Computing/IT as a subject for 16-19 year olds

What should we be teaching? Why? How? What are the barriers to success? There is a lot of debate at the moment about the content of IT/ICT/IS/Computing for 16 -19 year olds.

This BCS Thought Leadership Debate was held on 24 April 2006 at the Royal Society with speakers Professor Jon Crowcroft, University of Cambridge and Andy Puttock of BT who is one of the employers on the e-skills UK Diploma Development Partnership, starting the event.

Opening comments

Just what should children be learning about IT till the age of 19?

Some background will help give perspective on this question.

Computing science and IT jobs and their supporting school and higher education (HE) courses grew exponentially until 2000 so much so that today the UK has over 100 university departments with more than 150 students each. There are ever increasing numbers of jobs for programmers, skilled IT people, and core-computer scientists.

Even during the dotcom crash the number of jobs went on going up. But now school and HE numbers in IT are falling,

But which people are leaving? More perhaps from computer science than IT, but core well rounded computer science people are probably a small minority of the students anyway.

In the last 2-3 years, the problem has been increasing. It is widespread in Europe and North America (barring some exceptions like Poland and Italy). Gender imbalance has got worse and is heading towards a fiasco (exception is Italy, again). School teachers, university admissions staff and employers are all puzzled.

Part of the cause of this might be the evolution of what is a relatively new subject over the last 30 odd years, which is three generations of staff and students.

Back in the 1970s, the first people to work in computer science had crossed over from physics, maths, statistics, languages, music and other subjects. By some definition, these were unusual - creative people capable of change. They had no formal training in computer science because there wasn't any, but learnt the subject as they created it.

In the 1980s, more and more people joined from a younger age group: youngsters who learnt from owning ZX Spectrums, BBC Micros or the like and hobbyists who solved computing science problems to make their machines more usable and more fun.

By the 1990s, teachers who had been some of those youngsters of the 1980s were starting to come through.

In the 21st century, we do not have these people, as there is no more crossover, nor are there hobbyists who have to solve problems to make their computer useful; the computer has become for most a commodity device for playing games, shopping, word processing, creating spreadsheets and enjoying music and video.

How can we define computer science? It is the systematic discipline of problem solving. We could call it the third place (to paraphrase Sony's advert) not quite a science, and not quite engineering (or engineering science), nor is it a branch of anything else such as mathematics.

This is clear when you look at aptitude and crossover: mathematics is not an indicator that someone will be good at computer science and engineers typically do not make good programmers. There's something different about it, perhaps a flavour of music (also associated with mathematical ability) and of languages, but something else too. Some people call it the science of the artificial, but whatever it is, it's here, it's big, and there's a lot of demand for people who are good at it. IT continues to grow and advance at a relentless pace unseen in any other discipline.

Why is it important in the UK? Some examples might clarify: The UK has Vodafone, Symbian and ARM, which are the biggest cell phone company in the world, a company that licenses the operating system in more cell phones than there are PCs in the world and a company that licenses a chip design used in more devices than Intel sell.

The UK also generates around 50 per cent of software for new media (principally, games software, but also CGI and web). We also contribute significantly in science, aerospace engineering software, medical software, and, of course, financial services software.

This is a significant contribution to the UK economy, and people that work in these industries say they really enjoy their work!

We can no longer pretend there is room for 100 computer science departments at university level. Plenty of this material can be used in support of things like eScience, eBizz, new media, and so on, all of which also needing strong programming (problem solving) skills. Only core computer science should be in universities - perhaps 20 departments.

One possible solution

It is IT's ubiquity and subsequent invisibility that is a barrier to children's interest. Microsoft, BT, Ford and other blue chip organizations have been working on a possible solution to these issues.

