

# A Second Opinion on the Current State of Affairs in Computer Science Education – An Australian Perspective

*V. Lakshmi Narasimhan*

*School of Electrical Engineering and Computer Science  
The University of Newcastle, NSW, Australia*

[lakshmi.narasimhan@newcastle.edu.au](mailto:lakshmi.narasimhan@newcastle.edu.au)

## Abstract

This paper presents a critical look at the likely demise of Computer Science (CS) as a discipline, in the light of various mishaps that academics in Universities have met with due to trial and error. We trace the issues of the past and present and, identify the reasons why computer science as a discipline in many Universities has not moved with time – in particular why it is not kept up with the pace required by the field. The loss of computer science is also well-reflected in the huge downturn in CS student enrolment across the Anglo-Saxon world. The paper also details some steps on how to change the ways and means of CS academics. We record that some of the arguments and recommendations presented in this paper can be controversial and may not necessarily relate to scenarios that exist in other countries. Further, some of the assertions made are without comprehensive statistical data, as we could not gather them, besides gathering anecdotal evidences.

**Keywords:** Academic morality, Relationship management in CS, Multi-disciplinary approach, Change management and Downturn in CS education

## Introduction

Many Universities have reported considerable decline in the enrolment of Computer Science (CS) majors around the Anglo-Saxon world and, many Australian Universities are also no exception to this down-turn (see Figures 1-3 adapted from Higher Education Research Institute, n.d.). Briefly, the following observations can be made: i) student uptake of Computer Science has declined (Figure 1) by over 60% between 2000-2004 and is now 70% lower than its peak in the early 1980, ii) interest in CS among women fell 80% between 1998-2004 (Figure 2) and 93% since its peak in 1982 and iii) the degree of production of CS female graduates is dwindling fast (Figure 3). However, some Universities have managed to fill-up their intake and some even managed to increase their student numbers. The following questions therefore beget the observer:

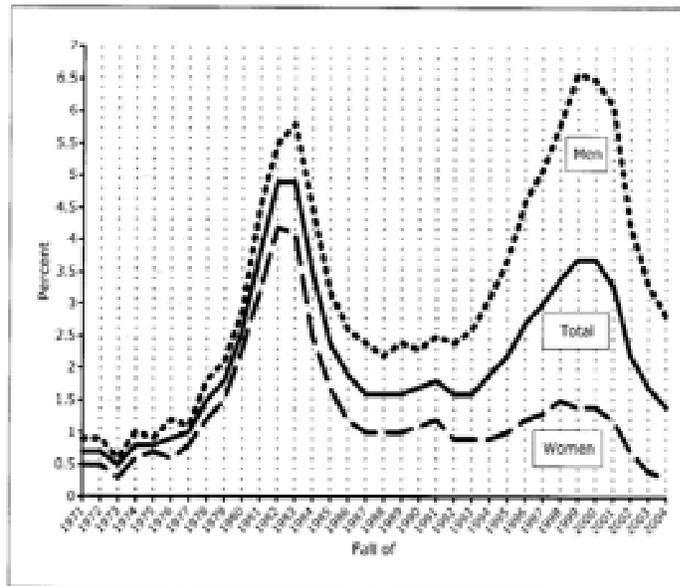
---

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact [Publisher@InformingScience.org](mailto:Publisher@InformingScience.org) to request redistribution permission.

1. Why did the downturn happen in CS education?
2. What are the real contributing factors to the downturn in CS education?
3. What actions could CS educators taken to prevent the falling enrolment?

## Second Opinion on the Current State of Affairs

Figure 1. Computer Science Listed as Probable Major Among Incoming Freshmen  
Source: HERI at UCLA



4. Why did the CS educators not take the required actions?
5. Why are some Universities able to attract full enrolment or even over-enrolment despite the downturn?
6. What lessons can be learnt from this downfall?
7. What are the real underlying issues?
8. Why are that comparative perspectives on CS education in the non-Anglo-Saxon world?

