Learning software engineering at a distance

Learning Software Engineering at a Distance

Brendan Quinn, Leonor Barroca, Bashar Nuseibeh, Juan Fernández-Ramil, Lucia Rapanotti, Pete Thomas, and Michel Wermelinger, *The Open University*

The Open University's SE curriculum and delivery mechanisms are shaped by its commitment to offering professionally accredited part-time, open, and large-scale distance learning primarily aimed at IT practitioners.

Delivering a software engineering curriculum by distance learning requires innovative and flexible approaches to presenting and managing the learning materials. At the Open University, we've been offering a broadly based master's degree in Computing for Commerce and Industry by distance learning for over 20 years.

Recently, the OU's Computing Department (www.computing.open.ac.uk) has developed more specialized master's programs in software development and in the management of software projects. We characterize these professionally accredited programs as part-time, open, and large-scale distance learning primarily aimed at IT practitioners. In this article, we show how these characteristics have influenced our software engineering curriculum, drawing distinctions from more conventional programs. Although we discuss our particular experience at the OU, we note broader lessons learned that might benefit other institutions engaged in designing and delivering software engineering curricula to a geographically distributed student body.

Background

Many people working in the IT industry fit the following profile: They have a university education, possibly including some study of computing or software development. They're well established in responsible jobs as developers, managers, database administrators, and so on. They possess knowledge about their own specialist areas and attend occasional training courses on new technological developments. After some years in a role like this, they want to broaden their understanding of computing or software engineering and to have the opportunity to study one or more topics in some depth. They also feel that additional formal qualifications would help their career development.

However, many such professionals can't afford to take time off from their jobs to study full-time. They can study part-time at many institutions, but not everyone lives near a suitable institution. With busy jobs and often considerable traveling, it can be hard to fit in even the limited attendance required. Many such people find a solution to their dilemma in a distance learning master's program, such as that offered by the OU.

The OU, based in the UK, has been a large-scale provider of higher education by distance learning since its foundation in 1970. The OU has become known worldwide for its many pioneering initiatives in opening up higher education, its quality course materials, and its teaching. Current enrollment is over 200,000 students in a range of programs from foundation (college preparatory) level to master's and PhD.

0740-7459/06/$20.00 © 2006 IEEE

Curriculum structure

The postgraduate computing program consists of a number of courses, each of which has a unique identifying code (such as M865 for Project Management) and a points rating, indicating the amount of study and the weighting of its contribution to a program. (A *course* in the OU corresponds to a course in a US program of study—that is, a unit of study making up part of a degree program; most UK universities use the term *module* instead.) With one exception, courses in the computing program are rated at 15 points, and each course is expected to require about 150 hours of part-time study over a six-month period. Students need eight such courses for the 120 points required for a postgraduate diploma. To complete the master's program, they must also pass a 60-point research and dissertation project, which takes just over a year part-time. The total time to achieve a postgraduate diploma can take from two years upwards, with some programs having a maximum permitted duration of four years to diploma.

The OU offers four possible routes, each with a different title for the postgraduate diploma or master's. For this article, we focus on the master's in software development (see http://pgcomp.open.ac.uk). Table 1 shows the program's structure.

Table 1

Program structure for the master's in software development

Course code	Course title
Core for postgraduate diploma—90 points	
M882	Managing the Software Enterprise
M883	Software Requirements for Business Systems
M885	Analysis and Design of Enterprise Systems
M873	User Interface Design and Evaluation
M874	Software Development for Networked Applications Using Java
M876	Relational Database Systems
Optional courses for postgraduate diploma—select 30 points (two of the following)	
M865	Project Management
M877	Advanced Database Technology
M879	Distributed Applications and e-Commerce
M881	Architecture and Networks
M886	Information Security Management
Compulsory for master's degree—60 points	
M801	Research Project and Dissertation

The student experience

We call the OU's approach to distance education *supported open learning*, which has been refined over many years. Students in our master's programs study using various media, including specially produced printed material, standard textbooks, online material, CDs, and DVDs. We also make all OU-produced printed material available online in the form of e-books, so students can easily store or search them on a computer or access them from anywhere with Internet access. Students each have an allocated *tutor*, who marks assignments and provides academic support for groups of up to 18. Tutors are university academics or suitably qualified IT professionals employed by the OU on a part-time basis. Students communicate with each other and with their tutors mainly via email and electronic conferencing.

