessed 15 Sep. 07).

11 Suguri Vera et al, 2002 ‘Pedagogical uses of web-based chat’. In TechKnowLogia, issue January - March 2002, pp38-43.