Figure 2. Probable Majors Indicated by Incoming Freshmen  
Source: HERI/UCLA

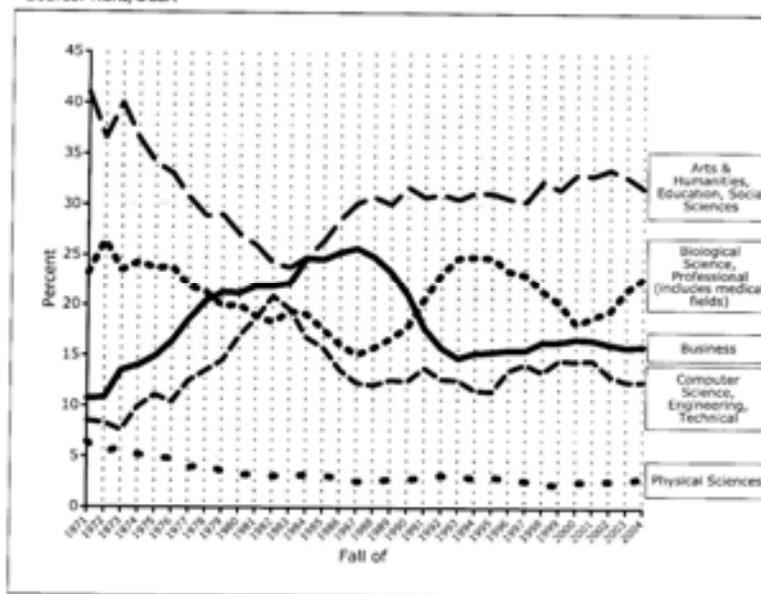
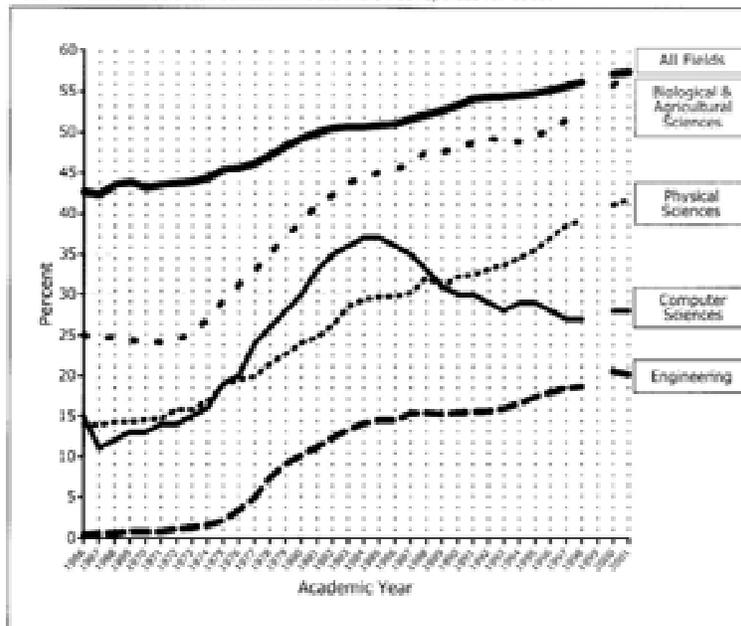


Figure 3. Portion of Bachelor's Degrees Granted to Women  
 Source: National Science Foundation. Data were not reported for 1999.



This paper addresses some of these questions and provides some pointers for introspection. The rest of the paper is organised as follows: the next section provides the context and perspectives on Computer Science, followed by the analysis of the list of issues raised in section 3. We discuss the lessons learnt in section 4 and give some pointers for further work in section 5. The conclusion summarises the paper and raises further investigative ideas.

As a special note, we record here that some of the arguments and recommendations presented in this paper can be controversial and may not necessarily relate to scenarios that exist in other countries. In addition, some of the assertions made are without comprehensive statistical data, as we could not gather them; however, the assertions and conjectures we make do have anecdotal, verbal and document-based support, some of which cannot be identified due to privacy reasons.

## Perspectives on Computer Science

It is well accepted that computer science has its foundations in Logic, Mathematics, Electrical Engineering and Linguistics and according to Wikipedia (n.d.), its ideals are about “*Finding eternal truths about problems and algorithms for posterity*”. Its emphasis is on analysing various problems, finding out their complexities, correctness and algorithms that yield the best solution. When the problems are solved, they do not change in a major way, except for “*changes around the frills*”. Thus for example, the solution to the famous matrix multiplication problem has been found by many people with different degrees of complexities for different applications and their underlying contexts (e.g., sequential, parallel, etc). Formal theory, most often based on (computational) logic, have also been developed to understand and get a handle on problems in computer programming language, natural language, database structures and artificial intelligence.

Most Computer Science departments have their origins traceable to schools belonging to either Mathematics (most often) or Electrical Engineering (mostly in technology dominated Universities). In general, mathematics-originated CS departments (which we will call CSM departments)

tend to be strong in theoretical aspects, while technology background CS departments (which we will call CST department) tend to be strong in practical aspects.

Computer science finds its niche by finding association with other disciplines and fields. Indeed Edsger Dijkstra, renowned computer scientist, said, “*Computer science is no more about computers than astronomy is about telescopes*”. It is worth noting in this context there are Universities in which CS departments trace their origins to such diverse disciplines as medicine, fashion design, business, finance, mechanical engineering, library science, pure artistic design and pure linguistics. Such CS departments (which we will call CSO departments) tend to have a finer focus compared to other CS departments and have been generally successful in maintaining their focus. The CSOs have indeed maintained their links with their origins and have taken considerable efforts to diverse their efforts in teaching, research and application focus.

We take the proposed issues in this context in the next section and provide a second opinion on why CSM and CST departments have not fared well in relation to the handling of student education.