We assess a student's progress through coursework assignments—normally two or three in each course—and a final examination. Typically, the exam and the coursework assignments are equally weighted at 50 percent of the course's overall marks. The final research project is different because we primarily use the dissertation to assess it. Students submit all coursework electronically but take their examinations in person at examination centers located throughout the UK and in other countries.

Each course has *learning outcomes* that define the knowledge and understanding, cognitive skills, and practical and professional skills that students are expected to attain. We use a specified range of learning outcomes in designing each assignment and exam and make sure that students know how the learning outcomes are assessed. For example, in an assignment for course M882 Managing the Software Enterprise, we asked students to demonstrate their knowledge and understanding of these learning outcomes:

- the management of human, technical, financial, timescale, and organizational resources;
- quality management for software development, software evolution, and process improvement approaches, and
- the risks involved in developing and evolving software and how to identify, assess, and manage them.

How we develop the curriculum

A team of OU full-time academic staff, in

Figure 1. Selected features of courses from the master's in software development.

M882 Managing the Software Enterprise
This wide-ranging course includes topics often neglected in traditional software engineering programs such as ethical issues, intellectual property rights, human motivation, risk assessment, and software evolution, many of which are also required for professional accreditation. It's aimed at people in or working toward a management role involving software. The course examines software's role in organizations from human, social, knowledge, business, and software engineering perspectives. Students can explore software development processes and concepts by means of system dynamics simulations.

M883 Software Requirements for Business Systems
This course highlights the importance of requirements in software engineering. Web and DVD resources keep students abreast of current research and practice, including videos of seminars by experts in the field. Students can also use a specially designed software application for requirements capture and as a case study of requirements elicitation (by examining and extending the tool's development).

M885 Analysis and Design of Enterprise Systems—An OO Approach
This course iteratively and incrementally teaches object concepts and techniques, paralleling the iterative software development processes being introduced. The course particularly aims to develop students' reflective skills about making the most appropriate modeling choices. For example, current topics include alternative approaches to modeling such as agile modeling, Extreme Programming, and model-driven architecture. It also reinforces a rigorous approach to UML modeling with consistent annotations using both natural language and the Object Constraint Language.

M886 Information Security Management
This course takes a security management focus, focusing on skills appropriate for the master's level. Based on UK and international standards (BS 7799/ISO 17799, see http://emea.bsi-global.com/InformationSecurity/Overview/index.xalter), it lets students apply the course content to an organization they know. Completing the course gives students the knowledge and skills required to develop and plan the implementation of an organization's information security management policy.

M801 Research Project and Dissertation
This course lets students research, in detail, a topic of professional relevance to them or their organization through a 13-month period of supervised work, leading to a dissertation (10,000 to 15,000 words) and poster presentation. Students and tutors use specially designed online systems to manage the process as students submit a series of gradually refined proposals and drafts leading to the final dissertation.

some cases with help from external specialist consultants, writes each course's academic content (see figure 1 for some course descriptions). Because most courses provide substantial, specially designed printed material, the course production process has much in common with conventional book publishing. The academic writers are supported by administrators, editors, graphic designers, and the whole apparatus needed to print course books or make them available in PDF format.

Because of this production process, we place great emphasis on the materials' initial quality—correcting significant errors or improving the teaching style later can be more difficult and costly than in a conventional university. We put all draft course material through detailed scrutiny by critical readers from academic and industrial backgrounds. Additionally, we appoint one or more independent external assessors, from another university or from industry, to review the course materials as a whole.

A program committee, composed of OU academic staff and external advisors, oversees the program's curriculum, again providing independent expertise from both academic and industrial perspectives. This committee aims to ensure the program's coherence and provides strategic vision to keep it current and relevant in the longer term.

