Use and Evaluation of SWEBOK by Postgraduate Students

J. Barrie Thompson and Colin J. Hardy
School of Computing Engineering and Technology
University of Sunderland, St Peter's Way,
Sunderland SR6 0DD, United Kingdom.
E-mail: barrie.thompson@sunderland.ac.uk
Colin.hardy@sunderland.ac.uk

Abstract

Background information on the Software Engineering Body of Knowledge (SWEBOK) project is presented and the relationship between SWEBOK and the IEEE Computer Society (IEEE-CS) and the Association for Computer Machinery (ACM) joint Software Engineering Coordinating Committee (SWECC) is detailed. The use of the SWEBOK guide to support learning and teaching on a software engineering module that is shared between two masters level programmes is outlined. Details concerning the two masters level programmes is given including their respective structures. The particular software Engineering module (entitled "Systems Engineering") is outlined and the theoretical and practical strands within it described. The assessment elements associated with the theoretical strand (research paper proposal, research paper, position paper, and discussion group report) are explained along with the relevant operation of the module. Use of the SWEBOK by the students and their evaluations are presented. Finally some general remarks are given along with details of future plans..

1. Introduction

By late spring 2000 it appeared that substantial progress was being made towards establishing Software Engineering (SE) as a profession. In particular, the IEEE Computer Society (IEEE-CS) and the Association for Computer Machinery (ACM) were actively working together via their joint Software Engineering Coordinating Committee (SWECC) which had been made responsible for coordinating, sponsoring and fostering all the various activities regarding SE within their spheres of operation. These included areas such as: standards of practice and ethics, body of knowledge, curriculum guide-lines, and exam guidelines. Many of these projects had made significant progress and results from them were in the public domain [e.g. 2 and 5]. Then, in the summer of 2000 ACM decided to withdraw from the IEEE-CS/ACM Software Engineering Coordinating Committee (SWECC) [1] and with this much of the progress that had been made was thrown into question.

A key requirement for a discipline to truly exist is that there is an accepted Body of Knowledge (BoK) that represents and defines the scope of the discipline [4]. Also, such a BoK must exist if educational institutions, professional bodies, and licensing organisations are to produce meaningful curriculum and examinations. Thus the Software Engineering Body of Knowledge (SWEBOK) project [8] promoted by SWECC could be seen to be fundamental to the efforts towards a SE profession. This project, which has been coordinated from the University of Quebec in Montreal, Canada, has taken a three phased approach consisting of Straw Man, Stone Man, and Iron Man phases. It represents a very systematic piece of work that has attempted a

Proceedings of the 15th Conference on Software Engineering Education and Training (CSEET'02)
1093-0175/02 $17.00 © 2002 **IEEE**

broad and international approach in its reviewing process. It should therefore produce a much more authoritative BoK than that which has been produced by the Software Engineering Institute at Carnegie Mellon University under contract in association with the US Federal Aviation Administration [6]. Of particular note is that the whole of the reviewing process has been visible and is available on the project's web site. [8]. However, the very features that have been highlighted above also have a downside. Clear difficulties with SWEBOK are its size, complexity and rather long timetable. During April 2000 a more or less finalised edition of the Stone Man Version of the Guide (version 0.7) was released on the project's web site. The Stone Man version was then further refined and renamed as the Trial Version (version 0.95) which was available from February 2001 for public use and evaluation (the name was apparently amended so that users will not incorrectly assume that the contents have been "set in stone"). The results of public use will subsequently feed into the Iron Man phase of the project. Further details of the project's progress can be found in the September 2001 issue of FASE [3].

Unfortunately, in addition to difficulties outlined above, a major problem with the SWEBOK project has been that some members of the ACM believe that that there was too close a relationship between the it and the model adopted for licensing Software Engineers in Texas [7]. This perceived relationship appears to have been at least partly instrumental in ACM's withdrawal from SWECC [9] and has caused significant negative reactions to SWEBOK within the SE community [10]. SWEBOK may not be perfect, but it is a piece of work that should not be ignored or simply thrown away. It needs to be used, evaluated and built upon.