## **Analysis of Issues**

### ***Why did the Downturn Happen in CS Education?***

Many educators in CS have attributed the downturn in student numbers in CS departments as something to do with the downturn in IT jobs available on the market. The negative publicity has somehow permeated to the level of high school leavers and this has led to the consequent reduction in student numbers at the Universities. The Dotcom burst added to this vows considerably and somehow the charm that kept CS students numbers on the upward profile for a decade or so seem to have evaporated. The potential to make “*a quick buck*” also appeared to have been lost in this process and this has led some parents to think that CS is no longer a commercially good proposition for their progenies. Added to these, is the fact that outsourcing of IT jobs to third world countries such as India and China have contributed significantly to the job losses and the consequent loss of interest in IT education. It is noted that these perceptions – to a degree - have their origins to academics, who belong to CSM and CST departments and other vested interests. If enrolments stabilise at the current level, it will still be 40% below that of two years ago (Wilson, n.d.).

### ***What are the Real Contributing Factors to the Downturn in CS Education?***

We contend that none of the reasons quoted in the previous section, namely, Dotcom burst, loss of jobs, job migrations, etc, are tenable towards justifying such a profound loss in student numbers in CS Departments in Australia and the Anglo-Saxon world. We believe that the real contributing factors are as follows.

### ***Profound Lack of Good Mathematics Education at High School Level***

It is well recognised that school leavers in Australia (in particular) do not necessarily have sufficient mathematics background that justifies their entry into CS. We recognise this as a serious problem, particularly since mathematics is somehow not considered/marketed as a “*cool and interesting subject*” at the school level. The lack of good mathematics teachers and the dilution of the rigour required for learning the subject are two other reasons for poor mathematical back-

ground for high school leavers. Universities seem to do an *ad hoc* job of patching the skill set requirement with bridge courses, which are once again inadequate.

### ***Untrained CS Educators at the University Level***

A number of CS educators at the University level have obtained their terminal degrees from the University at which they are currently teaching. While a degree of “*incest*” may be good in that the person may already be part of a research group with “*a known culture of operation*”, this could be detrimental for the broad development of the CS discipline in the University. This also impedes cross-fertilisation of ideas and thoughts. Anecdotal evidences suggest that a number of Universities in Australia have this problem. In contrast leading Universities in the USA (e.g., MIT, UC Berkeley) have policies for appointing internal Phds into their faculty stream. Usually they expect at least a period of activity being spent outside of their *alma mater* after completion of their Phds, before they could be considered for faculty positions.

Other serious issues include, limited amount of formal training in current CS related educational issues *per se*. Indeed most of the CS educators trace their background to (pure) mathematics and quite a number of them still continue to insist that CS is an extension of mathematics. Further, some of these academics do not realise that the field of CS has its origins in fields other than mathematics, i.e., in linguistics, logic and engineering. These issues coupled with the fact that most academics have little or no industrial experience in IT related areas, lead to poorly trained CS educators at the University level.

### ***Unprepared CS Educators at the University Level***

CS educators at many Universities in general have little industrial background or training (This has been observed by borrowing through a number of websites for academic resumes). A large majority of academics seem to have obtained their Phds straight after a series of course-based degrees (BS and MS typically) and with little IT industrial exposure, have joined academia to pursue typically academic research and teaching with little relevance to industry. Their PhDs are in advanced topics, which have little relevance to common teaching materials required to be addressed at the undergraduate level. Indeed at one University in Australia most of the CS academics have background in graph theory – not even in its applications. These CS educators, while they do teach well to CS students, do not do justice to the field in that they have little passion in teaching the materials; instead they teach out of necessity. Indeed in most Universities around Australia, very few academics spend their sabbatical leave at IT industries and further, spending quality time with industry is considered “*less than optimal*” by such academics too.

### ***Inappropriate Marketing of CS Education to High School Leavers***

CS courses are being marketed inappropriately at the high school level. Most of the high school leavers decide their career/course options by the time they reach their year 10 level. Unfortunately, Universities tend to market their courses at the year 12 level and consequently, with major failures. Anecdotal evidences suggest that Universities also do not articulate the differences between their various course offerings clearly to prospective students. Indeed a number of Universities have more than one CS (or related) Departments in various Faculties, which tend to compete with each other, rather than co-operate -- all to the detriment their very existence.

### ***Fears and Dreams of High School Leavers have not been Studied well in the Context of CS Education***

Most students who choose CS have “*potential for better job opportunities*” as the major reason for their choice. With the downturn in the IT industry, the fears and dreams of high school leavers

have not been studied and understood well. Indeed if only the educators address the very concerns of prospective students, their field would considerably improve in terms of industrial relevance and student numbers too (Arora & Chazelle, 2005). Unfortunately, a number of CS academics appear to be really out of touch with market forces that drive the field.