When children become graduates they need to be fit for work and this means that employers need to be involved early. In the February 2005 the government White Paper proposing a vision for children in the 14-19 year old bracket – a vision that includes 14 new diplomas, with IT as one of the four to be introduced from 2008. Other subjects included are media, health, engineering and construction. Once achieved these diplomas would be better than 3 'A' grade A-levels. They are designed to be inspiring, motivating and to be the preferred route to employment or University, offering a high degree of confidence in a wide range of abilities – as close to a guarantee of achievement and quality of as possible. e-Skills are responsible for the development of the IT diploma with the aim of motivating 14-19 year olds to get

involved in IT and also to ensure continuing professional development (CPD) is embedded in the programme.

For this to work it needs inclusive and wide-ranging consultation and in 2005 eSkills formed the Diploma Development Partnership (DDP) with employers (via the Employers Steering Group), schools, HE establishments the QCA and other relevant bodies. A survey of 600 employers in December 2005 highlighted the need for a balance of technical, business, functional and employability skills and the employers emphasized the need to maintain a high level of core skills in areas such as English and maths and the employability area – key features that should be embedded included real-world skills in such things as work experience and project management.

Whilst this is a great opportunity there are a number of challenges to delivery to the required level. For instance one key issue is the logistical problem of helping thousands of schoolchildren to get hands-on experience. There would be health and safety implications, legal implications and the like that would need to be addressed. Virtual work experience could provide the answer.

Is this diploma the solution? It can't be the only one. Some graduates actually require remedial education in grammar, maths, even IT skills. We need to get them to see beyond the surface of their iPods, mobile phones and PDAs and see them for what they are: marvels of technology, a fantastic treasure, opportunities to learn.

The debate

What should be taught?
Problem solving skills and an approach that encourages questioning are vital. It would seem to be better to do maths and physics at this age as they are taught with greater rigour than current IT courses. Current computing courses at A level need more maths and students in general would benefit from more on engineering and communications skills.

Areas such as computational thinking and algorithms should be taught and these can be related to the real world. For example, something as simple as the behaviour of traffic queues can teach lessons in these areas.

Innovation is required. IT is central to economy of UK but students are not generally interested in business applications – they see them as boring. For example, on one course students are actually taught to do a mail merge but this is not a useful skill, it's just a thing that technology happens to do. If they are taught well about the principles of IT they should be able to work out how to do these sorts of things anyway.

Above all, the subject should be made exciting – because it is.

How should it be taught?
An integrated curriculum would help students make better sense of things. Learning different subjects together in a cross-curriculum manner would help them make connections and put them at the nexus of subjects - where exciting things often happen. This is difficult to do, but continuing professional development for teachers would help.

This sort of approach would require more innovative teaching approaches so that children are genuinely enthused. UK PLC already has many IT literate consumers, but on the development side teaching seems to lag. An important consideration is ensuring that that gender or ethnicity does not marginalize certain students.

Any solution should include also continuing professional development (CPD) for teachers not only delivered by educationalists but industry. Virtual learning environments can be involved too, providing a rich curriculum that can also be populated by teachers, HE people and industry representatives.

It seems that some good practice at primary school level is not being fed into secondary level. What would happen if we freed up some curriculum space to develop innovative classroom approaches? Interestingly the DfES 'Power to Innovate' scheme has many requests from schools to allow them to be more innovative when in fact they don't need to ask permission – could it be that too many schools are bogged down by their own strategies?

Good communication is clearly required between government, schools and HE establishments. This implies significant cultural change, but that in itself is a concept understood in different ways by different people. Billions of pounds have been pumped into schools, but to achieve some of this we need to get the ear the appropriate government ministers.

Another way to engage enthusiasm may be to get students hands-on use of games, music software and technology, web and social software and the like because they already relate to this.

A partnership is needed between teachers, employers, HE and parents and we need to get these messages to all those groups to get success.

Two examples of encounters with schoolchildren demonstrate the positive ways that IT can be used and presented.

- A project set in a number of geography classes in primary schools in London involved the following work by 8 year-olds. They would go out to the nearest high street and measure a set of shop fronts and junctions and so on. Back at school they programmed (in logo) a turtle graphics system to draw a scale model of the high street on paper via a small robot with a pen. They then decorate the scale map with a painting of the high street. This is art, science, geometry, programming, technology and geography all taught in one fun package.