The following sections of this paper detail a use and evaluation of the SWEBOK document within a masters level SE module (entitled "Systems Engineering") which is taken by part-time postgraduate students following two "sister" masters programmes at the University of Sunderland in the UK. These are the M.Sc. in Computer Based Information Systems (CBIS) and the M.Sc. in the Management of Information Technology (MIT). Section 2 gives details of these two masters level programmes including their respective structures. Section 3 outlines the Systems Engineering module and describes the theoretical and practical strands within it. The assessment elements associated with the theoretical strand (research paper proposal, research paper, position paper, and discussion group report) are explained in section four along with details of the operation of the module. In section five use of the SWEBOK by the students and their evaluations are presented. Finally, in section six some general remarks are given along with details of our future plans.

2. The MSc. Programmes

The CBIS masters is a "conversion" programme for graduates from non-computing disciplines who have little or no computing experience and who are interested in gaining a theoretical and practical understanding of the construction of computer-based information systems. The MIT masters is primarily intended to assist graduates in computing, or others with an equivalent existing background, become hybrid managers. That is, managers who combine information technology and computing skills with business and organisational skills in order to ensure the effective deployment of information technology in their organisation. In many cases the CBIS programme is also undertaken by graduates who also become hybrid managers [11]. The students on the CBIS programme often already have well developed management skills but they lack IT knowledge. By undertaking CBIS they too gain hybrid skills but they are achieving them from a differing viewpoint to the MIT students. Both the CBIS and MIT

programmes consist of two main elements: a taught element and a master's level project which is normally externally sponsored. The CBIS programme was originally developed in 1989 following discussions between the then Polytechnic of Sunderland and the UK government's Training Agency, which was a part of the Department of Employment. The programme commenced operation in full-time mode September 1989. In 1990 and 1992 respectively part-time evening and block mode versions of the CBIS programme were introduced. The latter being partly supported by distance learning materials. These versions of the programme are undertaken over a three-year period. Additionally, in 1992 the then three modes of the programme were revised to fit into a University-wide modular Credit Accumulation and Transfer Scheme (CATS) which supported more flexible approaches to learning. At the same time the MIT programme was developed. This programme was produced by using some of the modules from the CBIS programme, modules from the Sunderland Business School's MBA programme, plus three MIT specific taught modules and an MIT specific project module. Since then both programmes have been revised and developed to reflect new technological developments and the needs of the market place. More details of these developments can be found in [12, 13]. The following two subsections briefly describe the versions of the programmes that were taken by the part-time students who are the subject of this paper.

2.1 MSc in Computer Based Information Systems

In the CBIS programme the taught element is equivalent to 30 weeks full-time study and the externally sponsored project is equivalent to 18 weeks full-time study. The detailed aims of the programme are to:

1. promote a critical awareness of the natures, roles and limitations of computer based information systems,
2. develop the knowledge, skills and understanding needed to specify, design, implement, document and furnish continued support for an effective computer based information system, either alone or as part of a team,
3. allow the student to consolidate and display skills, knowledge and awareness in a selected application area for computer based information systems and gain experience of implementing such a system.

The structure of the relevant version of the programme (1995 version) is shown in Table 1. The value of each module within the CATS system is shown. In this system 10 CATS points is equivalent to approximately 4 weeks full time study (120 hours). Part-time students normally complete four taught modules in each of their first two years of their programme. During each year they attend the University for formal sessions for two nights each week for 30 weeks. During each 15 week block they study two modules in parallel, one module per night. In the third year they undertake their individual project.

Within the CBIS programme the modules: Application Building, Network and Computer Hardware, Software Environments, Object Orientation, Relational Database Systems, and Software Construction Modules all have a very technical emphasis. Systems Engineering, Applied Knowledge Engineering and a significant part of Computing and Research Skills are all very research orientated. The modules which take a wider view with regards to human aspects such as stakeholder involvement, ethics and professionalism are: Computing and Research Skills, Systems Development, and Systems Engineering.

Table 1: Topics taught within CBIS Programme (1995 version)

Module	CATS value	Stage
Computing and Research Skills	10	
Application Building	5	
Systems Development	10	Certificate Stage
Network and Computer Hardware	5	
Software Environments	5	
Object Orientation	10	
Relational Database Systems	5	Diploma Stage
Systems Engineering	10	
Applied Knowledge Engineering or Software Construction	10 each	Optional Modules
Project	50	M.Sc.

