The Development of a Computing Module Incorporating Capability Curriculum

Peter K Oriogun, *London, UK*

Abstracts

This paper gives a brief introduction to the capability issues in the UK and expands on the University of North London's capability curriculum. A short background to the module of software engineering and elaborates on the rationale for and the development of the new module, software engineering for computer science. It discusses the need for a partnership between educators, employers and the government in moving forward the capability agenda.

Le développement d'un module d'ordinateur incorporant un curriculum de capacité.
Cet article donne une brève introduction des curriculum de capacité dans le Royaume Uni et traite des curriculum de capacité de l'Université de North London. Un bref rappel de l'arrière plan des modules de software pour l'engineering, puis on élabore un raisonnement pour le développement du module nouveau : Software d'Engineering pour la science informatique. L'article discute le besoin de partenariat entre éducateurs, employeurs et le gouvernement en avancant vers un programme de capacité.

Die Entwicklung eines Rechnermodule einschließenden Lehrplans
Dieser Beitrag gibt eine kurze Einführung zum Thema Begabung im UK und leitet über zur Lehrplanentwicklung der Universität von Nord-London. Er zeigt kurz den Hintergrund zum Thema Software Engineering auf und geht dann ausführlicher auf die Begründung und die Entwicklung der neuen Module Software Entwicklung für Computerwissenschaften ein. Er belegt die Notwendigkeit einer Partnerschaft zwischen Erziehungswissenschaftlern, der Wirtschaft und der Regierung bei der Weiterentwicklung der Lehrpläne.

Introduction

This paper reports on the development of a new module, software engineering for computer science (IM283), incorporating the capability curriculum. It also reports on students' performance on the new module during the 1998/99 academic session.

Capability issues in the UK

In the UK the development of competencies is adding tension to the higher education (HE) sector, which is already experiencing a period of significant structural and philosophical change. HE within the UK is undergoing a paradigm shift in moving what was, essentially, an elitist system in which only one adult in seventeen entered this sector (Shephard,1996) to one of the mass education (Harvey and Ashworth,1995).

The Higher Education for Capability (HEC) forum, based in the UK, seeks to address some of the apparent tensions that HE institutions in the UK are experiencing. HEC is predominantly an independent initiative based at Leeds Metropolitan University and the University of Leeds in the UK. HEC (1994) perceive:

> There is a serious imbalance in Britain in the full process which is described by the two words, education and training. The idea of the educated person is that of a scholarly individual who has been neither educated nor trained to exercise useful skills, who is able to understand but not act . . . This imbalance is harmful to individuals, to industry and to society. A balanced education should, of course, embrace analysis and the acquisition of knowledge. It must however, include the exercise of creative skills, the competence to understand and complete tasks and the ability to cope with everyday life (Bridge, 1997).

Education Media International ISSN 0952-3987 print/ISSN 1469-5790 online © 2000 International Council for Education Media
http://www.tandf.co.uk/journals

Affording students the opportunity to take responsibility for their own learning

The University of North London actively supports the higher education for capability movement because it is a broad approach to the curriculum, which aims at making knowledge overtly useful. This ties in with the university's strategic plan of producing employable graduates. Capability is a generalized approach to curriculum delivery, which aims to develop student autonomy as an effective learner. Many different strategies, designed and implemented with this intention would characterise a 'capability curriculum'.

The notion of the capability envelope

The 'capability envelope' is based on the proposition that much important learning goes on through reflection on the learning process itself. That is, at the beginning of the programme, students think about and articulate: where they are now, where they want to be (and why), how they are going to get there and how they will know when they have got there. As they go along, they continuously review their progress and plan the next stage in their learning. At the end, they review what they have learnt in the light of their initial plan. If the 'envelope' is well designed and executed the student should have learnt how to learn and have the confidence, skill and motivation to continue to learn throughout life.

The capability curriculum

The University of North London has recently developed and, from 1998/99, fully implemented a new curriculum based on the development and assessment of six core capabilities across all undergraduate degree programmes in the university. Table 1 shows generic definition of capabilities at threshold levels.

The six capabilities are as follows:

C1: Act Appropriately in the context of social and cultural diversity and the modern environment.
C2: Make ethical evaluations.
C3: Think critically and produce solution
C4: Manage self and relate to others.
C5: Communicate effectively in context
C6: Seek, handle and interpret information.

Each capability has different thresholds for preliminary (first year), advanced and advanced starred (University of North London, 1998) (second and third year) modules. All preliminary and advanced level degree modules across the university have been validated to take the six core capabilities into account. All degree modules require module convenors (the lecturer responsible for the module) to develop and assess two capabilities in a way that ensures that students must have reached the required threshold level, in associated capabilities, in order to successfully complete and pass the module.