### ***Significant Semantic Gap between CS Education Offered by the University and that Required by Industry***

As noted before, there exists a sharp and significant semantic gap between what are taught at the University and what are required by industry. Several industries have taken the task of re-training the recruited employees to fit into the industrial mould. On the other hand, there appears to be some value in the popular comment offered by academics that industries want only immediate technology knowledge, while Universities teach the theory that underpins not only a particular technology behind a product, but also whole raft of technologies that are still being attempted. However, if one looks at the market penetration of Windows, it is around 94%, but very few institutions offer Windows as the basis for teaching a course on operating systems!

### ***Lack of Exposure of CS Students to Industrial Practices and Procedures***

In general, CS educators have provided minimal insight to students on industrial practices and procedures, while software engineering students do get some exposure on these issues in the recent curriculum. While some would dispute this, the overwhelming evidence in industry indicates that CS students do need considerable re-training in the industry, prior to joining a major project. This could also be due to the fact that CS educators – a considerable number of them – come from mathematics background with little knowledge or respect for best-case practices and procedures.

During the Dotcom peak, the attrition rate in the IT industry was around 15%-27% per year, depending on the sub-field. Notwithstanding this, the attrition rate at SAS Institute (which produces such quality statistical packages like SAS) had an attrition rate of less than 1% per year and they attributed this low attrition rate to the profound efforts of their HRD (Human Resources Department (Harvard Business Review, 1996)). Indeed a typical successful employee has to undergo nearly 1.5 years of training, before s/he can be placed in a team that manages a project. During the training period, the person is “*unproductive*” to the company. The HRD places considerable emphasis on moulding the person and “*fitting*” the person into a team. They thus create champion teams that work on project. The old adage, “*A champion team will beat a team of champions*” holds good for SAS Institute and success follows them.

### ***Lack of Recognition of CS Educators***

Indeed in the current “*publish or perish*” environment, there is very little recognition for CS educators’ plight. The field changes by the day and its application potential (to other fields) increases by the day, but there is very little time given to the academics to explore the open world; they are instead forced to teach unduly long hours on worthless courses. Further, the publication rates and their nature (e.g., journal, conference, etc) of typical CS academics are often compared with engineers<sup>1</sup>. The facts in the CS field, which are conveniently ignored, include: i) conference publications have more relevance to current research, ii) journal publications are mainly for archival purposes only and iii) by the time journal publication comes in print, it has little relevance to current

---

<sup>1</sup> This comparison is worse for Software Engineering academics.

research. The publication rates and their citation rates of various sub-fields in CS are also vastly different. Typically graph theorists tend to publish a number of papers, also with considerable citation rates, but this is not the case with experimental CS scientists, where the gestation time is at least three years for a significant paper to be published in a serious journal (e.g., IEEE Transactions on Software Engineering). With a systematic lack of understanding of the CS discipline, administrators often resort to unfair comparison. As a consequence, a number of CS departments have skewed profiles of academics, mostly with non-CS trained academics dominating over properly CS-trained academics.

### ***Lack of Administrative Will to let the CS Educators do their Job***

We have also noticed that there is a lack of will on the part of administrators to let the CS educators do their job. For example, CS educators are frowned upon when they engage in paid consultancy. Somehow paid consultancy is considered “*work for personal benefit*”. On the contrary, paid consultancy must be made mandatory, just as it applies to General Practice in the field of Medicine. It is through consultancy, an academic can “*preach what s/he practices*”<sup>2</sup> and this has profound impact on the way CS courses are being taught. Consultancy also results in better education and understanding of current practices and procedures in the industry, besides providing knowledge in the state-of-the-art technology. Unfortunately, this issue is very much side-stepped by University administration.

### ***“Fill the class with sheep heads” Syndrome to Justify Existence of CS Department***

A lack of recognition and administrative will has other implications too. In the CS heyday, most Universities considered CS Departments as “*cash cows*” into which a considerable number of students could be pushed into in order to get their quota of monies from the Federal Government. These “*sheep heads*” (as one of the academics described succinctly his/her students) had neither background nor interest in the CS field; their only hope was the possibility of jobs. Furthermore, the academics had more than one reason to give a certain percentage of pass rates in the field in order to protect themselves. As a consequence, some Universities produced CS graduates with extremely poor quality and indeed some industries would not even interview graduates from certain Universities. Along with the considerable number of courses a typical CS academic has to teach, s/he is left with little time to keep themselves abreast of the field. This downward spiral has led to continued reduction in the quality of CS education and erosion of moral and ethical standards.

## **Lessons Learnt from Past Experience**

What actions could CS educators have taken to prevent the falling enrolment? Before we look into what actions CS educators can take to prevent the falling enrolment, it is worth looking at what is happening in the current “*CS-boom countries*”, such as India and China. We trace their successes and the underlying causes, before we give present our recommendations.

---

<sup>2</sup> Indeed the great management guru, Prof. Peter Drucker, mentions in his recent book that the reason he did not take up the professorship at the Harvard University (in their famous School of Business) was precisely because Harvard would not allow him to practice management through paid consultancy. According to Drucker, he could not preach what he could not practice. Harvard changed the policy much later, admitting their failure to do so much earlier was indeed a mistake.