2.2 MSc in Management of Information Technology

The overall aim of the MIT programme is to: Provide the knowledge and understanding needed to take a strategic view of the role of information technology in the operation of an organisation, and to lead the effective deployment of information systems to meet corporate objectives.

The structure of the relevant version of the programme (1997 version) is shown in Table 2. It should be noted that the Certificate, Diploma, and MSc Project stage within MIT each have the same value of 40 CATS points. This is different to CBIS where the values are 35, 35, and 50 respectively. The balancing of the stages is a University-wide change that will occur as each programme comes up for review and revalidation. The project has not really dropped in value as now there is within the Diploma stage a module "Project Proposal" (value 10 CATS points) which complements the project (valued at 40 points).

Table 2: Topics taught within the revised MIT Programme (1997)

Module	CATS value	Stage
Research, Ethical, Legal, and Professional Issues	10	0
IT Project Management	5	
Systems Development	10	Certificate Stage
Managing People	5	
The Environment of Business	5	
Information Systems Strategy	5	
Quality Management of IT	5	Diploma Stage
Systems Engineering	10	
Strategic Management	5	
Project Proposal	10	
Project	40	M.Sc.

The modules: Managing People, The Environment of Business, Strategic Management, and Managing Financial Resources all belong to the Business School. The modules: IT Project Management, Information Systems Strategy, and Quality Management of IT are all particular to the MIT degree. The modules: Research Ethical Professional and Legal Issues, Project Proposal, Systems Development, and Systems Engineering are all shared with other MSc programmes in our school, the latter two being shared with the current version of CBIS. Until 1997 MIT used the introductory CBIS module Computing and Research Skills which is still being used on the Distance learning mode of MIT.

3. The Systems Engineering Module

The module aims to ensure that students are aware of the need to engineer software systems that are maintainable and can be enhanced during their operational lives. A copy of the formal module descriptor can be found in the Appendix to this paper. The module is delivered in two strands: a theoretical strand and a practical strand. For the 2000/2001 academic year the theoretical strand considered several elements which were based on the Knowledge Areas identified in "A Stone man Version of the Guide to The Software Engineering Body of Knowledge" Version 0.7 i.e.

Software Requirements,
Software Design,
Software Construction,
Software Testing,
Software Maintenance.
Software Configuration Management,
Software Engineering Management,
Software Engineering Process,
Software Engineering Tools and Methods,
Software Quality

The SE aspects that were covered provided students with the opportunity to critically assess and appraise the relevance and use of a variety of factors that contribute towards the engineering of systems. Each student had the opportunity to focus on one of these aspects in more detail and the results of this analysis were presented in a research paper developed within the module. At the end of the module there was a "Postgraduate Workshop on Systems Engineering". At this the students had the opportunity to discuss the wider aspects of SE and the usefullness of the Guide to the Software Engineering Body of Knowledge (SWEBOK) in the light of the knowledge gained from their research. The practical strand of module endeavoured to mesh together issues covered in the theoretical elements of the module. Aspects associated with quality, quality assurance, standards, metrics, and management were examined. Students also had the opportunity to experience for themselves the problems that can result when small "enhancements" are made to software. There was also a small group assessment exercise concerned with software quality.

The module was structured to provide an initial set of "scene setting" seminars which provided overviews into particular SE topic areas. Students then formed themselves into small self-help groups (of three or four students). Each group then selected a topic area from those listed above and each student then worked on producing a final individual research paper (to conference standard) on an aspect within the topic area. Students were advised to devote some time each week for working in their self-help groups that were focused around the specific topics being investigated. During periodic tutorial

periods the self-help groups were expected to go through a structured (documented) process in the development of the individual research papers. The process followed was intended to mimic (as far as is possible) the processes that are important in the engineering of good quality software products. Therefore, although the product being developed is a research paper, the components of planning, quality assurance, structured walk-throughs, use of metrics and adherence to standards should be evident. The idea behind the process was that the students would learn from each other, but it was made clear that the final delivered papers must be the relevant student's own work. At the end of the module (last two evenings) each student had to take part in one of two discussion groups (in each discussion group there was a distribution from each self-help group). Each student was required to bring to their discussion group a position statement based on their research and experiences. Each discussion group was timetabled for six hours and each group had to produce a formal report at the end.