Developments in higher education in the UK have led to the requirement for HE institutions assessing their success in producing 'employable' graduates and institutions have had to clarify the nature of 'graduateness'. The University of North London's (UNL) Strategic Plan was drawn up in the context of producing employable graduates, and it is part of a move towards 'higher education for capability' and 'ability based higher education'. Alverno College in the USA has been influential in the development of UNL's new curriculum (Mumford, 1999). In the Faculty of Science, Computing and Engineering the new curriculum is the consolidation of a number of years of experience of 'skills' based science education.

There needs to be a partnership between educators and employers. This partnership should not be about training to specification but about educating for change and growth. It must be a richer and more inspiring business altogether. The partnership must come through joint recognition by educators and employers (all employers, not just industrial ones) of a shared responsibility: that of trying seriously to enable people to learn and develop their lives. Students' must be a major partner; they will increasingly want the knowledge as well as the specific skills, which put them in charge of their own learning lives. Government also has its role, not least to provide a financial framework for institutions and individuals within which all can plan with greater confidence than at present (Coldstream, 1997).

Table 1 *Generic definition of capabilities at threshold levels*

	Preliminary	**Advanced**
C1	To show awareness of the current, social and technological context of his/her professional area	To evaluate the social, economic, environmental or cultural consequences of professional activity within the subject discipline, against current standards of conduct and practice
C2	To show awareness of the ethical issues relating to professional academic responsibility and area of study	To recognize any ethical constraints which may influence the carrying out of professional activities; consider the present contrasting ethical position and make a reasoned argument for a course of action on ethical grounds
C3	To gather, select, organize and use relevant information from a variety of sources	To evaluate an unfamiliar or complex situation to pose questions and diverse solutions where appropriate, and to be able to redesign solutions where necessary and review outcome after critical reflection
C4	To produce accurate technical writing; obtaining information by asking; participate in group questions and communicate effectively with reference to target audience	To maintain and regularly update a record of work and reflect upon it; prioritize and maintain deadlines for independent and/or collective work and to be able to interact effectively with others by adopting constructive method of working
C5	To identify personal goals, strengths and weaknesses; develop and implement strategies for personal development and to agree working methods as a member of a team and work effectively	To use the conventions and notations normally employed in technical writing in the subject area. To be able to communicate with peers using a range of media as appropriate and to be able to produce a thorough analysis, appropriate to the target audience, of an experiment or investigative study
C6	To apply given tool or method to defined problem; producing an appropriate solution and reflect on the process and outcome	To be able to select an appropriate source(s) or method(s) for acquiring information and/or data; to analyse the data using a variety of appropriate methods and tools and to be able to draw appropriate conclusions, understanding the limitations of the available data

The school's interpretation of capabilities

The School of Informatics and Multimedia Technology (SIMT) has interpreted these capabilities in terms of situations experienced by and behaviour expected of graduates (Mumford, 1999) (see Appendix A, courtesy of M R Mumford, University of North London).

Background: the old software engineering module (CO209)

In the four years up to, and including 1997/8, the software engineering module (CO209) has been taught with one main lecturer and five tutorial assistants. Tutorial assistants monitor and deliver the weekly deliverables of the coursework assignments, and provide useful feedback to students regarding their progress. Each tutor group has a maximum of six coursework groups, with each coursework group having a maximum of six students. The class size is typically between 150–180. This figure includes those students registered for the module and reassessed students. The module is a core for three modular degree programme of study pathways – Computer Science, Computing and Business Information Systems.

All modules taught at the university on the modular degree scheme are fixed, weekly, four-hour blocks. The four hours are split into a two-hour lecture and two-hour tutorials. The tutorial focuses mainly on the group coursework assignment for the module. The coursework element of the module, relied heavily on students having a certain degree of knowledge and understanding of how to synthesise and analyse a software engineering problem, using a combination of paradigms and the mapping of that analysis onto a design.

The module was redesigned in 1994 to incorporate flexible learning in its delivery. The introduction of flexible learning was to encourage students to be more responsible for their learning by changing the timetable to reduce the length of formal lectures, so as to release more time for private study and personal interaction between the lecturer and students. Students can also work through a set of Study Guides (Jones, 1994) after each lecture at their own time and pace. The recommended main text for the module (Pressman, 1998) covers far more than can be taught in a single semester unit.