- A 6th form assembly is given a 30 minute talk about the challenge of ubiquitous computing: Cars will have 500 processors and sensors, homes will have hundreds of computers. You won't just have a phone, you will have wearable devices checking your health and location, telling you about offers in shops and warning you about weather or traffic. At the end, they are given a chance to ask questions. Every time this talk is given they come up with new questions, but they seize on consequences such as: What about the power consumption of all these computers - isn't that a problem for global warming? What about the radio for all the networks - isn't that a problem for health? What about the problem of big brother and privacy? What about the 'have nots', people that can't afford to be part of this hi-tech world? Without prompting, they have collectively put their fingers reliably on a set of key challenges in the research community. Yet few of these youngsters consider IT
  or computer science as a potential A-level, university topic or career.

We need to redefine computer science as the problem solving discipline that it is in school from 9 years-old upward. We need to embed IT in other topics completely (as is happening) and stop teaching it as if it were anything different from reading, writing and arithmetic (or technical drawing, design and technology, metalwork or whatever).

We need GCSE and AS level materials that teach the background to problem solving as a discipline and require programming as a significant fraction - but we should resist the political pressure to have lots of people doing this as it isn't needed for UK PLC anymore.

We also need a core of well rounded computer science people who will carry the subject itself forward. An A level based on what is currently mainly remedial first year CS degree material, with lots of real world inspired programming project work, will help. When established, such an A Level should be required by computer science degrees, but not for IT-related degrees.

A CPD program for people in schools who have self-learnt IT should be put in place, and longer term CPD for newer cohorts coming in that have IT skills from day one, but need to keep up to date with the pace of change.

Why should it be taught?
We face the paradox of being in an industry that is though of as being horrible, but nothing happens in 21st century Britain without it. This view needs to be addressed at school.

We need well-rounded students with a larger range of skills. We want to teach the ability to use technology and to adapt to it, not just about a few proprietary pieces of software.

What are the barriers to success?
A lack of a clear career structure is a major problem. In the medical profession there are a number of career structures catering to different paths, but not in IT. At school students don't learn 'nursing' or 'accountancy' but get the basics and then move into clear career structures and learning paths later.

Another barrier to success is the perception of IT people, as perpetuated by the media such as in the TV programme 'The IT Crowd'. This has led to a lack of respect for the subject.

Likewise the motivation of teachers may be lagging because it is so difficult to please employers with the many requirements they try to impose on those coming to them. Employers should be concerned with getting thinking, motivated, enthused and creative students, rather than focusing on a few skills that could easily be taught on the job.

Teachers face many difficult circumstances – economic and technological – and they have intellectual and societal pressures. Classrooms can often be affected by local concerns rather than the national context.

IT is ubiquitous, but it is not universal in terms of access or children's experience of it in schools.

There is a conflict in how teachers take advice – things viewed as key skills may conflict with the problem solving and creative approach being extolled here. The

curriculum itself is interpreted differently by teachers – is it subject-based? should things be done collaboratively in the classroom? is it projected-based? And so on.

IT in education is often used to support and overcome the boring things – to remove boring and repetitive jobs, rather than be an outlet for creativity.

We need to separate IT skills and computing. The skills should be embedded and seen as a tool rather than a subject. Experience suggests that IT is seen as a user tool rather than a maker tool to solve problems or be creative with.

Conclusion

As in IT in the health service and other domains, we can fix this. There are great potential benefits to society and to many individuals on either side of the teaching and learning experience.

IT is a topic that employees and children really enjoy - it is ludicrous that one of the leading countries in the world in this discipline has developed a reality gap between primary school and later life through a set of accidents of history (and no-one's fault).

Getting this right is vital for the UK economy, the education system and, most importantly, the students themselves. Wouldn't it be good if children started to have Arthur C Clarke's view: IT as applied philosophy and practical magick?