## ***A Brief Inspection of India, China, Japan and Other Asian & Third World Countries***

Currently there is a boom in IT education in India, thanks primarily to the significant change in geopolitical equations around the world and the consequent IT in-sourcing into that country. An off-shoot of one of the largest Indian IT companies in India, viz. Hindustan Computers Limited (HCL, India), is the NIIT (National Institute for IT; <http://www.niit.com/>), which is now the largest IT education service provider in the world. NIIT has even started a private University in the USA specifically aimed at IT education and research. NIIT currently operates in 42 countries around the world, including China. Similarly the Indian Government has started another institution, called the Indian Institute of Information Technology (IIIT) at Hyderabad (<http://www.iiita.ac.in/>) (now being fondly called Cyberabad) with the specific aim to foster IT education and research. Another institution, called the Indian Institute of Banking and Finance (IIBF) ([http://www.utiicm.com/fac\\_members.htm](http://www.utiicm.com/fac_members.htm)), has also been started recently to foster IT applications research into the financial sector. Both IIIT and IIBF are deemed Universities, which compare at par with the famous Indian Institute of Technologies (IITs, <http://www.iitd.ernet.in/>) and Indian Institute of Science (IISc, <http://www.iisc.ernet.in/>) in their quality of student in-take and educational services.

Broad and important observations that have been noted in India include the following:

1. High level of mathematics is being taught at the high school level and further, mathematics is still considered fundamental towards learning all scientific, engineering and various other fields.
2. High level of professionalism among teachers has been observed in teaching mathematics, formal logic and other analytical skills at the high school level.
3. Competition to get into good and reputed educational institutions is quite high and this competition in turn drives higher the quality of students seeking to study at the University level. In a recent *60 minutes* program (produced in 2005), one leading industrialist, Mr. Narayana Murthy (CEO of InfoSys Pty Ltd, currently the leading software outsourcing company in the world) remarked that his son could not get into one of the IITs despite very good preparations and hence he admitted him into Cornell University. A recent US Senate resolution singled out its praise for the profound contributions made by IITs to the economy of the USA.
4. A great degree of interaction exists among students with the spirit of “*co-operate to compete*”. Indeed at most IITs and equivalent institutions, a very substantial database of questions and answers for various types of examinations are available to students at their living quarters<sup>3</sup>. These datasets have been collected over a period of time by the student themselves. So much so, one academic remarked that if he wants to look for variations and novel ideas for a question, s/he gets into one of these databases to explore further. Another academic commented that these databases typically appear to be better than the Schaum’s Outline Series published by the McGraw Hill!
5. A good number of books are relatively cheaply available on fundamental topics of mathematics, computer science, linguistics and logic, thanks to most publishers printing exclusive “*India Only*” printed versions. These have since had a profound effect on CS education.
6. Having missed the “*Manufacturing Bus*” to China, the Indian Government decided to open up the economy for the software sector, which led to the extensive contracts in software outsourcing into India.

---

<sup>3</sup> The author has first hand experience on this, having graduated from the Indian Institute of Science.

7. The emphasis on good English language skills has led to numerous jobs being opened to CS graduates. Indeed English is a compulsory second language in almost all schools in India.
8. The availability of heterogeneous mix of people in India, particularly with vastly different linguistics skills, means that the thought processes during the various stages of the software life cycle (i.e., requirements elicitation, analysis, design, implementation, testing and maintenance) are diverse. This leads to a multitude of issues being documented and attempted during software development.
9. The “*publish or perish*” syndrome has not affected the Indian CS education environment considerably yet. This has resulted in good teaching being considered a major part of an academic activity. Further, good teachers are being recognised by the government and other NGO (Non-Governmental Organisations) organisations periodically.
10. The participation of private organisations in CS/IT education has been profound and their quality has also been quite good, thanks to the vibrant competition in this sector. The most notable one is the NIIT, which has now become the world’s largest CS/IT education service provider, covering some 42 countries around the globe. Indeed admission to NIIT Diploma courses is through several screening exams in which the selection rate is around 30% or less.
11. Industry participation even in Government section CS/IT education has been very good, as industries are willing to actively supervise final year project students. Their aim appears to be two fold: i) to get part of their work done at low or no cost and ii) to identify potential employees, that could have been well-trained. Several organisations have reported good success in this type of active involvement.
12. Various programs in mass media (e.g., television, radio and even local Women’s weeklies) have started educational columns on computing, which has revolutionised the way moms’ and dads’ thinking on computers. Nearly every village of India has at least one computer and this has transformed businesses and their requirements considerably. Thus farmers and fishermen are selling their products for the best prices they can fetch, instead of through the local middlemen. Indeed requirements by such customers have now forced various types of e-Commerce and related technologies to come to the forefront.
13. The multitude of languages in India<sup>4</sup> has made computer-linguistics develop rapidly. This is a field which is likely to have the next stage of growth and a company like Verity Inc. is likely to be the next Microsoft, because their products are highly sought after by companies involved in the fields of Finance, Marketing, Advertising and Media. A number of companies (notably CDAC – Centre for Development of Advanced Computing) have taken up the technical challenges in computer-linguistics from an international perspective and are fast competing with Multinationals.