The new computing module: software engineering for computer science

The new software engineering for computer science module (IM283) was developed to incorporate the capability curriculum (University of North London). The module is also starred (University of North London, 1998), sometimes referred to as a third level module in the UK. The aim of the module is to enable students to understand, participate in, control and manage the process of developing large complex software products. It is intended for students with the ability to program effectively in at least one programming language (preferably two different languages), together with the necessary data modelling techniques to analyse, design and implement a detailed software project as part of a team. The assessed capabilities are C2 and C4 (University of North London). The recommended text for the module is Pressman (1998). A further prerequisite for IM283 is a completion in the systems analysis and design module (IM201). Table 3 maps learning outcomes to capabilities.

A typical capability mapping for a single honours degree pathway

Table 2 illustrates a typical programme plan for a single honours degree pathway in computing. The programme for each student should have a good blend of the six capabilities at preliminary and advanced levels. Each module

Table 2 *Capability maps – computing (single)*

	C1	C2	C3	C4	C5	C6
Core units						
IM101		X				X
IM102		X	X			
IM201	X			X		
IM202			X			X
IM203			X	X		
IM207		X	X			
IM250				X	X	
IM250				X	X	
IM283		X		X		
CF102		X		X		
CF103	X		X			
Designated units:						
IM100	X				X	
IM103				X	X	
IM107	X				X	
IM109	X	X				
IM208			X			X
IM213			X			X
IM216				X	X	
IM218		X		X		
IM260	X					X
IM269			X			X
IM252	X				X	
IM271	X			X		

Table 3 *Mapping learning outcomes to capabilities*

Assessment instrument	Weight	Assessed learning objectives	Assessed capabilities
Coursework 1			
Individual contribution	5%	LO1, LO3, LO6	C4
Group contribution	35%	LO1,LO2,LO3, LO6	C4
Coursework 2			
Professional issues report	10%	LO4, LO5	C2
Examination	50%	LO4, LO5, LO6	C2, C4

must assess two core capabilities in its assessment instruments within the syllabus. A pass in the module assures that the student had performed satisfactorily in the core capabilities (since the learning outcomes of the module must map the capabilities). For more information on module codes see (SIMT).

Learning outcomes

On completing this module the student should be able to:

LO1: Develop a project plan as part of a team (not more than six members) for a specific software engineering problem; organize and plan their individual aspect of the overall project plan; use available CASE tools (within the university or otherwise) to plan schedule and implement the required software product and produce a test plan for the proposed software.

LO2: Select appropriate information systems development methodology to analyse, design and implement a software for the problem at hand. As part of the programme development team, produce on-going documentation of the system under development with appropriate feedback from the tutor; updating the work after evaluation and reflection within the group and participating in a constructive manner to produce effective solution.

LO3: Presentation of the completed software engineering product collectively and individually by participating in the documentation and writing of the group report and documenting individual contributions within the group by writing individual report at the end of the project.

LO4: Evaluate professional and ethical issues within the computing industry.

LO5: Consider, compare and present different ethical situations with justifiable argument by recognizing any ethical constraints which may influence the execution of professional activities within the computing industry.

LO6: Understand and relate to different software life cycle models; solve problems relating to the analysis, design, implementation, testing and documentation of software.

Software engineering for computer science: teaching and learning methods

During the lecture periods, students were introduced to the module, exposed to different software life-cycle paradigms and project management techniques (10 hours). The importance of group-work was an on-going theme throughout the module. Group coursework assignment, was supervised by a designated tutor. Tutors advised on the weekly deliverables for the module; they also monitored, amended and/or corrected the intermediate deliverables submitted by each of the groups allocated to their tutor group throughout the semester (24 hours). Software quality issues was a subject of two lectures (4 hours) and the various types of software testing strategies was the subject of one lecture (2 hours). Students were expected to spend some time (72 hours) unsupervised for their group coursework also some unsupervised time for individual coursework (24 hours).

Assessment instruments for software engineering for computer science

Coursework 1

The assessment of the new module (software engineering for computer science) consists of three components: two in-course assessments (ICA) and an examination. The first in-course assessment was a group coursework. It primarily dealt with the analysis, design and computer implementation of the assigned software project (from a list of prescribed projects). The target programming language for the implementation was to be decided by each group according to their programming expertise. A group presentation was required in the last two weeks of the semester together with a group report and individual report. The individual report should document the contribution of each member of the group. This may be in the form of a logbook with some critical appraisal of the project in general. Demonstration of the implementation is scheduled for week 11. This coursework predominantly focused on capability C4.