To avoid offering out-of-date course materials, we provide more topical or ephemeral material online in forms that we can easily update, and we regularly renew the case studies and scenarios used in assessments. We also annually review courses for minor changes and have a major review normally every few years, according to a preset schedule. After a major review, we might update, delete, or replace the course.

The OU also investigates new approaches to course production, sometimes inspired by the software development process. For example, when producing course M885, the academic

Table 2

Student numbers for 2005

Course (points)	No. of students	Pass (%)	Fail (%)
M801 Research Project and Dissertation (60)	88	73.9	25.0
M865 Project Management (15)	579	80.3	19.7
M873 User Interface Design and Evaluation (15)	218	85.5	14.5
M874 Software Development for Networked Apps (15)	230	93.9	6.1
M876 Relational Database Systems (15)	264	84.5	15.1
M877 Advanced Database Technology (15)	141	84.4	15.6
M885 Analysis and Design of Enterprise Systems (15)	215	89.6	10.4
M879 Distributed Applications and e-Commerce (15)	157	93.0	7.0
M880 Software Engineering (30)	261	90.0	10.0
M881 Architectures of Computing Systems (15)	94	84.0	16.0
M886 Information Security Management (15)	143	93.0	7.0
Total	2,360	86.6	13.3

team adopted various agile development practices, including short, time-boxed, and incremental production cycles (material was produced and revised on a weekly basis) and pair authoring.[1]

Program characteristics

We have characterized the OU master's program in software development as part-time, open, and large-scale distance learning, professionally accredited at the master's level and mainly aimed at IT practitioners. In the following sections, we consider each of these characteristics and discuss their influence on the curriculum's design and delivery.

Part-time

Students have considerable choice about the timing of their study and the associated workload. They can take one course at a time or several at once. They can vary the order of study (relatively few interdependencies exist among the courses), although we also recommend pathways, and they can take breaks from study. This lets them continue with their careers and helps them fit in the demands of study with competing pressures from work or home life.

Some students stop after one or several courses or after obtaining the postgraduate diploma (154 students completed the diploma in software development in 2005)—only a minority continue to the master's dissertation. We consider this early exit as normal and in no way a failure, as is sometimes the case in full-time programs. Students undertake the courses for various reasons, and this is reflected in their exit points. In recognition of this, we've recently added a postgraduate certificate award that will be available to students after completing 60 points.

Because some students only take one or several courses, we need to make each individual course attractive enough to students so that it's economically viable to produce. We normally withdraw or substantially rewrite courses with persistently low student populations (in our case, about 100 students or fewer). See table 2 for a snapshot of student numbers. In 2006, we offer two new courses, M882 Managing the Software Enterprise and M883 Software Requirements for Business Systems; they replace the 30-point M880 Software Engineering course.

Open

The OU's founding mission was to open up higher education to people who previously might have been denied access for lack of prior formal qualifications. We keep entry to our master's program in software development open in that we have no compulsory entrance requirements. In practice, we give strong advice to potential students about the appropriate background and study pathways—we normally expect a university degree or diploma and at least several years' experience in IT. Most students in the program have this background—some have much more.

For the minority without a strong back-

The program's part-time nature lets students spread their study over time to gradually acquire higher-level skills.

ground, there is an element of self-selection. They or their employers pay significant fees, and if they're not sufficiently motivated to catch up, they won't continue with the program. The program's part-time nature lets students spread their study over time if necessary to gradually acquire the necessary higher-level skills—the OU also offers numerous study skills resources. Actual completion rates on courses are normally 80 percent or higher—many of those who drop out do so for nonacademic reasons, such as work or home pressure, and might return later to resume their studies.

Unlike many conventional universities, we rely on output standards rather than on selection by input standards. We guarantee these output standards through independent external examiners and professional and academic accreditation.

Large scale

An OU course can have many hundreds of students enrolled in it at any one time. This has advantages—it provides the financial resources to fund high-quality materials. The many students and their tutors bring a wealth of experience in IT and other fields, which they can share to some extent, although this is currently perhaps an underused resource.