The module assessment was as follows:

Portfolio (60%)

(i) Research paper proposal (outline, structure and key references).	10%
(ii) Research paper on nominated, and approved, topic.	30%
(iii) Structured checklist for use in Software Quality TCTs.	10%
(iv) Position paper for Workshop Discussion Group.	10%

Time Constrained Tests (40%)

(i) Workshop Discussion (resulting in discussion group report – one per group).	20%
(ii) Software Quality Practical (group based).	20%

Thus research formed a major element within the module. Each student not only had to produce a final high quality paper, but had to produce a proposal (similar to the extended advance abstract asked for by many conferences), a position paper for a workshop, and participate in the workshop and the production of the workshop report.

4. Use of SWEBOK and Student Evaluation

The module was taken by 15 mature graduate students (12 CBIS, 3 MIT) and involved attendance at the University for one evening for 15 weeks. An overview of the SWEBOK project and the guide were presented at an early stage in the module. Each student-self help group negotiated their selection of a topic area from SWEBOK. There were a total of four groups and no two groups finished with the same topic area. Students then used the relevant part(s) of the guide to help them in the initial development of their research papers. The two discussion groups at the end of the module (of 8 and 7 students) were designed to give the students the opportunity to discuss SWEBOK. The discussion sessions took place on 27th June 2001, and 4th July 2001. Each student was required to bring to the first discussion sessions his/her position paper, which outlined the points of view of the member, and various discussion points, with a bias towards the chosen research area. The topics given to the discussion groups and the significant points that appeared in the reports are detailed in the following two subsections. In these two subsections it should be noted that all the text is taken verbatim from the student reports with less relevant parts being simply edited out, concatenation of over short paragraphs and correction of major grammatical errors.

4.1 Text from Group A's Report on: The Future of Software Engineering Education and how the SWEBOK Project can help in this.

"The discussion commenced with the view that if Software Engineering education should be academically or industry led. It was commented that academic qualifications alone were not seen by employers as sufficient training to be competent at doing a particular job. Rather the academic qualifications served as a foot in the door, but practical experience was required before employers would consider a person had the competence to carry out a job. Following on from this point, SWEBOK was considered in this light and whether there was fair and valid representation from both academic and industry sources. The conclusion was drawn that maybe SWEBOK was too academically biased to set the standard for Software Engineering, and this would hinder its support from other organisations.

The group discussed the fact that, although there are some valid points, and guidelines, the SWEBOK project was weighty, and the fact that the references are mainly offline, and companies find these references difficult to access. The SWEBOK project is an extremely useful reference for students of Software Engineering, however the size of the guide, and its lack of a comprehensive indexing system, also the various references which are not available online or in public libraries, makes the guide less valuable for those users outside of education. There was also the point that, by the time the latest version of the SWEBOK project is finished, the industry will have moved on, and it is argued that it will be out of date, and will be required to start again. This means that the work will only be of any true use if it is periodically updated, as the industry is constantly changing, and it will be out of date. Due to the nature of the referencing, SWEBOK, although a good background reference tool, will be seen as an inadequate educational resource if its references and topic coverage are outdated. SWEBOK would be more suited to an online database with the functionality of being dynamic and searchable, and the references would be constantly up-to-date. It was argued that the SWEBOK project can be seen as a time capsule, in other words, a view of Software Engineering at a particular part of time, and what methods were used at the time.

There was also the issue of professional recognition for Software Engineering. In the UK, there is no requirement for Software Engineers to be accredited or licensed, it is the professional consciousness. However, degree courses are accredited by professional institutions, which includes the British Computer Society. This accreditation allows students to become Chartered Engineers, if they complete professional examinations, and are recognised as engineers. In the USA, there was 1998 legislation requiring a software engineer to be accredited with Professional Engineer status to apply their trade in the state of Texas has raised the spectre of quality software engineers. The legislation aims to ensure that there would be the reassurance that software developers have received the appropriate education and training and would adhere to the code of ethics and professionalism established by the board that licensed them. The SWEBOK project would be used to develop and accredit university curricula and license engineers. The group argued that this would not be a good idea to use the SWEBOK project, as this would not be efficient, as software engineers are required to have a wide range of skills, and they need to adapt quickly, and as mentioned earlier, the SWEBOK project would not be able to adapt quickly enough, and it is more academically focused, and software engineers would be working in industry, and as mentioned earlier, this would be inefficient. The SWEBOK project needs to be more commercially focused.