The situation appears to be similar in China and other third world countries. In particular, the cost of “*intellectual labour*” is considerably less than in the Anglo-countries. While China is still struggling to cope with India in CS/IT area, mainly due to the lack of English skills among its service providers, it is nonetheless catching up fast. Indeed several self-learning English kiosks are operating in China, in addition to private organisations involved in teaching English.

---

<sup>4</sup> India has 18 official languages and 50+ actual languages and over 500 dialects. The two mother languages are Tamil and Sanskrit.

### ***What are the Real Underlying Issues?***

IT is now going through what “*manufacturing*” was going through some 40 years ago. There was big hue and cry, when mass manufacturing migrated to Japan, particularly a decade after World War II. American “*jobs were being lost*”, as it was portrayed then, unfortunately did happen. However, America moved on to other areas, particularly in computer systems manufacturing, popularly known as “*Silicon Manufacturing*”. Later even silicon manufacturing moved to Japan. Presently, most types of manufacturing are happening in China and other Dragon countries, while the majority of high-end manufacturing is happening in Japan. Whereas the United States still maintains a comfortable edge in high-end CPU (Central Processing Unit) manufacturing and related R&D areas.

Another important issue that is happening in the IT sector is the consolidation of major companies – i.e., big fish eating small fish – just as it happened to railways in the 1930s in the USA. As a consequence, a few multinational companies are dominating several areas of IT and the Internet and related services. Their Standards are over-riding Standards set by such organisations, as the IEEE (Institute of Electrical and Electronics Engineers), ISO (International Standards Organisation) and OMG (Object Management Group), particularly so in the software services area. CS educators who are not following these and adapting to the current state-of-the-art are doing so at their own peril.

There is a crying need to open up new and innovative areas in CS and IT. As mentioned before, CS/IT owns their continued existence to identifying and substantiating on-going relations with many other disciplines. One such area (besides a number of others), where such issues are dominating is the area of Information Management and Information Fusion, particularly due to information overload that is currently happening. However, few academic institutions are offering formal courses or degrees in these upcoming areas. In particular, media technologies will become an important IT application focus area in the future to come.

### ***Why didn't the CS Educators Take the Required Actions?***

It is our contention that one of the reasons that CS educators did not take the required actions is that they were driven by blind competition to fill-up seats so that their positions are safe. It is also true that the process was partially driven by greed to expand the CS faculty at a time when opportunity prevailed. Furthermore, Government regulations have limited University finances to such an extent that this opening provided a good opportunity get further funding in a “*growth area*”. However, little or no efforts were made at planned growth. It is not that issues were not unknown to CS educators, but the syndromes persisted as to “*Who will bell the cat?*” and “*Why should we do that?*” We also contend that there was a systematic lack of vision to “*foresee, accept and act*” on the future on the part of CS educators. Given the ground realities of CS and IT, the educators need to pull up the socks – particularly at high school level education.

### ***Why are Some Universities able to Attract Full Enrolment or Even Over-Enrolment Despite the Downturn?***

There are a number of reasons as to why some Universities are able to attract full enrolment or even over-enrolment despite the downturn, which include, but not limited to the following:

- Mainly because of their historically well-known high profile (e.g., MIT, UC Berkeley, etc).
- Active fostering of foreign students – foreign students are not considered as “*necessary evils, but as affirmative good*”. A number of G8 Universities in Australia (the top 8 Universities are called G8) have nearly failed in this area.

- Understanding and appreciating foreign student culture, learning mechanisms (particularly learning by examples and group learning), delivery of lectures and linguistic requirements.
- Industry involvement and participation.
- Addressing the issues mentioned in this paper and others in a logical manner taking into considerations the market realities.

Indeed we believe that less than 10% of the Universities in the Anglo-Saxon world are taking these issues seriously and attempting to find their way ahead.

### ***What Lessons can be Learnt out of this Downfall?***

Sound planning is required to support CS/IT education. Key industrialists and lobby groups must be fostered to keep the relevant Minister and their Departments to be in-tune with ground realities and latest market conditions. In a recent conference, Bill Gates mentioned that funding from the American NSF (National Science Foundation) has nearly halved for CS/IT research and he has voiced profound concerns it will have on almost all fields of research. In Australia, the Tourism industry advisory group lobbies and obtains funding to the tune of 400+ million dollars, but the IT lobby is poorly represented. The IT education lobby is even more poorly represented – perhaps, it may even be non-existent.

Periodic re-learning is fundamental to CS/IT education. This has been one of the key deficiencies of the current environment. Engaging in on-going industrial consultancies and learning current state-of-the-art technology is of paramount importance.