The large scale (together with the distance learning) could make some students feel rather isolated and anonymous. We address this by allocating students in small groups to tutors, by online conferencing, and by a limited amount of optional face-to-face teaching.

A key issue is ensuring consistency in the student experience, in particular in standards of marking and feedback on assignments. We monitor a sample of marked assignments for every tutor and feed the outcome back to the tutor. This takes place electronically—from student submission through assignment monitoring and tutor feedback. We also perform statistical monitoring to identify any potential cases of excessively lenient or harsh marking. Again, we inform tutors so that they can adjust their marking if necessary. We use the same sort of statistical approaches in examination marking to prevent or rectify any potential inconsistencies between markers.

Distance learning

One of the major attractions of distance learning is the flexibility in the place and time of study for students. Many students use time spent traveling or waiting at airports to study the course materials, which are specially designed for self-study (for example, the courses have no lectures and few tutorials). Some find the printed materials most convenient for this—others access our online materials.

Because students don't normally have close contact with academic staff or fellow students, the course materials must have particular characteristics. The materials must be fairly self-contained. They must also facilitate active learning by means of activities, self-assessment questions, simulations, and so on.

Some issues apply particularly to using distance learning for software engineering. We've observed two main aspects to this—one technical, relating to the potential diversity of student hardware and software platforms—the other being the need for software engineering to be a social activity as well as a technical discipline.

Students must provide their own personal computers and arrange for Internet access. It can be challenging to find software engineering tools that are compatible with a wide range of platforms, have stable versions, and are affordable enough to distribute as course material. Sometimes we distribute a standard software tool but permit students to use different tools. At other times, we develop tools especially for a course.

Software engineering is rarely an individual activity—teamwork is the norm. Many of our students, being practitioners, will have teamwork experience in their employment. However, it can be difficult to provide opportunities for teamwork in a distance-learning environment. At the moment, the master's programs don't have a provision for teamwork—all assessment is individual. Students can communicate using the OU online-conferencing systems, but this isn't normally used for working in teams. We've recently had encouraging experiences using the online-conferencing system for teamwork in undergraduate courses. The OU has also adopted Moodle (http://moodle.org) as its virtual learning environment to provide an integrated, high-quality online learning experience for students. We're working to adapt, extend, and integrate Moodle with existing systems (including online conferencing). Soon, this will let us make full and integrated use of newer technologies, such as wikis and blogs.

Similarly, students normally present the out-

come of their research projects to other students and to colleagues. Because this isn't feasible in a distance-learning context, students must prepare an online presentation as part of their final submission in the master's research project.

Master's level

The Quality Assurance Agency is an independent body that oversees UK universities such as the OU. The QAA defines a qualifications framework (www.qaa.ac.uk/academicinfrastructure/FHEQ) specifying appropriate levels and approaches for all higher-education qualifications, such as bachelor's and master's degrees. The OU uses this framework when developing the curriculum for its programs.

The QAA also sponsors work on benchmarks for all major subjects, including computing, which more closely specify the key requirements and expected standards for each academic discipline. According to the QAA, subject benchmarks "describe what gives a discipline its coherence and identity and define what can be expected of a graduate in terms of the techniques and skills needed to develop understanding in the subject" (www.qaa.ac.uk/academicinfrastructure/benchmark). The QAA has approved a benchmark for bachelor's programs in computing subjects and has drafted the benchmark for master's programs.

Subject benchmarks don't prescribe the curriculum's detailed content, unlike, say, the ACM curriculum (see www.acm.org/education/curricula.html). The benchmarks do include a list of appropriate topics, rather like that set out for software engineering by the Software Engineering Body of Knowledge (www.swebok.org/home.html). However, the QAA benchmark covers a wide spectrum of possible computing topics and no feasible program could cover all of it—rather, it indicates currently appropriate topics that program designers might choose to use.

Regular inspections and reviews, both internal and external, ensure that universities comply with these academic requirements. All UK universities have their courses scrutinized by external examiners. The QAA defines an examiner's role as follows: to verify that academic awards are appropriate and to help institutions maintain and assure academic standards and ensure that assessment processes are sound (www.qaa.ac.uk/academicinfrastructure/codeOfPractice/section4).

