Some Suggestions for the (Software) Engineering Degree Curriculum[1]

Felix Redmill
Redmill Consultancy
Email: Felix.Redmill@ncl.ac.uk

This paper is written in the light of my experience of software development in industry and lecturing to students of software engineering. The views expressed in it are therefore nominally aimed at software engineering education, but it is felt that they are appropriate to engineering curriculums in general.

Introduction

It is recognised that there is a mis-match between the attributes of new engineering graduates and the requirements of industry for engineers and junior managers. As a manager in industry I did not believe (and I still do not believe) that it is the universities' function to run a production line for engineers who meet industry's every need on their first day of employment. I considered that the universities should provide us with educated thinkers, the ability to think being the attribute which would best equip them to learn from their training and experiences in industry, so that, with the resulting wisdom, they would be of greater benefit to industry in the longer term. I accepted the responsibility to recognise the discrepancies and provide appropriate training of graduates to bridge the gaps.

However, while graduates could quickly be 'introduced to industry', and many of their deficiencies could be reduced or eradicated by learning, there were four recurrent shortcomings which were of particular concern. These were more ingrained than others and required a change in attitude for their alleviation. They were:

- Graduates were problem solvers, not problem definers or specifiers;
- They were individualists rather than team workers;
- They had no concept of a project as a co-operative enterprise, and therefore no understanding of, or of the need for, project management;
- They thought in terms of modules, not systems, and so were narrow thinkers and specialists rather than systems engineers.

As a manager, I accepted these shortcomings (among others). I did not examine in detail why they existed, but supposed that universities did their best and that what they did not cover was necessarily the responsibility of industry. However, in the light of having now lectured at a number of universities, and been a visiting lecturer for some time at two, I have reviewed the above issues and come to three conclusions.

First, the deficiencies mentioned above are not necessary results of a university education; graduates need not enter industry with them. They arise as the result of our method of teaching and do not need to be inculcated in the first place.

----

[1] Published in Engineering Science and Education Journal, Vol. 4 No. 3, IEE, London, June 1995

Second, the time has come for them to be addressed by the universities. I do not imply an attempt to define graduates as systems engineers, ready on emission from their learning institutions to be project managers and trouble-shooters. Rather that in today's industrial climate it is important that graduates should enter industry with an awareness of the principles of the four issues, through teaching and directed experience, planned and included in their curriculums.

Third, it is quite feasible for universities to include these issues in engineering curriculums, though in some cases it would require a change in university management style and culture to achieve it.

In addition, there are three further issues which I believe it is worth mentioning. They are increasing in importance, not only in engineering-based industry but also more generally. They are:

- Understanding human relationships and behaviour;
- Professionalism;
- Being able to speak a second European language.

In the following sections, I review each of the seven issues, discussing them and making suggestions for how they might be addressed in universities. It will be seen that the seven issues are inter-related.

Problem solvers, not problem definers

In industry, problems do not define themselves; someone has to define them. Determining a suitable solution is a process which may require a number of steps, including those of identifying various options, assessing their feasibility, choosing the most appropriate, and then planning its implementation.

Given the nature of the education system, it is understandable that none of these activities was familiar to graduates, or even recognised by them as being necessary. The assessment of engineering knowledge at university is based on the ability to solve defined problems. This makes both teaching and the setting and marking of examination questions easier, but it is contrived and does not prepare the budding engineers for what they will experience in the 'real world'. There (in industry) the greatest difficulty is usually in uncovering, understanding, defining and specifying the problem to be solved. This may involve identifying and investigating symptoms, and testing diagnostic hypotheses. Doing this often takes longer than to effect a solution.

I mentioned above that both teaching and assessment are facilitated by basing a syllabus on problem-solving techniques: 'Given this problem, this is how you go about solving it.' The system encourages this, being based on 'what is known' rather than on enquiry. Thus, a repeat to the teacher of what has been taught is proof of learning what is known (and, by implication, what is true).

Examination questions in the form of problems which can be solved by previously taught classical techniques have another advantage for the lecturer: they allow questions to be set so that answers are unambiguous and easy to mark. The solution of an equation, or the repetition of a list of important points, is quickly assessed as being right or wrong; a discursive exposition is not.

Classical problem-solving techniques offer the added convenience of not being threatening. For lecturers without real industrial experience, they provide a means of choosing a syllabus based on a carefully defined subset of the subject which can easily be understood, controlled and taught, and in which the chance of wide-ranging and difficult-to-answer questions from the students is minimised.