Actively diversifying CS/IT into other areas, establishing and sustaining their R&D is very important.

Creating a core body of knowledge in CS/IT that can be easily replicated using automation is the foundation to the success of this profession – just in as much as Civil and other Engineers have done in their worlds. Prof. David Parnas (a pioneer in software engineering) in his keynote address at the ASWEC conference (Parnas, 2005), specifically mentioned the lack of core body of knowledge in CS/IT/SE as a major impediment to further progress in these fields. As a consequence, people keep on re-inventing the wheel.

Exposing future students to such opportunities will be the key to increasing student numbers.

### ***What are the Comparative Perspectives on CS Education in the Non-Anglo-Saxon World?***

Lacking a core body of knowledge in CS/IT, most of the non-Anglo-Saxon world is placing considerable emphasis on fundamental subjects like Mathematics, Statistics, Physics and others. This approach provides their students with critical analytical abilities to think logically and to handle complex problems, which in turn help these countries establish a sound environment for software in-sourcing. China and Korea, for example, have created several Universities and Schools, where English is the medium of instruction, particularly in IT related areas.

These countries are trying to understand and come to terms with the global white collar labour market, their potentials and their added-value to their economies. They have created a new stream of high value employment. As a consequence, these countries are providing substantial tax concessions or subsidies to ICT industries to start new venture companies.

Several countries are also encouraging professionals from IT industries to actively participate in educational programs; they offer grants and aids-in-support to encourage such transitional arrangements.

It is recognised that the Digital Divide still persists, particularly in the third world countries, but these countries are developing useful mechanisms to bridge the divide through their sheer “*knowledge power*” that is inherent in their people.

## Recommendations for Managing Future CS Departments

We present below a list of recommendations that CS Departments should consider for their future:

1. Allow teaching staff to undergo at least one month of industrial training per year. Attending “*academic/s (dominated) conferences*” is not at all a substitute in this case.
2. Allow teaching and tutorial staff to be active participants in consultancy programs. Indeed this activity must be made mandatory to the extent that their corresponding educational institutions must provide avenues for staff to perform consultancies. Note that this is a win-win situation for both the staff and their Institution.
3. Encourage academic staff to spend their sabbatical (study) leave at relevant IT industries, rather than in another academic institution.
4. Encourage industry based professionals to offer serious academic equivalent courses at the University. Their participation is vital for the robustness of the entire program.
5. Provide mandatory summer-term bridge courses in advanced mathematics (make this a prerequisite, if necessary) that high school leavers must undergo and pass, before they can take their CS program. Do not allow unprepared students into the CS course. Avoid the temptation to fill the class with poor students just for the sake of money. They will become a drain to the staff, the University and the society. This will also lead to greater intellectual competition leading to better quality of outputs.
6. Create or be part of lobby groups that advise the relevant Minister (or their Cabinet Secretary) on IT related issues and, in particular, on the corresponding skill shortages and related educational issues.
7. Accept that CS/SE/IT is different to traditional Science and/or Engineering and allow them to be evaluated on their performance differently, particularly since the area is still evolving. This will give academics the much-needed confidence (particularly in rural sectors) to compete on a “*level playing field*”. CS staff performance must also include the fact that the field keeps changing all the time (in particular in IT and SE areas). Their preparation times are grossly under-valued currently. Their research outputs, which appear mostly in conferences (as opposed to other fields, where they tend to prefer journal publications), are often disparaged. This attitude must change and a concerted effort is required to educate the people around.
8. Allow private participants to offer relevant CS/IT programs. Their quality can be monitored and adjudicated through accreditation programs from the ACS (Australian Computer Society), IEAust (Institution of Engineers, Australia) and other similar bodies.
9. Allow research and academic staff to develop broader field research profile and give them suitable avenues and opportunities to explore newer areas. During this period suitable funding and performance measures (not the traditional measures) must be applied.
10. Use appropriate marketing strategies to educate high school leavers; “*catch them young*” a popular advertisement slogan from the McDonalds is very relevant here. Involve student participation in CS/IT/SE related areas at various levels and let Universities employ a suitable (paid) person as high school liaison officer to address prospective students’ fears and dreams.

11. The curriculum must be amended to include real world industrial input. This means even some of the Microsoft<sup>5</sup> specific technical issues must be addressed and discussed in the curriculum.
12. While emphasis must be placed on programming skills, it is very essential that a greater emphasis must be placed on design, maintenance and testing of software systems. The latter issues determine the success of products better. While coding will become a routine job, design will still require advanced cognitive skills.
13. Emphasise good English language<sup>6</sup> and oral and written communication skills. Longer term team-based projects are also essential. Allow “*incubators*” to be set-up within the University system, with proper academic and industry joint supervisions.
14. Develop University-based “*Continuing Education in IT*” and “*Advanced Professional Development Programs*” so that the community and industry professionals can be engaged with the University system on an on-going basis. The “*Quality Improvement Program (QIP)*” conducted at the Indian Institute of Science, Bangalore, has been an outstanding success (on a number of factors) for over 30 years.
15. Aim to develop a corporate body of knowledge. This cannot be done in isolation, but in partnership with Universities and industries around the world. One approach is to foster “*sister University*” concept further.
16. Be not alarmed by the “*invasion of skilled professionals*” from other parts of the world. The resulting heterogeneous talent pool will only enhance the quality of software systems. Accept that while collar jobs are globally re-locatable and that too at short notice.