Coursework 2

The second coursework was on Professional issues (Sadler). This coursework was assessed through case studies posing a software related ethical dilemma whose resolution involves recourse to the indicative legislation and codes. This coursework focused on developing the capability C2.

Examination

The examination was in two parts, the first part had six questions from which students were required to answer four. The four questions made up two-thirds of the total marks for the examination. The subject of the first six questions was covered during the weekly lectures. Students were expected to read about the topics in more depth during their private study.

The subject of the compulsory question for the examination was not the subject of any lecture. The topic was issued to students in the first week of the semester. It was up to individual student to research the topic for the question. Students were expected to spend up to 1 hour on this particular question during the examination. The examination was designed to test each student's understanding of the software engineering process. This took into account the knowledge and understanding already gained from the two in-course assessments.

Access to course materials on the web

Students had access to course materials via a link from the web page of the module convenor. This meant that students have direct access to course materials at home provided they have a link to the web. The course information provided includes course outline, scheme of work, books, coursework information and the special examination question for the examination. The IM283 web page (Software Engineering for Computer Science web page) was designed to effectively utilize distance and flexible mode of course delivery.

Assessment results

Tables 4 and 5 below shows the average coursework mark for software engineering for computer science is 4.6% higher than the average coursework mark for CO209 the previous year. Similarly, the average examination mark for IM283 is higher than the average examination mark for CO209 the previous year by 7.7%. The overall module mark went up by 5.9% for IM283 compared with CO209 the previous year.

It is not possible to conclude from Table 5 that the marginally improved students' performance on the new computing module compared to the previous year was due to the incorporation of the capability curriculum within the syllabus. The course team in the 1997/98 academic session was different from the course team during the 1998/99 academic session, as a consequence, the difference in performance could be due to the tuition received. Table 5 does not differentiate between courses taken by students. In order to determine the effect of the capability curriculum on the results of students, a detailed analysis of students graduating within the capability curriculum framework is required. This will involve a partnership between past students, their employers and the university.

Table 4 *Software engineering results summer 1998*

	ICA	Exam	Unit
Number	134	129	134
Average	58.4	43.6	50.4
Standard deviation	11.9	14.8	12.2
Weighting	50	50	

Table 5 *Software engineering for computer science results summer 1999*

	ICA	Exam	Unit
Number	88	84	88
Average	63.0	51.3	56.3
Standard deviation	9.1	16.4	12.7
Weighting	50	50	

Performance of the first batch of students on the new module

The performance of students on software engineering for computer science (IM283) was marginally better than that of software engineering (CO209) the previous year. The group coursework was particularly well done by the students. Students used a variety of paradigms for the analysis and design of their project. The average module mark scored by students completing IM283 is higher than for CO209 the previous year.

It was evident from the group reports submitted by students on IM283 for the first coursework and the individual contribution to the first coursework that, students understood most of the problems associated with group work. They were able to manage themselves and relate to others as prescribed by capability C4. The second coursework gave students the opportunity to really think about a professional ethical dilemma situation and resolving the dilemma with sound reasoning based on various legislation and code. Students were thus, able to make ethical evaluations, satisfying capability C2.

Conclusion

The results are encouraging since the incorporation of the capability curriculum within the syllabus. If students have the capabilities, it is expected that they will be more employable. In the future we will be able to verify these expectations, through the university's First Destination Monitoring, which is an on-going procedure carried out by the careers office of the university.

Acknowledgements

I wish to take this opportunity to thank the following individuals for their contributions to this paper. Ms Elli Georgiadou for reviewing the paper and her invaluable contribution to the final product. Dr Tom Boyle for reviewing the paper and his invaluable comments and suggestions. Professor Jim Yip, my Head of School for his encouragement and sponsoring me to attend the Euro Education 2000 Conference. Mr CJ Sadler who teaches the professional issues aspect of the module IM283. Mr MR Mumford for the reproduction of extracts of his paper in Appendix A. Mrs CV Smallwood, the Chair of the Faculty Curriculum Action Group.

References

Bridge, A (1997) The role of competency based training with capability learning, HEC Conference.

Coldstream, P (1997) Guest opinion: training minds for the future world of work, *HEC Capability*, 3, 2, np Available http://www.lle.mdx.ac.uk/hec/partners/journal/3-2/0-1.htm [accessed Oct 21st, 2000].

Definitive Document Awards (1998) based in the School of Informatics and Multimedia Technology, Faculty of Science, Computing and Engineering', University of North London, 31–118

Jones R (1994) Adapting the delivery of a computing unit for flexible learning, The Second All-Ireland Conference on Teaching of Computing, Dublin, Ireland. September 1994

Mumford, MR (1999) The capability curriculum – a briefing document for hong kong computer institute staff, University of North London, SIMT, October.