Each level in the QAA framework has associated qualification descriptors. These give examples of each level's expected outcomes and demonstrate the progression between levels. Flexibility is possible, as the outcomes are meant to serve as a framework, not a rigid set of requirements. Here, we summarize the expected outcomes for a master's program:

- Much of the study undertaken at the master's level will have been at, or informed by, the "forefront" of an academic or professional discipline. Students will have shown originality in the application of knowledge, and they will understand how the boundaries of knowledge are advanced through research. They will be able to deal with complex issues both systematically and creatively, and they will show originality in tackling and solving problems.
- They will have the qualities needed for employment in circumstances requiring sound judgment, personal responsibility, and initiative, in complex and unpredictable professional environments.

Many courses in the OU's master's in software development use the assessment either in coursework or exams to explicitly encourage students to engage in current research and practice in the field. Some might also supply a reader, a booklet, or an online folder of key academic or industry papers, and the OU offers extensive online library facilities. Students typically use the reader for their assessment, which is geared toward the development of reflective skills[2] and critical awareness of different points of view and perspectives, as required at the master's level.

For example, in course M882 Managing the Software Enterprise, we base the assignments on industrial case studies and academic papers. We ask students to consider how the companies portrayed in the case studies addressed issues such as ethics, software procurement processes, and so on. We use the academic papers to form an essay-based question that seeks to engage students in a deeper reflection about major topics (such as the role of human motivation in software development). We follow this up in the exam's second part, which is a long essay on a given theme based on two or three papers selected by the students with guidance from their tutors.

Many students use time spent traveling or waiting at airports to study the course materials.

Students can build some coursework assignments, and potentially projects, around their work experience.

Professionally accredited

It's very important to our main constituency of IT practitioners and to their employers that our courses have professional recognition. The main UK professional body for computing is the British Computer Society (www.bcs.org). The BCS accredits programs at universities so that graduates can more easily become full members of the professional body after an appropriate period of experience. This can also lead to the prestigious status of chartered engineer (CEng). This status is fully equivalent to chartered engineers in long-standing engineering professions such as mechanical or civil engineering. In these more established fields of engineering, you need chartered status for appointment to any senior job or to leadership of a major project—this isn't normally the case in the UK IT industry at present.

The BCS accreditation process is based on the QAA requirements for master's programs, but it makes further demands to ensure programs have appropriate content for the IT profession. These include specific teaching of legal, social, ethical, and professional issues applied to IT and the need for students to complete a practical, problem-solving individual project. Accreditation involves a visit to the university by a team of academics and industrial practitioners who scrutinize program documentation and meet staff and students for detailed interviews. Accreditation normally lasts for five years, after which time a further visit is required for renewal.

After the most recent BCS visit in May 2005, the OU computing programs, including the master's programs, were fully accredited by the BCS for five years. The BCS accredited our master's programs as specialist awards, which is the higher of the two accreditation levels (generalist and specialist) available. The Institution of Engineering and Technology (www.theiet.org), another important UK professional institution, also recognizes our master's programs.

Aimed at IT practitioners

Our curriculum is constantly exposed to industrial scrutiny and external academic oversight, including that provided by our students, their employers, our course developers, and tutors.

The topics we teach and the ways we present them must navigate between two extremes. We generally don't consider primarily theoretical topics as appropriate for a whole course. For example, mathematical formal methods have in recent years gone from a whole course, to part of a course, to being dropped completely. However, we must cover topics at an appropriate academic depth—we're not offering pure training. We maintain an appropriate degree of rigor in the way we teach software development. For example, M885 Analysis and Design of Enterprise Systems reinforces a rigorous approach to modeling with consistent annotations using both natural language and the Object Constraint Language. We base our approach to the development of modeling skills on the practical application of an iterative and incremental process to case studies, with well-defined modeling perspectives and an emphasis on rigor and precision.

Students can build some coursework assignments, and potentially projects, around their work experience. For example, M865 Project Management asks students to apply the principles and techniques taught in the course to a real project that they've worked on. Similarly, the M886 Information Security Management coursework incrementally builds up an information security policy for an organization chosen by the student. This tends to work well except in the few cases where students aren't working in the appropriate field or role to have a suitable project or organization. In such cases, we offer alternatives.