Yet, in industry, once a problem has been defined the solution is frequently (though not invariably) apparent. This is not to say that it is trivial or inexpensive; indeed, a feasibility study to determine the merits of different approaches may be necessary. But it suggests that a necessary and important, indeed a primary, element of an engineer's armoury is 'problem definition'. In the main, a problem has not only local effects but also wider-ranging effects due to its interaction with other components or items of equipment or with the environment (hence the need for a systems view, as discussed below).

My recommendation is that all engineering students, in their second and third years, should have Requirements Engineering modules in their curriculum (as well as learning classical techniques). In such courses, they would learn not only about requirements capture, analysis and specification, but about effectiveness - doing the right thing. The appropriate problem-solving technique offers efficiency, but it is worse than useless if the wrong problem is being solved.

It is well known that in the development of software systems there has been more re-work (and therefore waste) due to writing the wrong software than due to writing poor software. It is time that students were made aware of this at an early stage. Diagnosing a problem, capturing requirements from customers, and preparing a specification are not only essential aspects of engineering but also difficult. A curriculum based on solving pre-defined problems conceals this from the students.

Previously I had assumed that a course in Requirements Engineering might not be suitable for undergraduates with no industrial experience; that they would not easily apprehend the relevance of the abstract concepts. But such an assumption must have arisen out of a way of thinking which was itself informed by the problem-solving-based curriculum. During the last few years, I have been delivering a Requirements Engineering course to students at a university in Poland, and I have been astonished at how interested they have been in it and how readily they have assimilated and used the principles. Perhaps an important point has been that the course is delivered within the context of a team project (see the next section). In this, the students are provided with an outline requirement for a system, and they have to interview and negotiate with a real customer in order to determine and document the full requirements on the system. In the context of this task, the students quickly appreciate the importance of Requirements Engineering; they experience its difficulty and begin to apply the course content. In doing so, they come to recognise that the requirements they had assumed from the initial problem description are neither complete nor necessarily correct, and the system which they had conceived of in response to them would not satisfy the customer. They get their first whiff of 'software engineering', of 'requirements capture', of 'customer satisfaction', of 'quality' - and they do it for themselves.

Individualists, not team workers

My experience has been that recent graduates lack an appreciation of the concepts and importance of team-work and an inclination to work in teams. Most have not experienced team projects at their learning institutions. True, there has always been laboratory work in which students work in groups, but frequently this has been designed more to overcome the shortages of laboratory space and equipment than to promote cooperative relationships, communication and team-work.

Perhaps Western culture inculcates in everyone, graduate or not, a self-reliant attitude, a reluctance to share information, and a disinclination to work cooperatively. Yet, cooperative relationships, communication and team-work are some of the indicators of 'quality' in an organisation and are therefore of fundamental importance in industry and in life. All the more reason for guidance to be given towards a cooperative attitude at university rather than later.

My proposal is, first, to encourage a cooperative and enquiring attitude among students from the beginning. Then, to emphasise the point and aid the process, to include a team project in the second year of the engineering curriculum. This is not revolutionary; there are institutions already doing something along these lines, and some are successful. Many, however, do not take the administration of the project seriously enough. They allow themselves the excuse that the staff requirements and the planning and coordination necessary for the management of the project took them by surprise and is beyond their resources. Yet, these requirements are predictable. The fact that they were not predicted suggests a lack of management acumen at lecturer and professor level in learning institutions. And, remembering that one purpose of the project is to teach students the principles of project management, which includes planning, forecasting, estimating resource needs, and managing resources, the question of whether the students are receiving guidance from competent sources must arise.

To discuss fully the requirements and design of team projects is beyond the scope of this paper. It is perhaps sufficient to say that it requires proper planning and dedicated attention throughout. Moreover, it should not be an appendage to the curriculum but an integral part of it. It should be clear to the students that it is relevant to the remainder of the course, that the other subjects of the curriculum are relevant to it, and that it is representative of a real industrial project. It should not merely expose the students to the technical aspects of developing a system, but also promote the concept of cooperative team-work and an enquiring attitude in the budding engineers. The next section continues this theme.