Pressman RS (1998) *Software Engineering: A Practitioner's Approach, European Adaptation*, McGraw Hill.

Sadler, CJ, Professional issues home page, available online: http://www.idun.unl.ac.uk/~11sadlerc/htm2.html

SIMT Definitive Document, available online: http://www.homepages.unl.ac.uk/~mumfordm/revalidation/simtsyl.htm

Software Engineering for Computer Science Page: http://www.unl.ac.uk/simt/im283

University of North London- Capability Curriculum Page: http://www2.unl.ac.uk/~head/capabil.htm

Biographical note

Peter Oriogun BSc, MSc, MBCS, ILTM is currently a Senior Lecturer in Software Engineering at the University of North London, England. Before joining the UNL, he worked as a Systems programmer for Rolls Royce plc (Aero Engine Division), a Lecturer in Software Engineering Methods at Sheffield Hallam University, a Lecturer in Computing at Newham Community College, a Lecturer in Information Technology at South Bristol College and for Community Service Volunteers as a Computer Consultant. His research interest is in Teaching and Learning of Computing in Higher Education, and software metrics. He is a professional member of the British Computer Society, and a professional member of The Institute of Learning and Teaching in Higher Education.

Address for correspondence

Peter K Oriogun, School of Informatics and Multimedia Technology, University of North London, 2–16 Eden Grove, Holloway Road, London N7 8EA, U.K; e-mail: p.oriogun@unl.ac.uk

Appendix: the school's interpretation of capabilities

Act appropriately in the context of social and cultural diversity in modern day environment

Software specification, design, development and operation are activities involving a wide range of 'stakeholders' such as professional colleagues, users of the system, suppliers of hardware and software and the general public who are impacted directly or indirectly.

A professional in the general area of computing must be able to understand the needs and attitudes of all these groups, typically including the full spectrum of social, educational and ethnic backgrounds, and tailor their behaviour and actions to respect these needs and attitudes, in order to work effectively.

Make ethical evaluations

Computer systems affect the lives of other people, both as employees and as a member of the general public. The effects may arise through the correct operation of these systems, eg. in reducing employment, and/or through their failure, e.g.. the failure of an avionics system. In addition, Information Technology as a whole gives rise to potentials for the abuse of confidentiality (e.g. unauthorized disclosure or hacking) and social control (e.g. AI software processing images from CCTV systems covering public places).

A professional in the area is expected to be able to resolve any conflict between their personal interests, the interests of their employer and/or customer and the public interest and not to take blind advantage of the potential of the technology irrespective of its consequences on others. In doing this he/she will be guided by their personal ethical beliefs, personal codes of conduct (such as the BCS) and the current legal framework (e.g. the Data Protection Act in the UK).

Think critically and produce solutions

This capability is almost a definition of the software professional in any area of the field. Whether a systems analyst subjecting operations to rigorous critical analysis and devising designs to meet requirements and problems of the user, or a programmer taking a software design and implementing it through the potentiality (and problems) of the software development environment to him/her. A professional in any area will demonstrate these capabilities through effective working.

Manage self and relate to others

Software development and support is almost universally a team process. Professionals in this field are required to work as effective team members which involves both delivering their own assigned work, on time to quality standards, and contributing to the common team purpose through supporting others in their work and integrating the effort of the team. In addition, professionals, in delivering and maintaining software systems, are part of a wider team consisting of the organization as a whole. They must therefore organise their efforts to contribute to the wider goals of the organization and work effectively with colleagues with very different backgrounds and perceptions.

Communicate effectively in context

As mentioned in the previous heading, software development and support are team activities involving communicating with other professionals within the team but also with the users or customers of the software, who form a much more diverse group. Computing professionals must master the ability to communicate using the formal professional techniques, established for documentation and communication in the development and support processes, and the ability to communicate with users in whatever is the appropriate medium (speech, writing, diagrams etc.) and vocabulary and mindset of their audience.

Seek, handle and interpret information

In many aspects of their work (such as requirements definition or problem identification) software professionals are required to gather appropriate data, process it to obtain that information which is necessary for their purpose and extract from that information the answers they need (e.g. the users' requirements or the characteristics of the software bug). Most aspects of the daily work of professional in the field of computing involve one or more of these activities, from the effective use of technical documentation (e.g. specification and technical manuals) to interpreting error messages and other information generated by computer based integrated development environments.