## Conclusions

This paper presents a number of issues that have been plaguing the CS education environment in the Anglo-Saxon world over the recent past, with particular reference to Australia. We argue – some times controversially - that most of the issues and problems were predictable, but CS educators failed to admit, accept and act on them. We present in this paper a list of issues that could have been addressed wisely, at least in hindsight. There are a number of lessons to be learnt from the past and also from other countries, wherein CS education is really booming. Two of the key issues we argue are i) the lack of recognisable private CS/IT education providers in Australia and ii) significant lack of industry participation in CS/IT educational scenario.

We argue that all is still not lost. History has shown a number of scenarios from which several lessons can be learnt. We identify some of these issues in this paper and also the ways to look forward to the future of CS/IT education in Australia. Even though some of our arguments and assertions may be controversial, they do form pointers for brainstorming the underlying problem in the CS Educational Community. It is a pity that we could not provide some of the real information due to privacy reasons; consequently they are mentioned as anecdotal.

## References

Arora, S. & Chazelle, B. (2005). Is the thrill gone?. *Communications of the ACM*, 48(8), 31-33.

---

<sup>5</sup> The author does not own any shares in Microsoft, but if any one wishes to bequeath some to the author, he will gladly accept -;)

<sup>6</sup> English is still the preferred language, while addressing various issues that arise during software development across international boundaries.

Harvard Business Review, 1996.

Higher Education Research Institute at the University of California at Los Angeles (HERI/UCLA).

Parnas, D. (2005). Keynote address at the Australian Software Engineering Conference (ASWEC'05).

Wikipedia. (n.d.). entry on comparing software engineering with other disciplines:

[http://en.wikipedia.org/wiki/Comparing\\_software\\_engineering\\_and\\_related\\_fields](http://en.wikipedia.org/wiki/Comparing_software_engineering_and_related_fields).

Wilson, B. (n.d.). Possible consequences of reduced student numbers for CSE teaching. Retrieved from

<http://www.cse.unsw.edu.au/~teachadmin/tc/downturn.html>

## Biography



Prof. **Lakshmi Narasimhan** obtained his Bachelor's degrees in Physics and Electronics Engineering from the University of Madras and Indian Institute of Science respectively. He also obtained his Master's and Phd degrees respectively from the Madras Institute of Technology and University of Queensland, Australia. He worked at his *alma mater* in Australia as a lecturer and senior lecturer and was also the Director of the PA3SE (Parallel Algorithms, Architectures And Software Engineering) Research Laboratory. In 1995 he moved to the Australian Defence and Science and Technology Organisation (DSTO) as a Principal Research Scientist and was leading the program on Information Management and Information Fusion. He has been instrumental in initiating

few key computer information systems projects with strategic clients within the Australian Defense Force, which have since led to the deployment of several mission critical systems at various customer sites. Later he moved to the USA, where worked as a full professor at the Computer Science Department, University of North Texas at Denton. Currently he is the Professor and Chair in Software Engineering at the University of Newcastle, Australia.

Prof. Narasimhan has published over 120 papers in the areas of Software Engineering and Information Engineering. In particular, his research interests are in computer architecture, parallel and distributed computing, software testing, text & audio processing, E-Commerce, Software process, and information management & fusion. His papers have appeared in such archival journals as the various IEEE Transactions and IEE Proceedings. Prof. Narasimhan was the founding chair of the IEEE Computer Society, Queensland Chapter and initiated the International Conference series ICA3PP (IEEE International Conference on Algorithms And Architectures) and acted as its Inaugurating Chair. He has also been the technical chair of two other conferences and has been on the technical panel of over 30 leading International Conferences. Currently his research interests are the areas of large-scale system engineering, testing & visualization, Agent technology for Decision Support System design and distributed object computing. He has consulted to a number of industries and educational institutions on various IT and Software Engineering projects. Presently he is also associated with two Co-Operative Research Centres (CRC on Construction Innovation and CRC on Integrated Engineering Asset Management). He is also the Director of the MESTEC Research Centre at the University of Newcastle, which specialises in the applications of mobile technology to medical information management and patient care. Prof. Narasimhan is a Senior Member of the IEEE, ACM, Fellow of ACS and IEAust. He is an (technical) Expert Panel Member of ISO (International Standards Organisation) and MIMOSA (Machinery Information Management Open Standards Organisation, USA).