The SWEBOK project can help in Software Engineering education. It provides a framework for courses to be based on. The SWEBOK project allows students to base research on the various subjects presented in the document. This is very beneficial, as Software Engineering is a vast subject, and as Software Engineering is constantly changing, the documents change to reflect these changes. However, this is mainly suitable to academic. The SWEBOK project has some valid principles, and guidelines, but the main problem is that it is academically focused. The problem is that, this is not necessarily relevant to the Software Engineering industry. It is more academically focused, the SWEBOK project needs to be more commercially focused. It is difficult to achieve a consensus between academic, and industry, as companies tend to use parts of traditional methods, and to adapt them to their own needs.

There needs to be more cooperation between academic, and industry to be able to be more effective. It was argued that courses need to be more vocational, and they will be more relevant to industry. The SWEBOK project is driven by academics, but it is argued that the Software Engineering industry should be industry led. It is argued that academics should support industry, and universities should communicate with industry, and develop courses based on industry. However, this could be difficult, as different companies use different methods. The SWEBOK project can be beneficial to students to have an understanding of the various areas of Software Engineering, but it does not necessarily reflect what is happening in industry. If there is more cooperation between academic, and industry, is would be possible to develop a body of knowledge, which is relevant to both academic, and industry."

4.2 Text from Group B's Report on: Personal Experiences with SWEBOK - beneficial or not?

"SOFTWARE is arguably the word's most important industry. The presence of software has made possible many new businesses and is responsible for increased efficiencies in most traditional businesses. SWEBOK can serve as a framework that can be used to conduct research into software engineering methods, techniques, and practices etc. Hence, this is definitely a beneficial point for an academy development, which provides masses of guidelines and references for general research areas. SWEBOK provides a consistent definition for the practice of software engineering; a definition of the scope of software engineering and its relationship to project management and computer science.

These statements were accepted as a fair summary of the current state of the industry. However, a major point discussed was how useful the SWEBOK product was in real life situations, and how the guide could be applied to actual problems. A number of points were raised by several contributors: the style of the writing (prescriptive narrative), use of jargon, the academic bias, and the omission of non-English contributions.

Is there an overuse of narrative, personal views and jargon within the SWEBOK product? In addition, can a guide full of normative information from generally accepted [English] knowledge basis be of use to the industry as a whole?

SWEBOK provides a good deal of normative information - prescribing what an engineer should do in a specified situation rather than providing information that might be helpful. This normative literature is validated by consensus formed among practitioners and is concentrated in standards and related documents. Yet, stating that this information must be known by software engineers, is not the same as stating that this knowledge falls within the bounds of the software engineering discipline. Instead,

it must be stated that software engineers need to know some things taken from other disciplines - and that is not the approach adopted by this Guide.

SWEBOK states that generally accepted knowledge is the chief criterion for inclusion of material. However, the selection process for including topics does not account for what is achievable good practice and consequently leaves a lot to be desired. … It is our opinion that there is usually a gap [indeed, often an enormous gap] between actual [generally accepted] practice and what appears in textbooks.

Are the references cited within the guide a good source of knowledge for the real-world software industry, or are they not comprehensive enough and simply out of date reflections of traditional engineering processes?

A lot of knowledge has been developed in the software engineering field in the past 10 to 20 years, and professionals who haven't kept up with these developments probably have become out of date. In this respect, it was acknowledged that SWEBOK is an excellent starting point. However, this is a subjective opinion as the scope of the knowledge base is questioned because of incomplete references, mainly from the mid '90's. In addition, lack of non-English texts begs the question of how good is a reference guide that reinforces traditional engineering processes. In addition, the structure of the product [SWEBOK] made it difficult to find references and even when found they are generally from the mid-nineties. Whilst it was conceded that some texts will be dated (core text have to be dated), it was felt that these could have been backed up by more up to date texts.