No concept of project management

If, in general, graduates are individualists rather than team workers, it is not surprising that they should have a limited understanding of the scope of a project or of project management. This is not to say that they are ignorant of the fact that some projects need to be managed; rather it implies that they do not recognise the implications of project management, such as planning, coordination, administration, the creation of the project environment, and monitoring and reporting.

In software engineering, graduates typically think of carrying out a project as going into isolation and writing a program. The recent graduate sees his place in a project as a solo artist, both vocalizing and playing the accompaniment, rather than as a member of an orchestra whose contribution is only a part, but without which the performance would be incomplete.

In industry, the manager of recent graduates needs to understand this, be tolerant of it, and take time to help the graduates to understand the nature of industrial projects, appreciate their own place and importance in them, and learn that a good team does not simply use the contributions of individuals, but harmonises them.

As a visiting lecturer at two universities, one in the UK and one in Poland, I have recently been involved in group projects in Software Engineering which have demanded not only that the students work cooperatively in teams, but also that they recognise and discharge the responsibilities implicit in project management, complete the project according to plan, and account for the time and resources expended in doing so. In both cases, the students have enjoyed the experience, participated enthusiastically, and declared that they learned more in those projects than in the lecture courses (although in both cases the projects were designed to support and reinforce the lecture courses). Such projects are demanding of staff time, but they add a new dimension to engineering education.

In both universities I have lectured to the students involved in the projects on Project Management. As with Requirements Engineering (see the previous section), I had not anticipated that students without industrial experience would find this course interesting. Yet, perhaps because of the team project, and perhaps because I took care to use the project in examples so that students could see the relationship of the subject to the success of their projects, I found a level of keenness and attention which I could not have foreseen. The principles of Project Management were in some measure understood, their relevance was appreciated, and they were immediately applied.

But there is a problem. Project Management is too often seen as merely the application of techniques to overcome technical problems, with no account taken of human affairs such as leadership, team building, customer-developer interaction and other relationships. If it is taught by the wrong people who *know* that 'this is how it is done', it will be made into a manage-by-formula affair. Then, the means of achieving the ends proposed in this paper will be used for perpetuating the problem.

Parochial Thinkers, not Systems Engineers

In industry, there is frequently the complaint that there is a shortage of 'systems thinkers'. I have heard senior engineers desperately wondering where such people will come from. The answer should be, from the universities. Yet, I have not heard it suggested that graduates should be more systems-oriented. They are not, and it is accepted that things are as they are.

In their problem-solving-based education, students have their eyes lowered to the component level where the mathematics apply rather than to the system level where the effects of the system are felt - by industry, the public and the environment - and which forms the context for the component and design decisions.

When I took a degree in Electrical Engineering, I was green, having not been exposed to practical applications. But, although they were talking to 18 year olds, the lecturers offered no explanations of context, going directly to the solution of problems. I was taught about power factors without being told about power generation, and about stub matching and transmission equations without being told about telephony or what might be transmitted. I learned about the intricacies of gear wheels without having them placed in the context of a vehicle. It is true that the mathematical solutions do not either assume or require a knowledge of their contexts, but I came to recognise that the broader picture is necessary to a lasting understanding rather than a mere manipulation of variables for the purpose of satisfying the examiners.

In my more recent university experiences, I have not found an emphasis on systems, but rather on techniques, and even on tools.

So, what can be done to lead young engineers to 'think systems'? Creating an attitude is not something which can be achieved by a single action, as has been discovered by managers who have seriously attempted to initiate a quality culture. Systems thinking would need to be inculcated over the duration of the degree course - just as 'problem-solving thinking' is now. Yet, a significant change could be achieved if lecturers made more of an effort to devote some lectures to describing 'how things work' and providing the context for the specific issues of the subsequent lectures. I believe that there is a need for lecturers to 'put themselves in their students' shoes' and always provide contextual information so that the manner of 'system operation' is clear before getting down to problem solving at the subsystem or component level. And why would they do this? - after all, it places an onus on lecturers who themselves may well be no more than problem solvers. They must do it if it is certain that the examiners will set questions on contextual matters. And if the lecturers themselves are the examiners, why would they bother then? They would if their professors were more assertive in defining their requirements on teaching methods. But this requires a fundamental change in the philosophy of teaching and examining as well as of management at universities. How the universities would need to go about making these improvements is beyond the scope of this paper, but it may be useful to suggest that departments take a more strategic view of courses and coordinate subjects and syllabi so as to achieve defined goals.