This practitioner orientation also makes certain courses attractive to organizations seeking professional development for their staff. For example, a large multinational company recently selected M882 Managing the Software Enterprise to be adapted for continuing professional development of some of its staff, and we're discussing similar arrangements for other courses with this and other companies.

Guided by its educational mission and key characteristics, the OU's research program in computing continues to innovate in the creation of learning materials and their delivery. Areas for further improvement include the automatic assessment of student assignments, agile development processes, social and collaborative software, and privacy requirements management.

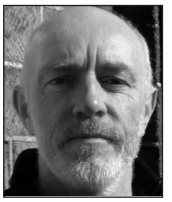
Future directions for the program will likely involve responding to the increasing globalization of IT development and the consequent movement toward more strategic and business-oriented skills. We also anticipate increasing our use of online methods to appeal to a more

global audience. The OU has several active research projects in online education, including the use of virtual learning environments, and we hope to take advantage of some of these in the near future.

References

1. *Agile Manifesto*, www.agilemanifesto.org.
2. D. Schön, *The Reflective Practitioner*, Basic Books, 1983.

About the Authors



Brendan Quinn is the Director of Postgraduate Computing at the Open University. His research interests include software engineering, particularly embedded systems, bioinformatics, and curriculum design. He received his MSc in software systems from Sheffield University. He is a chartered engineer and a member of the BCS. Contact him at the Computing Dept., The Open Univ., Walton Hall, Milton Keynes MK7 6AA, UK; b.quinn@open.ac.uk.

Leonor Barroca is a senior lecturer in computing at the Open University. She was the Director of Postgraduate Computing at the Open University between 2002 and 2005. Her research interests include architectural approaches to software engineering, object-oriented development, component-based development, and state-based specifications of temporal behavior. She received her PhD in computing from Southampton University. She's a member of the BCS. Contact her at the Computing Dept., The Open Univ., Walton Hall, Milton Keynes MK7 6AA, UK; l.barroca@open.ac.uk.

Bashar Nuseibeh is a professor of computing and the Director of Research in the Computing Department at the Open University. His interests include systems requirements engineering and design, software process modeling and technology, and technology transfer. He received his PhD in software engineering from Imperial College London. He is a chartered engineer, a fellow of the BCS, and a member of the IET and holds the Royal Academy of Engineering/Leverhulme Trust Senior Research Fellowship. Contact him at the Computing Dept., The Open Univ., Walton Hall, Milton Keynes MK7 6AA, UK; b.nuseibeh@open.ac.uk.

Juan Fernández-Ramil is a lecturer in computing at the Open University. His research interests include software evolution, effort estimation, system dynamics models of the software process, empirical studies of software-related work, and software management education. He received his PhD in computing from Imperial College London. Contact him at the Computing Dept., The Open Univ., Walton Hall, Milton Keynes MK7 6AA, UK; J.F.Ramil@open.ac.uk.

Lucia Rapanotti is a senior lecturer in computing at the Open University. Her research interests include requirements engineering, software architectures, and groupware systems. She received her PhD in computing science from the University of Newcastle upon Tyne. Contact her at the Computing Dept., The Open Univ., Walton Hall, Milton Keynes MK7 6AA, UK; l.rapanotti@open.ac.uk.

Pete Thomas is a senior lecturer in computing at the Open University. His research interests include e-learning, online assessment in computing, and automatic diagram recognition. He received his PhD in computing from the University of Dundee. Contact him at the Computing Dept., The Open Univ., Walton Hall, Milton Keynes MK7 6AA, UK; p.g.thomas@open.ac.uk.

Michel Wermelinger is a senior lecturer in computing at the Open University. His research interests include software architecture and software evolution. He received his PhD in computer science from the New University of Lisbon. Contact him at the Computing Dept., The Open Univ., Walton Hall, Milton Keynes MK7 6AA, UK; m.a.wermelinger@open.ac.uk.