Has the product been properly [globally] field tested; when it is clear that the current audience has more than a seventv five percent Western bias (50% from the USA; 25% from the European Union)?

It is clear that the western world [Americans] believe that they are the world leaders in software engineering, as from the opening pages of the document the scope of cultural, regional and industrial factors is called into question. Yet it would have been better to collaborate with others, from the start, in order to make it a truly global collaboration. In addition, SWEBOK is advertised as a global guide but is only based upon English Language texts. SWEBOK is yet another American World Series', but this time within the software engineering fraternity. (No one else is invited to the party as there is an English Language [only] guest list.)

There are numerous [financial] stakeholders formally represented within the SWEBOK project. How can an Industrial Advisory Board (IAB), composed of representatives from industry and professional societies that provide financial support for the project give an objective view of the project?

It is not yet clear whether SWEBOK will carry any formal authority, although this is clearly a central goal of the SWEBOK team as reflected in their literature as well as in their drive to include many organizations (including ACM) in the effort. At the same time, the group believes that the companies represented on the SWEBOK Industrial Advisory Board make substantial financial contributions to the SWEBOK effort. Although this may help to ensure that SWEBOK is properly funded, it almost certainly harms the potential authority that the product might otherwise hold. The credibility and authority of the individuals who participate in the effort will also have a significant effect on the credibility and authority of the product.

Conclusions [from group B's report]

The SWEBOK guide as a whole is an informative piece of work. It works well as a reference when starting work, but does not help toward any in depth work. Furthermore, the contents of the guide are viewed as an informed and reasonable characterisation of the software engineering body of knowledge, and as a baseline document for the Iron-man phase. As its stands at present it falls short of being a comprehensive [reference] guide.

Overall, it is clear that the SWEBOK effort is structurally unable to satisfy any substantial set of the requirements we identified for bodies of knowledge in software engineering, independent of its specific content. The issues surrounding software engineering bodies of knowledge and the entire discussion of "software engineering as a profession" are extremely subtle and complex. We believe very strongly that the fundamental goal of people in the software community is shared: there is a great and legitimate societal and economic need to improve our ability to effectively engineer software systems. It will continue to be produced so in that respect it does have a future.

The consensus of opinion formed amongst the group was that the product was an incredibly useful reference point. However, the document is an abject failure as a holistic guide to the software engineering approach. It is clear that what may have been found helpful by some may well be considered a hindrance to others."

5. Final Remarks

This paper has reported an initial attempt to use the SWEBOK guide to support learning and teaching at postgraduate level. The number of students following the module in this part-time mode was low so their remarks need to be treated with caution. However, these are mature students most of whom are employed in responsible positions and are motivated to carry on learning in their own time and in most cases at their own expense. Also, this exercise has provided feedback on SWEBOK from a group not included (to our knowledge) in the previous reviews – real students using the guide as an academic resource. Some of their comments are quite telling. It should be noted that the first author of this paper in providing an overview of SWEBOK was careful to ensure that his own concerns with the project were not raised with the students – they were left to draw their own conclusions.

We believe this has been a useful exercise and we will repeat it with the next part-time cohort of students. We also intend to encourage our colleagues who teach the module on the full-time masters programmes to do the same this coming academic year. This should be quite interesting as the number of students on the full-time module could exceed 100. Also, many of these students are from outside the United Kingdom and a sizeable majority do not have English as their first language.

6. Acknowledgements

We would like to thank the students who engaged so enthusiastically with the module and participated so well.

Group A: Michael McKie, Marie McCarty, Mina Barnes, David Conway, Peter Aynsley, Mark Sims and Phil Dodd

Group B: Nusrat Amir, Esther Dontoh, Karen Fletcher, Steve Frank, Chee Seong Lim, David Reed, Ian Simpson. and James Thwaites.

References

[1] Bagert D.J., ACM Withdraws from SWEcc, in Forum for Advancing Software engineering Education (FASE), Volume 10, Number 07, July 2000, available online at http://www.cs.ttu.edu/fase

[2] Bourque, P. et al, The Guide to the Software Engineering Body of Knowledge, IEEE Software, Vol. 16, No. 6, pp35-44, November/December, 1999

[3] Dupuis, R. et al, The Guide to the SWEBOK Project: A Status Report: in Forum for Advancing Software engineering Education (FASE), Volume 11, Number 09, September, 2001, available online at http://www.cs.ttu.edu/fase

[4] Ford, G. and Gibbs, N.E., A Mature Profession of Software Engineering, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-96-TR-004, 1996.