Then, if students are to develop a systems rather than merely a problem-solving view, they need to be exposed to books which encourage this. One which comes to mind is Donald Norman's treatise on design, *The Psychology of Everyday Things* (Norman 1988). Two others are E F Schumacher's *Small is Beautiful* (Schumacher 1974) and *Good Work* (Schumacher 1980) which, though written by an economist, make many points on the appropriateness of design, technology and 'common sense', and emphasise the importance of context to the value of a system. These books remind their readers that systems are designed to be used and that the designers therefore should consider both the users and the context of use - something of which students do not currently seem to be made aware. Their authors speak from the point of view not of mere technique but of wisdom, and it is time that students were exposed to wisdom as well as to technique.

But examination questions set on such subjects are not so readily assessed, and engineering lecturers and examiners are deterred by this. Yet system-level questions are commonplace in other disciplines, such as Psychology and Sociology. And, as we know ourselves to be superior to these quasi scientists, we must also know that if they can do it, so can we.

Understanding human relationships and behaviour

Some years ago, a conversation with a barrister friend set me thinking. I mentioned my interest in psychology and he said, 'Well, I've had to learn a lot about psychology. I only wish I had had a course in it at university.'

He went on to point out that as a young lawyer his first tasks had been those which he was least well equipped to tackle. Had he been set to preparing briefs for criminal proceedings, he would have had no trouble; that would have called on the technical knowledge of his training. But his first jobs were debt collecting and divorce. These called for sensitive interaction with often emotionally charged individuals; and emotion was something with which he had not been trained to cope. In retrospect, he felt that many divorce litigants would have preferred amicable settlements, but all he could offer was to 'screw the other party' for all he could get.

This conversation led me to ponder the average medical General Practitioner's apparent insensitivity and inability to listen. A GP friend confirmed, astonishingly, that there were then no psychology courses in the medical curriculum - though I understand that this has now changed.

But what does this have to do with us? Well, young engineering graduates are not removed from human relationships - and frequently they are gauche. If problem definition is important, then so must be an understanding of human interactions, for systems analysis is largely a human-interaction affair, with profound psychological implications. Information does not reliably leap the gap between users and the analyst unless they are in a satisfactory relationship; recognising the correct ('real') requirements can depend on an astute psychological awareness. Similarly, young graduates do not find it easy to cope with operational and maintenance staff when sent to investigate operational problems; and the staff, in turn, are not sympathetic to brash newcomers. The result is great difficulty in discovering, let alone analysing, the symptoms of the problems.

Project management, team leadership, and, indeed, all management, comes down to 'people issues'. So does dealing with customers. It should not be expected that a graduate is immediately proficient in these matters - which many 'managers' do not master in a lifetime. But their brash confidence could usefully be tempered by an awareness that success depends not merely on knowledge, or competence, but also on achieving harmonious relationships with others, that the 'others' are sensitive humans like themselves, and that achieving relationships is a delicate business.

So what is recommended? A 'human behaviour' course for engineering students. From my discussions with psychology lecturers and students, I suspect that engineering would not be well served by the direct transfer of courses from a psychology department. It would be more appropriate and, I think, not too difficult, to design a course for the purpose which would place behavioural (psychological and sociological) principles in an engineering context. The student should be trained in being alert to the sensitivity and reactions of others, and to the importance of human relations in management, teams, and the achievement of quality. Perhaps Berne's Transactional Analysis 'Parent-Adult-Child' model (found in Berne's interesting, easy to read, and even amusing books, such as *Games People Play* (Berne 1968) and *What Do You Say After You Say Hello?* (Berne 1975), as well as in his earlier papers) would be a suitable basis for such training.

We should not expect to make sages of students, but promoting self awareness and an awareness of others would be a help to them and to industry. Moreover, I believe that the need for this awareness is growing. Whereas in the now receding past respect was taken for granted, it cannot now be. Society is more selfish, the inclination to 'put yourself in someone else's shoes' has diminished, and an urgency to get 'what I want', even to the detriment of others, is more pervasive. Such attitudes are not conducive to good relationships. If we want harmony in industry, we need to inculcate it.

Professionalism

At a recent forum of experts, the conclusion was reached that the greatest cause of accidents is a lack of professionalism: people not acting in accordance with their responsibilities, not recognising their limitations or knowing when to seek assistance, and making unsound judgements.

At the same time, liability laws are homing in not only on organisations but also on individuals. And the individuals are not ready for them.

It would not be difficult for a module on professionalism to be included in the curriculum. It could be based on the codes of professional practice produced for the guidance of engineers by the Engineering Council, for example in *Engineers and Risk Issues* (EC 1992) and *Engineers and the Environment* (EC 1993). Both of these emphasise the need for engineers to be aware of their conduct, to understand the legal framework within which they work, to encourage a public understanding of the issues in hand, to keep up to date by education and training, and to be aware of their various other responsibilities.

There are also booklets on professional conduct produced by all the engineering institutions. Currently, it is typical for graduates never to have come across any of these publications and never to have been invited to consider ethical issues within the field of engineering.

I do not suggest that students should be made to behave as though they were in industry. Savouring the university environment is too valuable an experience to be usurped by such a premature, artificial and unnecessary step. But if, as is proposed throughout this paper, students need to develop understanding as well as knowledge, an understanding of what it takes to be professional should be included. And example goes a long way to developing such an understanding.

But to teach such a subject for one hour each week for a term is not enough. Lecturers see themselves as professionals. Yet they do not act as professionals in industry are expected to do. In one institution, when I mentioned that I needed to get to the class so as to be on time for a lecture, it was suggested that I need not hurry as the students never arrived on time. I found this to be true. Yet, when the students found my lectures to commence on time, latecomers became exceptional.

Lecturers are the role models for potential Professionals who will be professional or not according to the example set them.

Ability to speak a second European language

It is time that every professional engineer had the ability to speak a second European language. More and more, our engineers and managers have meetings with their Continental European counterparts, and on almost every occasion they find themselves at a disadvantage and embarrassed because of their inability to understand or converse in the language of their hosts. I know that the use of a language laboratory is an option open to many students. But this is insufficient. A European language should be a compulsory part of every engineering degree course and should be taught throughout each year of the course. It should be examined, and examination success should be a necessary prerequisite to passage to each successive year and to the award of a degree.

The reasons why not

These proposals will no doubt encounter detractors on the grounds that engineering syllabuses are already too full, the contact hours of engineering students are higher than those of most, if not all, others, and that it would therefore be impossible to introduce them into the curriculum. But that would be to miss the point. The issues I raise are of culture and attitude, and they call for a reappraisal of the curriculum in order to define and introduce a new philosophy of engineering education. Then it would not be a matter of trying to find time to add modules to an already overcrowded curriculum, but to determine what is important both in terms of knowledge to be acquired and culture to be developed.

Perhaps the Institutions responsible for accrediting engineering courses would also consider this point. It is said that universities dare not alter courses without their approval and, as they are persistent in demanding the inclusion of more and more material while seldom sanctioning withdrawals from the curriculum, it would be impossible to make the proposed changes. Perhaps the Institutions too might reappraise their strategic goals for undergraduate engineering education.

The strategic context is as important to education as it is to industry. I regret that the scope of this paper does not allow me to lay out suggestions for strategic change at universities.

Summary

While universities should not be expected to match graduates to every need of industry, there are certain attitudes inculcated in graduates by the education system which are not merely shortcomings but are detrimental both to them as future professional engineers and to industry. It is the thesis of this paper that changes could and should be made so as to cease inculcating these attitudes.

I have identified seven issues, and for each I have explained the problem and made proposals. Adopting the proposals would require changes at universities, including improvements in strategic planing, management, professionalism, and planning courses on the basis of what we want our graduates to be rather than merely on what we want them to know. While acknowledging that some of the points made have already been recognised to some extent in some teaching institutions, I know of no systematic attempt to examine the philosophy or strategy of educating engineering students. It is perhaps time for such an examination.

Acknowledgment

Thanks to Neil Storey and Pat Hall for their helpful comments.

References

(Berne 1968) Berne E: *Games People Play*. Penguin, 1968.
(Berne 1975) Berne E: *What Do You Say After You Say Hello?* Corgi, 1975.
(EC 1992) *Engineers and Risk Issues*. The Engineering Council, 1992.
(EC 1993) *Engineers and the Environment*. The Engineering Council, 1993.
(Norman 1988) Norman D A: *The Psychology of Everyday Things*. Basic Books Inc., New York, 1988.
(Schumacher 1974) Schumacher E F: *Small Is Beautiful*. Abacus Books, 1974.
(Schumacher 1980) Schumacher E F: *Good Work*. Abacus Books, 1980.