[5] Gotterbarn, D, How the New Software Engineering Code of Ethics Affects You, IEEE Software, Vol. 16, No. 6, pp58-64, November/December, 1999

[6] Hilburn, T. et al. ASoftware Engineering Body of Knowledge Version 1.0, ref CMU/SEI-99-TR-004, Available from Software Engineering Institute Web site at http://www.sei.cmu.edu

[7] Speed J.R. Software Engineering: An Examination of the Actions Taken by the Texas Board of Professional Engineers, October 12th 1998, Available from Texas Board of Professional Engineers Web site at http://www.main.org/pebody/soft.htm

[8] SWEBOK Project Web Site. Available at http://www.swebok.org

9. Thompson, J. B., Time to go back and think again, in Forum for Advancing Software engineering Education (FASE), Volume 10, Number 09 September 2000, available online at :http://www.cs.ttu.edu/fase

[10] Thompson J. B., Any Real Progress, or Is It Just Politics and Turf Wars, in Forum for Advancing Software engineering Education (FASE), Volume 11, Number 09, September, 2001, available online at: http://www.cs.ttu.edu/fase

[11] Thompson J. B. and Edwards, H. M, The Hybrid Manager: Achievement within a World-Wide Dimension. IFIP WG3.4 International Working Conference on Information Technology in Management and Business Education, Melbourne, July 1996, Proceedings edited by Ben-Zion Barta, Arthur Tatnall and Peter Juliff, published by Chapman & Hall, London 1997, ISBN 0-412-79960-X, " The Place of Information Technology in Management and Business Education", pp. 199-206.

[12] Thompson J. B. and Edwards, H. M, Providing New Graduate Opportunities in Software Engineering: Experiences with a UK Master's Level Conversion Course, Journal of Systems and Software 49 (1999) 135-143

[13] Thompson J. B. and Edwards, H. M, Ensuring that Ethical and Related Issues are Addressed in Two Masters Level Computing Courses: a Report on Developments and experiences Over a five Year Period, Australian Institute of Computer Ethics Conference AICEC99, 14-16 July 1999, Lilydale, Melbourne. pp 365-375.

APPENDIX: MODULE SYLLABUS.

CODE: COMM62	TITLE: Systems Engineering	LEVEL: M
CREDITS: 10	LEARNING TIME: 120 hrs	SUBJECTGROUP: Computing
PRE-REQUISITES:	Systems Development	CO-REQUISITES: None

AIM: To extend the expertise of students to enable them to participate in the engineering of software systems.

LEARNING OUTCOMES:
Knowledge:

1. Various systems engineering approaches
2. Critical awareness of selected tools and techniques which support software development
3. Good management techniques for the development of software

Abilities:

4. Can participate as an effective member of a team in the specification and production of quality products.
5. Can identify and apply appropriate standards, metrics and management techniques.

INDICATIVE CONTENT:
Engineering approach to software development, construction and maintenance. Effects of current practices. Approaches for software development (e.g. soft systems, systematic, RAD). Quality and quality assurance approaches. Management of the production process and support methods (e.g. PRINCE). Standards and metrics. Tool support for life cycle.

TEACHING, LEARNING AND ASSESSMENT:
The module will be taught using a mixture of lectorials, directed reading, summary and milestone sessions, tutorial/seminar sessions and self study sessions. Tutorials will be primarily for group based activities some of which will be machine based. Students will be expected to study selected papers, produce literature reports and publicise the results to their peers.

Learning Strategy:

Lectorials:	12h	Summary and milestone sessions:	12h
Tutorial /seminar sessions:	18h	Group study & assignment:	36h
Self study/Assignment work:	42h		

Assessment Balance: Time Constrained Tests: 40% Portfolio: 60%

KEY TEXT: Pressman, R S., "Software Engineering: A Practitioners Approach - European Edition", McGraw-Hill, 1997